

**Томский государственный университет  
систем управления и радиоэлектроники**

**В.В. Кручинин, Ю.Н. Тановицкий,  
С.Л. Хомич**

**КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ  
В НАУКЕ, ОБРАЗОВАНИИ И  
ПРОИЗВОДСТВЕ ЭЛЕКТРОННОЙ  
ТЕХНИКЕ**

**Учебное пособие**

**Томск  
2012**

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное учреждение высшего  
профессионального образования  
**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ**

**Кафедра промышленной электроники (ПрЭ)**

**В.В. Кручинин, Ю.Н. Тановицкий, С..Л. Хомич**

**Компьютерные технологии в науке,  
образовании и производстве  
электронной технике**

**Учебное пособие**

**2012**

**Кручинин В.В., Тановицкий Ю.Н., Хомич С.Л.**

Компьютерные технологии в науке, образовании и производстве электронной технике: Учебное пособие. — Томск: Томский государственный университет систем управления и радиоэлектроники, 2012. — 154 с.

© Кручинин В.В., Тановицкий Ю.Н.,  
Хомич С.Л., 2012

© ТУСУР, 2012

## ОГЛАВЛЕНИЕ

<b>Введение .....</b>	<b>7</b>
<b>1. Наука как объект компьютеризации .....</b>	<b>9</b>
1.1. Виды научной информации и способы ее представления ...	11
<b>2. Компьютерные технологии в образовании .....</b>	<b>14</b>
2.1. История вопроса .....	14
2.2. Основные определения и классификация .....	17
2.2.1. Автоматизированные учебные курсы и компьютерные учебники .....	19
2.2.2. Компьютерные лекции .....	21
2.2.3. Интеллектуальные обучающие системы.....	21
2.2.4. Генераторы тестовых заданий, задач и вопросов .....	24
2.2.5. Тестирующие программы .....	29
2.2.6. Тренажерные программы.....	31
2.2.7. Моделирующие программы .....	32
2.2.8. Компьютерные учебные игры .....	34
2.2.9. Прикладные программы, используемые при обучении ....	35
2.2.10. Инструментальные системы .....	37
2.2.11. Интегрированные системы .....	41
2.3. Программное обеспечение дистанционного обучения .....	42
<b>3. Научный поиск в Интернет .....</b>	<b>47</b>
3.1. Универсальная поисковая система.....	47
3.2. Высшая аттестационная комиссия .....	48
3.3. Федеральное государственное бюджетное учреждение науки Всероссийский институт научной и технической информации Российской академии наук.....	50
3.4. Федеральное государственное научное учреждение «Центр информационных технологий и систем органов исполнительной власти» .....	51
3.5. arXiv.org.....	52
3.6. Специализированная научная поисковая система Scirus ..	53
3.7. Академия Google .....	55
3.8. Фонд содействия развитию малых форм предприятий в научно-технической сфере .....	55
3.9. Российский фонд фундаментальных исследований.....	56
3.10. Федеральное государственное бюджетное учреждение «Федеральный институт промышленной собственности» (ФИПС).....	58

3.11. Роспатент .....	59
<b>4. Система представления учебной и научной информации (LATEX).....</b>	<b>61</b>
4.1. Общие сведения. История. Текущее состояние.....	61
4.1.1. Система компьютерной верстки TeX .....	61
4.2. Быстрое введение. Документ и его структура .....	64
4.2.1. Входной файл .....	64
4.2.2. Преамбула.....	65
4.2.3. Текст документа.....	66
4.2.4. Поддержка русского языка .....	67
4.2.5. Комментарии .....	68
4.3. Печатный документ .....	69
4.3.1. Параметры страницы.....	69
4.3.2. Титульная страница и аннотация .....	70
4.3.3. Секционирование документа.....	71
4.3.4. Части.....	72
4.3.5. Главы .....	73
4.3.6. Разделы.....	73
4.3.7. Параграфы .....	74
4.3.8. Приложения .....	74
4.4. Основные команды .....	74
4.4.1. Переключение в математическую моду.....	74
4.4.2. Формулы внутри текста .....	75
4.5. Выделенные формулы .....	76
4.5.1. Однострочные уравнения .....	76
4.5.2. Системы уравнений .....	77
4.6. Списки .....	79
4.6.1. Маркированные списки.....	79
4.6.2. Нумерованные списки.....	80
4.7. Позиционирование текста в строке.....	81
4.8. Таблицы.....	82
4.8.1. Таблицы заданной ширины .....	83
4.8.2. Размещение таблицы на нескольких страницах.....	83
4.9. Вставка графики .....	85
4.10. Плавающие объекты .....	85
4.10.1. Рисунки и таблицы.....	85
4.10.2. Рисунки и таблицы, обтекаемые текстом.....	87
4.10.3. Управление плавающими объектами .....	88

4.11. Обзор пакетов .....	89
4.11.1. Интернационализация и локализация.....	89
4.11.2. Рубрикация.....	90
4.11.3. Пакеты для работы с рисунками .....	90
4.11.4. Пакеты для работы с таблицами .....	91
4.11.5. Пакеты для работы с презентациями.....	91
4.11.6. Пакеты для точных наук .....	92
4.11.7. Пакеты для гуманитарных направлений .....	96
4.12. Редактор TEXSTUDIO .....	97
4.12.1. Редактирование документов TeX.....	98
<b>5. Система Maxima .....</b>	<b>105</b>
5.1. Системы численных и символьных вычислений .....	105
5.2. Об истории системы Maxima .....	106
5.3. Графический интерфейс системы Maxima .....	106
5.4. Выражения .....	107
5.5. Массивы .....	108
5.6. Списки .....	109
5.7. Упрощение выражений.....	110
5.8. Решение уравнений.....	111
5.9. Интегрирование.....	111
5.10. Нахождение пределов .....	112
5.11. Разложение функции в ряд.....	112
5.12. Программирование в системе Maxima.....	113
5.12.1. Операторы цикла.....	113
5.12.2. Условный оператор.....	113
5.12.3. Функция map .....	114
5.12.4. Функция block .....	114
5.12.5. Создание собственных функций .....	115
5.12.6. Вывод графиков .....	115
<b>6. Система моделирования asimesc .....</b>	<b>117</b>
6.1. Назначение .....	117
6.2. Основное окно ASIMESC.....	118
6.2.1. Создание схемы.....	119
6.2.2. Параметры электронных компонентов .....	120
6.3. Элементы схем .....	121
6.3.1. Источники.....	121
6.3.2. Транзисторы и диоды .....	123
6.3.3. Общие сведения о моделировании .....	124

6.3.4. Управление режимами моделирования.....	126
6.3.5. Особенности проведения измерений.....	130
6.4. Аналоговое, цифровое и смешанное моделирование .....	131
6.4.1. Аналоговое моделирование .....	131
6.4.2. Цифровое моделирование .....	132
6.4.3. Смешанное моделирование .....	132
6.5. Примеры некорректных схем .....	132
<b>7. Система дистанционного обучения Moodle .....</b>	<b>134</b>
7.1. Основные принципы и функции системы дистанционного обучения Moodle .....	135
7.2. Управление системой дистанционного обучения .....	135
7.3. Управление пользователями.....	136
7.4. Запись на курсы.....	137
7.5. Роли.....	137
7.6. Управление курсом.....	138
7.7. Ресурс .....	139
7.7.1. Лекция .....	140
7.7.2. Глоссарий.....	140
7.7.3. База данных .....	141
7.7.4. Тест.....	142
7.7.5. Задание.....	143
7.7.6. Форум.....	144
7.7.7. Чат.....	144
7.7.8. Опрос.....	145
7.7.9. Анкета .....	145
7.7.10. Wiki.....	145
7.7.11. Семинар.....	145
7.8. Система дистанционного обучения ТУСУРа .....	146
7.9. Система контроля знаний ФДО ТУСУРа.....	147
7.9.1. Выполнение экзаменационной работы в режиме online.....	148
7.10. Типы вопросов.....	150
7.10.1. Меню .....	150
7.10.2. Ввод ответа .....	151
7.10.3. Вопрос с пропусками.....	151
7.10.4. Множественный выбор .....	152
<b>Литература .....</b>	<b>153</b>

## ВВЕДЕНИЕ

Целью данной дисциплины является изучение и освоение современных компьютерных и информационных технологий, позволяющих при проведении научных исследований пользоваться глобальными информационными ресурсами, современными пакетами моделирования и автоматизации научных исследований.

После изучения данной дисциплины студент должен знать принципы построения глобальных компьютерных сетей, уметь пользоваться их информационными, вычислительными ресурсами, поисковыми системами, системами издания и редактирования научных публикаций, системами моделирования.

Изучение дисциплины базируется на фундаментальной физико-математической подготовке, на знании методов программирования и моделирования, элементной базы микропроцессорной техники и сетей передачи данных. Изучаемая дисциплина является предшествующей при изучении специальных и профилирующих дисциплин, а также при подготовке магистерской диссертации.

Под компьютерной технологией понимается использование компьютеров для достижения некоторых целей в определенной предметной области. Применение компьютеров в науке, образовании и производстве электронной техники должно обеспечить:

1. Упрощение и ускорение процессов обработки, передачи, представления и хранения научной или учебной информации.
2. Увеличение объема полезной информации с накопителем типовых решений и обобщением опыта научных разработок.
3. Повышение качества решаемых задач. Возможность реализации задач ранее не решаемых. Постановка исследований и получение результатов, недостижимых другими средствами.
4. Возможность анализа большого числа вариантов синтеза объектов и принятия решений.
5. Сокращение сроков разработки, трудоемкости и стоимости НИР при улучшении условий работы специалистов.

6. Улучшение качества обучения за счет разработки и использования программно-методического обеспечения

7. Внедрение дистанционных технологий обучения.

В результате изучения дисциплины студент должен:

**Знать:** основные методы создания и редактирования научных изданий (препринтов, отчетов, статей, монографий, диссертаций). Методы поиска и глобальные системы поиска научной информации. Методы и системы компьютерного моделирования и символьных вычислений.

**Уметь:** обоснованно выбирать и применять системы поиска и моделирования и представлять результаты научных исследований с помощью современных компьютерных и информационных технологий.

**Владеть:** навыками создания и редактирования научно-технической информации средствами компьютерных издательских систем, поиска и публикации научных изданий в Интернет, использования систем моделирования и символьных вычислений.

## 1. НАУКА КАК ОБЪЕКТ КОМПЬЮТЕРИЗАЦИИ

Известно, что наука — это сфера деятельности, направленная на получение новых знаний, которая реализуется с помощью научных исследований. Целью НИ является изучение определенных свойств объекта (процесса, явления) и на этой основе разработка теории или получение необходимых для практики обобщенных выводов. По целевому назначению научные исследования делятся на фундаментальные и прикладные.

Фундаментальные связаны с изучением новых явлений и законов природы, с созданием новых принципов исследований (физика, математика, биология, химия и т.д.).

Прикладные исследования (ПНИ) — это нахождение способов использования законов природы и научных знаний, полученных в ФНИ, в практической деятельности человека. Важным показателем прикладных исследований является улучшение качества жизни человечества в самой широкой ее интерпретации. Одним из видов прикладных исследований являются научно исследовательские опытно-конструкторские разработки (НИОКР). НИОКР — это процесс создания новой техники, систем, материалов и технологий, включающий подготовку документов для внедрения в практику результатов прикладных научных исследований. В зависимости от *жизненного цикла изделий*, могут быть выделены следующие типовые этапы НИОКР:

### *Исследование*

- Проведение исследований, разработка технического предложения.
- Разработка технического задания на опытно-конструкторские (технологические) работы.

### *Разработка*

- Разработка эскизного проекта.
- Разработка технического проекта.
- Разработка рабочей конструкторской документации на изготовление опытного образца.
- Изготовление опытного образца.
- Проведение испытаний опытного образца.
- Отработка документации.

- Утверждение рабочей конструкторской документации для организации промышленного (серийного) производства изделий.

#### *Поставка продукции на производство и эксплуатация*

- Корректировка конструкторской документации по выявленным скрытым недостаткам.
- Разработка эксплуатационной документации.

#### *Ремонт*

- Разработка рабочей конструкторской документации на проведение ремонтных работ.

#### *Снятие с производства*

- Разработка рабочей конструкторской документации на утилизацию.

При выполнении научных исследований широко применяются разнообразные методы исследований, которые делят на следующие группы: эмпирические, экспериментальные и теоретические.

Эмпирические исследования выполняются с целью накопления систематической информации о процессе (явлении). При этом используются методы: наблюдение, регистрация, измерение, анкетный опрос, тесты, экспертный анализ.

Экспериментальный уровень научных исследований — это изучение свойств объекта по определенной программе с использованием лабораторного или иного оборудования.

Теоретические исследования проводятся с целью разработки новых методов решения научно-технических задач, обобщения и объяснения эмпирических и экспериментальных данных, выявления общих закономерностей и их формализации.

На двух последних уровнях используются методы моделирования, методы анализа и синтеза, логические построения (предположения, умозаключения), аналогии, идеализации.

Рациональная организация НИР строится с использованием принципов системного подхода и схематично может быть представлена следующим образом.

Исходя из задач НИ и порядка их реализации, можно определить следующие основные направления рационального применения КТ в научных исследованиях:

1. Сбор, хранение, поиск и выдача научно-технической информации (НТИ).
2. Подготовка программ НИ, подбор оборудования и экспериментальных устройств.
3. Математические расчеты.
4. Решение интеллектуально-логических задач.
5. Моделирование объектов и процессов.
6. Управление экспериментальными установками.
7. Регистрация и ввод в ЭВМ экспериментальных данных.
8. Обработка одномерных и многомерных (изображения) сигналов.
9. Обобщение и оценка результатов НИ.
10. Оформление и представление итогов НИ.
11. Управление научно-исследовательскими работами (НИР).
12. Сбор и обработка НТИ (эмпирические исследования).
13. Теоретические исследования.
14. Обработка результатов исследования.
15. Экспериментальные исследования.
16. Моделирование.
17. Представление и оформление результатов.

### **1.1. Виды научной информации и способы ее представления**

При системном подходе НИ начинаются со сбора и предварительной обработки НТИ по теме исследования. Эта информация может включать сведения о достижениях в исследуемой области, об оригинальных идеях, об открытых эффектах, научных разработках, технических решениях и т.д.

Углубленное изучение информации по предмету исследования позволяет исключить риск ненужных затрат времени на уже решенную проблему, детально изучить весь круг вопросов по исследуемой теме и найти научно-техническое решение, отвечающее высокому уровню.

Основным источником информации являются научные документы, которые по способу представления могут быть текстовыми, графическими, аудиовизуальными и машиночитаемыми, электронными и т.д. Научные документы подразделяются на пер-

вичные и вторичные, опубликованные и неопубликованные. Первичные документы — это книги, брошюры, периодические издания (журналы, труды), научно-технические документы (стандарты, методические указания). Важное значение здесь имеет также патентная документация, под которой подразумеваются издания, содержащие сведения об открытиях, изобретениях и т.п. К неопубликованным первичным документам относятся: научные отчеты, диссертации, депонированные рукописи и т.п.

Вторичные документы содержат краткую обобщенную информацию из одного или нескольких первичных документов: справочники, реферативные издания, библиографические указатели и т.п. Рассмотрим подробнее виды научно-технической информации:

1. **Тезисы докладов и выступлений** — кратко сформулированные основные научные идеи по теме исследования.

2. **Научная статья** — это произведение, посвященное тематике диссертационного труда, имеющее цельный и законченный вид. Используется для отражения более значимых научных результатов, требующих развернутой аргументации.

3. **Монография** — вид публикации, содержащий систематическое изложение основных данных научного труда. Чаще всего используется как форма публикации при написании диссертации на соискание докторской степени.

4. **Методические разработки (рекомендации)** — часто применяемая форма для изложения результатов научных исследований по различного рода проблемам.

5. **Учебное пособие (учебник)** — наиболее трудоемкий вид публикации, поскольку помимо изложения результатов диссертационного труда соискатель обязан в полной мере раскрывать темы и программу соответствующей учебной дисциплины.

6. **Депонирование** — размещение результатов исследования в глобальной сети для общего доступа или же передача их на хранение в специализированные хранилища (поскольку исследования могут быть адресованы лишь узкому кругу специалистов и не нуждаются в больших тиражах).

7. **Диссертация** — это научно-исследовательская работа, проводимая соискателем с целью получения ученой степени. Эта работа отражает степень зрелости научного сотрудника, грамотно

ставящего и решающего научные вопросы, что показывает уровень его владения

8. **Отчет по НИР** — отчет по научно-исследовательской работе.

9. **Препринт** — предварительное издание статьи, доклада или монографии, в которой излагаются научные результаты. В настоящее время всемирно признанным хранилищем является [Arxiv.org](http://Arxiv.org).

10. **Автореферат**.

11. **Отзыв** — научный труд, в котором дается оценка статьи, монографии или диссертации. Как правила, отзыв должен содержать: актуальность работы, научную новизну, практическую значимость, положительное и отрицательные стороны научной работы.

12. **Рецензия** — научный труд, в которой с критической стороны оцениваются научные результаты работы.

Поиск выполняется по каталогам, реферативным и библиографическим изданиям и в сети Интернет.

## 2. КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

### 2.1. История вопроса

Исследования, связанные с использованием компьютеров в обучении в нашей стране и за рубежом начались в конце 50-х годов. Одним из первых можно назвать проект системы PLATO. Но из-за высокой стоимости больших ЭВМ эти проекты носили экспериментальный характер. В это время ЭВМ появляются в крупных университетах и институтах. В основном они использовались для проведения исследований и обучения студентов работе на компьютере. С появлением семейства ЕС ЭВМ происходит этап широкого внедрения больших ЭВМ в вузах. Практически каждый технический вуз в нашей стране стал иметь свой вычислительный центр. В это время появляются первые промышленные системы, предназначенные для автоматизации учебного процесса, это системы СПОК, АСУ ВУЗ, ОСКАР. В США мировую известность получили системы PLATO-4, TICCIT. В основном эти системы были многотерминальными, работающими под управлением одной большой ЭВМ. В это же время появляются средства автоматизации для создания обучающих курсов. Это, прежде всего авторские языки, которые ориентированы на преподавателей-методистов, слабо знакомых с программированием. Примерами таких языков являются COURSEWRITER, TUTOR, СПОК, ЯОК.

С появлением мини-ЭВМ происходит становление учебно-исследовательских лабораторий. В типовой учебно-исследовательской лаборатории, оснащенной мини-ЭВМ, производятся лабораторные работы по моделированию различных систем, по курсам автоматизированных систем научных исследований, систем автоматизированного проектирования и т.д. Однако, кроме вузов и специализированных техникумов, такие системы образовательные учреждения не могли приобрести распространения из-за их большой стоимости. В конце 70-х — начале 80-х появляются микроЭВМ, начинается их массовое производство. Появляются персональные ЭВМ (ПЭВМ) Apple, IBM PC и т.д. Начинается широкое внедрение ПЭВМ в различные сферы человеческой деятельности. Ввиду повсеместного внедрения ПЭВМ перед развитыми странами встала задача овладения компьютерной грамотностью.

стью всего общества. В это время принимаются национальные программы по компьютеризации среднего и начального образования Франции, Англии, США, Японии и т.д. В СССР такая программа была принята в 1985 году партийно-правительственным постановлением «О мерах по обеспечению компьютерной грамотности учащихся средних учебных заведений и широкого внедрения электронно-вычислительной техники в учебный процесс». Происходит широкое внедрение ПЭВМ в учебный процесс как среднего, так и высшего образования, при этом ЭВМ используется в основном как объект изучения. В настоящее время компьютеры получили широкое распространение в обучении, появились компьютерные учебные сети, сначала локальные, объединяющие класс ПЭВМ и несколько классов, потом большие, объединяющие школы и вузы некоторого района. Ставятся эксперименты для создания глобальных учебных сетей, когда учащиеся из разных стран мира могут объединяться, передавать и принимать информацию, используя спутниковую связь. Тем самым даже маленькие сельские школы могут иметь доступ к последним достижениям науки, в том числе и педагогической. Примером такого эксперимента является проект «Велихов-Гамбург».

Другим перспективным направлением является направление, связанное с интеграцией различных технических средств обучения (ТСО) под управлением ЭВМ. Это подключение к ЭВМ видеоманитов и проекторов, аудиотехники, специальных тренажеров и т.д. Это также позволяет увеличить эффективность обучения. Это направление получило название «мультимедиа».

Интеграция компьютеров и средств связи, в том числе и спутниковой в компьютерные сети, открывает еще большие возможности компьютерным технологиям обучения, а именно:

1. Быстрое распространение знаний. Через компьютерную сеть можно иметь быстрый доступ к научным достижениям и знаниям высококвалифицированных экспертов.

2. Быстрый доступ к большим объемам информации.

3. Быстрое и эффективное внедрение педагогической науки в практику.

4. Безбумажная технология.

5. Доступность образования (независимость от места нахождения обучаемого). Маленькие сельские школы имеют те же возможности, что и городские.

6. Качественное изменение форм образования (компьютерные технологии).

Выделяют несколько этапов развития этого направления:

1. Процесс обучения строится на основе непосредственного подключения компьютеров одного или нескольких обучаемых к компьютеру преподавателя.

2. Появление университетских учебных компьютерных сетей, позволяющих организовать обучение достаточно больших групп обучаемых. Стало возможным использовать электронную почту и проводить телеконференции.

3. Объединение университетских и региональных компьютерных сетей. Появление компьютерной сети Интернет. Теперь компьютерные сети используют космические спутники связи, существенно увеличилась скорость передачи данных. Стало возможным обмениваться не только текстовой информацией, но и другими видами представления информации: графической, звуковой, мультипликационной и видео.

4. Этап, связанный с существенным влиянием сети Интернет на образование. Количество пользователей возросло до нескольких миллионов. Этот этап характеризуется виртуализацией образовательных учреждений. Появляются «виртуальные классные комнаты» и «виртуальные университеты».

Здесь под «виртуальной классной комнатой» понимается использование компьютерной сети Интернет для организации классных занятий, т.е. преподаватель и обучаемые, имея различное местоположение, включают одновременно компьютеры и, используя возможности сети Интернет, общаются между собой. Под «виртуальным университетом» понимается такая организация обучения, при которой учащийся посредством сети Интернет проходит полный курс обучения по той или иной специальности. В таком «виртуальном университете» имеется «виртуальные классные комнаты», «виртуальная библиотека», «виртуальный деканат» и т.д.

## 2.2. Основные определения и классификация

В настоящее время существует достаточно много разнообразных терминов, используемых при обсуждении вопросов внедрения и использования компьютеров в образовании. Одним из самых общих терминов является термин «Информатизация образования», который подразумевает внедрение компьютеров и компьютерных сетей во все сферы образовательной деятельности.

Для реализации новых информационных технологий в образовании необходимо разработать автоматизированную обучающую систему (АОС). Под АОС будем понимать компьютерную сеть, состоящую из компьютеров разной мощности, видео- и аудио техники, различных сервисных средств, предназначенную для поддержки учебного процесса в интерактивном режиме работы всех его участников (учащихся, преподавателей, методистов, экспертов-предметников, администраторов, разработчиков учебных и сервисных программ и др.). Для своего функционирования АОС требует разнообразного обеспечения: программного, технического, информационного, методического, лингвистического, экономического, административно-правового, кадрового. В настоящее время АОС строят на базе Интернет-технологий. Примером может служить организация Российской системы открытого образования,

Рассмотрим классификацию программ (рис. 2.1), используемых в образовательных технологиях обучения, которая основана на иерархическом представлении. Программное обеспечение — это совокупность программ, обеспечивающих функционирование АОС. Программное обеспечение можно разделить на системное и проблемное. К системному программному обеспечению относятся: операционные системы, трансляторы, загрузчики и пр. К проблемному программному обеспечению относятся программы, используемые в образовательной деятельности. Синонимом проблемного программного обеспечения можно считать термин «педагогические программные средства (ППС)», который можно определить так: ППС — это программы, предназначенные для автоматизации педагогической деятельности и реализующие некоторый этап или функцию в данной педагогической технологии. ППС могут не участвовать непосредственно в процессе обучения.

Например, программы составления расписания занятий, некоторые инструментальные средства, средства информационной поддержки и т.д. Педагогические программные средства можно разделить на следующие группы программ:

1. Компьютерные учебные программы.
2. Программы поддержки текущей деятельности преподавателя.
3. Инструментальные программы.
4. Программы управления процессом обучения.

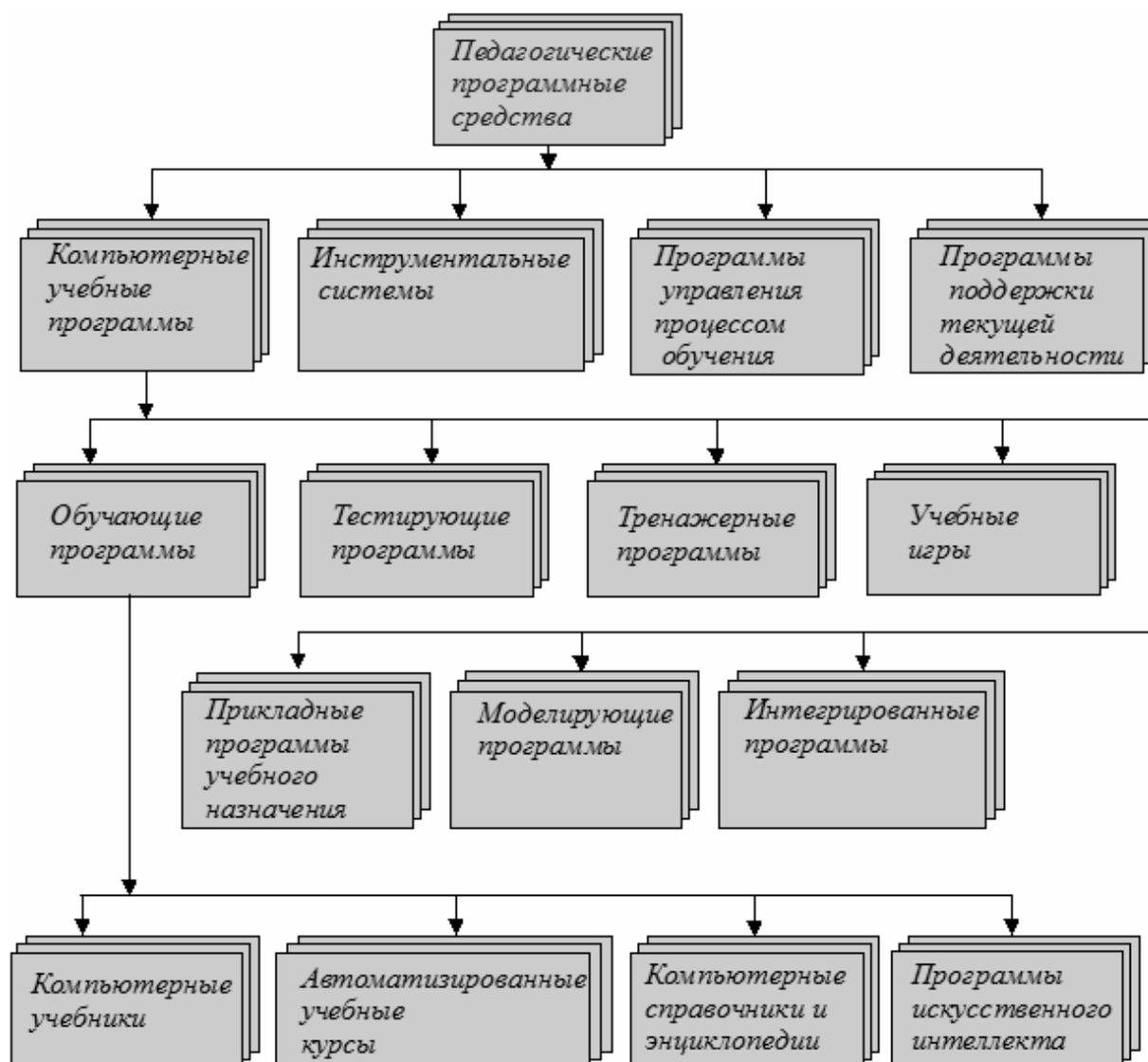


Рис. 2.1 — Классификация педагогических программных средств

Компьютерные учебные программы — это программы, предназначенные для непосредственного использования в учебном процессе. Отличительной особенностью компьютерных

учебных программ (КУП) является взаимодействие с обучаемым. Компьютерные учебные программы можно разделить:

- 1) на обучающие программы;
- 2) генераторы;
- 3) тестирующие программы;
- 4) тренажерные программы;
- 5) моделирующие программы;
- 6) учебные игры;
- 7) прикладные программы учебного назначения;
- 8) интегрированные программы.

В группу обучающих программ входят:

- 1) компьютерные лекции и презентации;
- 2) автоматизированные учебные курсы (АУК);
- 3) компьютерные учебники;
- 4) программы искусственного интеллекта;
- 5) компьютерные справочники и энциклопедии.

Рассмотрим подробнее каждую из выделенных групп программ.

### **2.2.1. Автоматизированные учебные курсы и компьютерные учебники**

Термины «автоматизированный учебный курс» и «компьютерный учебник» с точки зрения возложенных на них функций можно считать синонимами. Различия между ними заключаются в отношении их к другим компьютерным учебным программам. Обычно говорят об АУК, когда хотят подчеркнуть, что данный тип КУП является составной частью некоторой автоматизированной обучающей системы или создан с использованием некоторой инструментальной среды. Например, в рамках известной системы PLATO разработано несколько тысяч автоматизированных учебных курсов или АУК, разработанных в системе АОС ВУЗ. О компьютерном учебнике (КУ) говорят, когда хотят подчеркнуть независимость применения данного типа КУП от каких-либо внешних систем.

АУК и КУ предназначены для представления, предъявления учебного материала и проверки его усвоения. В некоторых случаях этап проверки может отсутствовать. Этот тип КУП может

использоваться во всех предметных областях. Начинаются эти программы со вступительной части, знакомящей обучаемого с задачами и характером урока. Затем подается учебный материал, после которого следуют вопросы на понимание. Ответы обучаемого оцениваются, и в соответствии с этой оценкой осуществляется обратная связь: обучаемому предлагаются либо дополнительные вопросы, либо помощь, либо правильный ответ, либо следующая порция учебного материала. АУК могут иметь предварительные тесты для определения уровня подготовки. Обучаемый может управлять процессом обучения на начальном этапе путем выбора заданий из предлагаемых меню. Представление учебного материала может производиться в текстовой и графической формах, в сопровождении мультимедиа, с выводом аудио- и видеoinформации. Проверка усвоения учебного материала осуществляется в основном с помощью многовариантного выбора, требующих односложных ответов, заданий на выполнение несложных операций (переместить курсор, щелкнуть «мышкой» и т.д.), конструктивных вопросов (вставить слово или символ, закончить предложение, составить предложение из имеющихся элементов) и т.п. Ответы учащихся обычно вводятся с клавиатуры или указываются с помощью «мышки». После ввода ответа АУК оценивает ответ, может вывести реплику и по мере проведения обучения накапливает информацию о данном учащемся. АУК могут быть линейными и разветвленными. Линейные программы последовательно предъявляют учебный материал, вопросы по выбранной теме, и в конце занятия проставляется оценка. Обучение полностью производится под управлением ЭВМ. В разветвленных программах управление может быть смешанным, после каждого вопроса производится его оценка, и в зависимости от ответа производится переход на ту или иную ветвь. Примерами таких систем могут служить учебные курсы в системах PLATO, TICCIT, АОС ВУЗ, АСТРА, Прометей.

В настоящее время компьютерные учебники это достаточно широкий класс компьютерных учебных программ, начиная от электронных учебников (представление обычного учебника в электронной форме), кончая сложными интегрированными программами.

### **2.2.2. Компьютерные лекции**

Компьютерные лекции — класс компьютерных учебных программ, предназначенных для представления лекций в компьютерном виде и предъявлении их студенту. Важно здесь провести разграничение между компьютерной лекцией и компьютерным учебником. Компьютерная лекция представлена в нетекстовом виде. Это может быть фрагмент видео или аудио записи, это может озвученная компьютерная мультипликация или моделирование, это также может быть озвученная презентация. Студент в процессе проведения компьютерной лекции, не читает текст с экрана, а смотрит и слушает лектора, который объясняет или показывает конкретную тему курса. Компьютерные лекции являются эффективным инструментом современных педагогических технологий. Современное развитие сети Интернет и соответствующей телевизионной техники привело к понятию online-лекции, когда производится трансляция лекции не по телевидению, а в сети Интернет. Кроме того, создаются целые видеотеки, хранящиеся в оцифрованном виде и содержащие большое количество видеолекций.

Однако современное техническое обеспечение каналов и дороговизна соответствующих телевизионных систем не позволяют широко внедрить в реальный учебный процесс.

### **2.2.3. Интеллектуальные обучающие системы**

Интеллектуальные обучающие системы (ИОС) — класс компьютерных учебных программ, в основе которого лежит применение идей искусственного интеллекта. Это направление связано с моделированием интеллектуальной деятельности человека в сфере обучения. В западных странах это направление получило название Intelligent Tutorial System (ITS). На рис. 2.2 приведена обобщенная схема работы интеллектуальной обучающей системы. ИОС в соответствии с программой обучения генерирует учебное задание, которое должен выполнить студент. Далее студент и система самостоятельно выполняют это задание. Затем выполненные задания сравниваются и, по результату сравнения, формируется реакция системы. Это может быть просто сообще-

ние о правильности выполнения или подробный отчет сравнения. Затем производится корректировка модели обучаемого и процесс повторяется. Процесс обучения с помощью ИОС на основе достижения определенного состояния модели обучаемого.

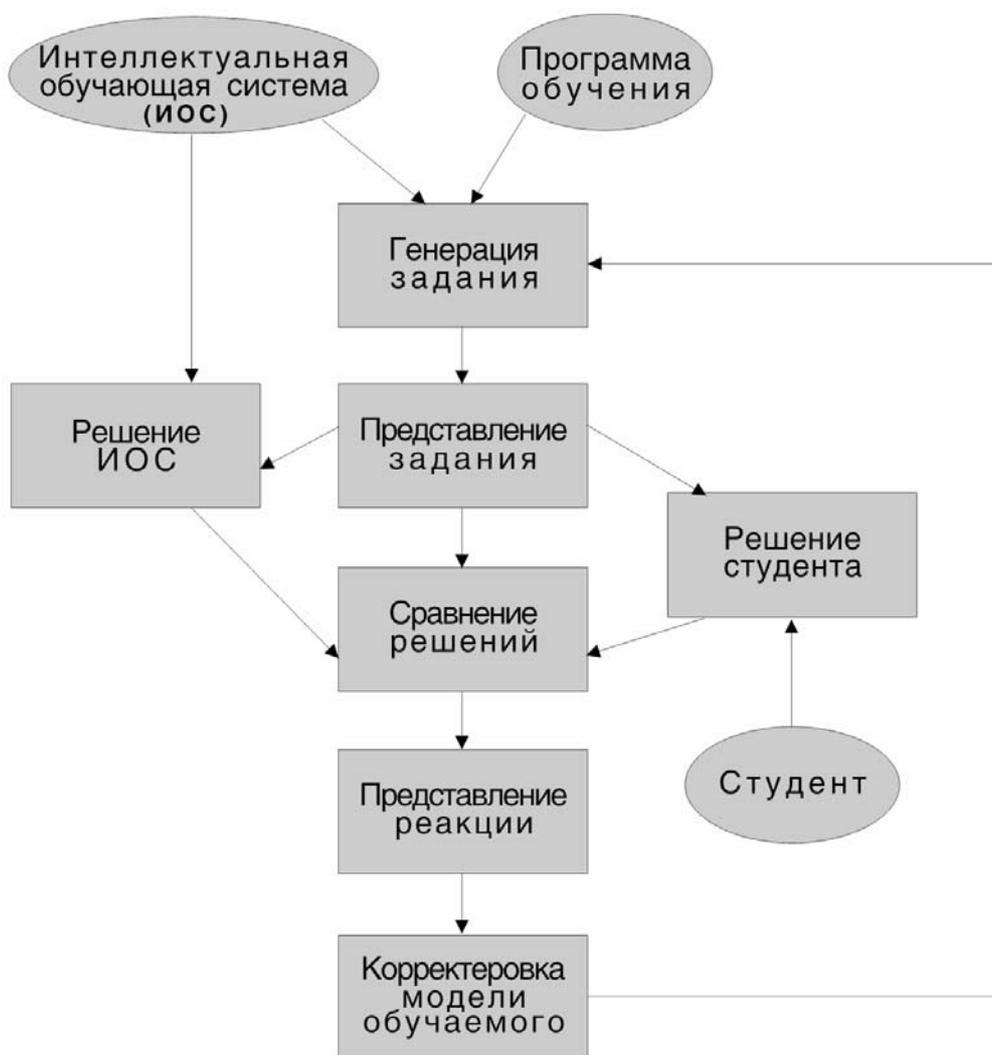


Рис. 2.2 — Обобщенная схема работы интеллектуальной обучающей системы

Для работы ИОС необходимо иметь базу знаний, в которой должны быть представлены следующие знания:

- знания из предметной области;
- знания технологий обучения;
- знания о конкретном студенте.

Одним из характерных признаков ИОС является наличие модели обучаемого. Различают три основных класса моделей: скалярная; оверлейная, пертурбационная. Скалярная модель простейшая отражающая уровень знания студента в виде некоторого числа. Оверлейная модель представлена в виде некоторого массива или структуры, каждый элемент которого отвечает за конкретный фрагмент знаний. Пертурбационная модель учитывает не только истинные знания студента, но и искажение знаний, в простейшем случае, это типичные ошибки студентов.

Важнейшим свойством ИОС является способность решать задачи и моделировать деятельность эксперта. Это свойство роднит ИОС с экспертными системами и интеллектуальными решателями. Часто интеллектуальный решатель или экспертная система является частью ИОС, которая отвечает за реализацию механизмов семантического вывода на основе базы знаний предметной области.

Обобщенная структура ИОС содержит пять модулей: модуль решения, модуль планирования, модуль студента, модуль интерфейса, генератор.

Модуль решения обеспечивает решение задания, задачи или проблемы, которая формируется генератором. Модуль планирования обеспечивает стратегию обучения для заданного студента. Он определяет программу обучения конкретного студента. Примером такого модуля служит программа *Curriculum Planner*, реализованная в ИОС *Circsim-Tutor*, предназначенную для обучения студентов медиков. Модуль студента обеспечивает поддержку представления модели студента, строит и изменяет ее в процессе обучения, сравнивает с эталонной моделью и т.д. Модуль интерфейса обеспечивает взаимодействие ИОС со студентом. Рассмотрим кратко наиболее известные ИОС.

1. Система SOPHIE, основана на использовании следующих тьюторских правил:

1) если студент делает фактическую ошибку, программа ее исправляет;

2) если студент делает ошибку, не имеющую отношения к данной теме, программа не дает подробного исправления;

3) если студент делает неправильные обобщения, программа дает контр — пример;

4) если студент старается сделать вывод слишком быстро, программа обучает его навыкам мышления.

Эта система реагирует на вопросы студентов и решения студентов по обнаружению и устранению неполадок электронных цепей. Глобальная задача системы — создание среды, в которой студенты могут исследовать свои собственные гипотезы о неполадках в цепях, система же осуществляет обратную связь, комментируя правомерность их решений, генерирует контрпримеры для демонстрации ошибок. SOPHIE имеет базу знаний о повреждениях электронных цепей и ограниченный интерпретатор естественного языка для ответа на вопросы.

2. Система WUSOR-I обучает студентов играть в игру WUMPUS, в которой участники охотятся за опасным зверем. ИОС является экспертом по оценке действий студента, предполагает построение модели его умений и навыков. Программа-помощник обеспечивает помощь в усвоении понятий путем отсылки учащегося к экспертным знаниям. Задача таких систем — ознакомить учащихся с характером принятых ими решений. Программа-помощник направляет процесс обучения, предлагая время от времени свои знания.

3. Система GUIDON, разработанная проф. Кленси как инструмент обучения экспертной системы MYCIN. Эта система использует знания экспертной системы MYCIN для генерации заданий для студентов медиков и получения диагноза. Кроме того, система GUIDON имеет модель преподавателя, которая обеспечивает помощь и консультации в интерпретации диагноза генерируемого системой MYCIN.

4. Система PROUST предназначена для студентов изучающих язык Паскаль. Она способна анализировать программы, написанные на языке Паскале и генерировать объяснения синтаксических ошибок. Система Lisp Tutor предназначена для аналогичных задач для языка программирования LISP.

#### **2.2.4. Генераторы тестовых заданий, задач и вопросов**

Генераторы — класс компьютерных учебных программ, предназначенных для формирования учебного задания или вопроса. Данный класс программ является также подклассом про-

граммных генераторов. Термин «Программный генератор» появился при создании программ моделирования сложных систем, где в качестве метода моделирования использовались методы Монте-Карло, а затем методы теории имитационного моделирования. В этом случае строились упрощенные программные модели, имитирующие как элементы исследуемой системы, так и существенные элементы среды, влияющие на поведение системы.

Одним их важнейших элементов систем имитационного моделирования являются датчики случайных чисел (ДСЧ). ДСЧ это программы, генерирующие псевдослучайные последовательности целых или вещественных чисел. Для исследования различного класса систем появились разнообразные программные генераторы, основанные на использовании ДСЧ. Примерами могут служить генераторы изображений с заданными статистическими свойствами, генераторы карт, генераторы сценариев.

Другими важными классами программных генераторов являются генераторы отчетов, широко применяемых в базах данных. Однако данный класс программ нельзя считать в чистом виде генератором, поскольку результат их работы не подается на вход системы, а служит конечным продуктом. Генераторы программ еще один класс подобных систем, который по заданному описанию на выходе получает программу на некотором языке программирования, например генератор различного рода трансляторов и интерпретаторов по описанию входного языка.

Генераторы тестовых заданий могут иметь самостоятельное значение, когда он используется преподавателем или студентом для получения некоторого учебного задания. Однако часто такой генератор является частью некоторой компьютерной учебной программ. Например, генерация задания для закрепления практических навыков или курсового проектирования. Особое значение имеет генератор для компьютерного тестирования:

- генератор может получить огромное число тестовых заданий;
- правильные ответы для сравнения со студенческим ответом, генератор явно не хранит, а получает в процессе формулировки задания.

Рассмотрим обобщенную структуру генератора тестовых заданий и вопросов, показанную на рис. 2.3.

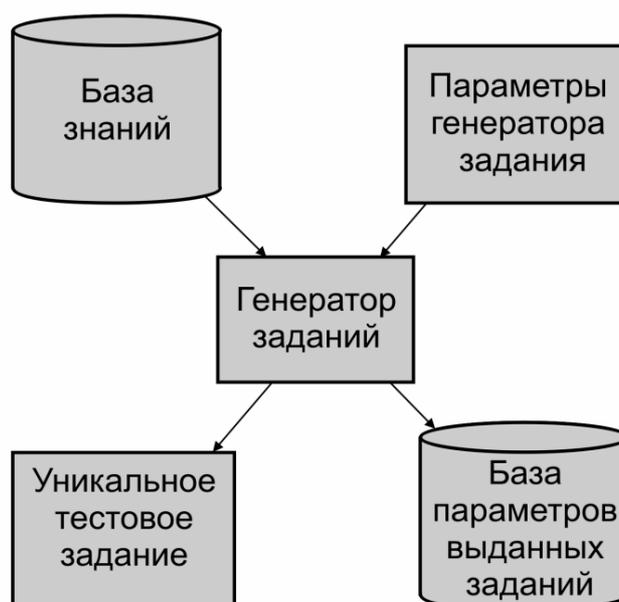


Рис. 2.3 — Обобщенная структура генератора

База знаний содержит полную информацию для генерации задания. Если задание формулируется в форме задачи, то база знаний содержит банк шаблонов формулировок задач, области изменения параметров задач, алгоритм нахождения решения или проверки наличия решения, алгоритм изменения значений параметров. Если задание представляет собой последовательность вопросов, то могут использоваться разнообразные модели и алгоритмы генерации вопросов: табличные, графовые, и т.д.

Параметры генерации задания задают начальные значения параметров генерации и ограничения на тип генерируемых заданий. Начальные значения параметров генерации могут быть получены на основании: 1) датчика случайных чисел; 2) индивидуальных параметров студента: фамилия, имя, отчество, место и дата рождения, семестр и прочее; 3) на основании некоторого унифицированного кода, присваиваемого студенту при поступлении в вуз.

Уникальное тестовое задание является конечной целью работы генератора и представляется на некотором декларативном языке для последующего этапа визуализации. Например, генераторы в системе «Фея» формируют тестовое задание на языке представления теста этой системы. Можно также представлять тестовые задания на языке Tex или XML.

База параметров выданных тестовых заданий предназначается для двух основных целей:

- Идентифицировать задание, выполненное студентом. Опыт в ТМЦДО показывает, что некоторые студенты присылают ответы или решения не на те задания, которые им были выданы.
- Способствовать выявлению ошибок в генераторах. Генераторы тестовых заданий являются сложными программами, отладка которых является достаточно трудоемким делом. Проверить работоспособность его во всех случаях практически невозможно. Поэтому желательно иметь такую базу для оперативного анализа и исправления ошибок.

Алгоритм работы генератора зависит от типа задания и моделей представления знаний. Обобщенная схема работы показана на рис. 2.4.

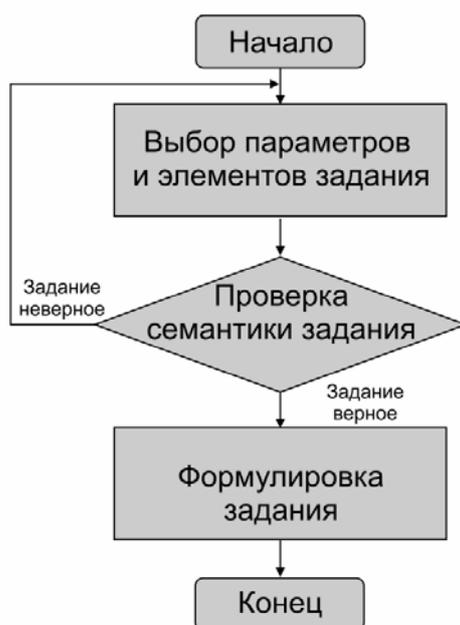


Рис. 2.4

Рассмотрим основные шаги работы алгоритма:

1. Выбор параметров и элементов задания это шаг, на котором производится синтез задания. Выбор значений параметров и элементов задания осуществляется, как правило, на основе использования датчика случайных чисел. Это в некоторых случаях существенно снижает параметры генерирующего алгоритма. Если параметры генерировать от входных параметров студента, что

необходимо подбирать хеш-функции, которые бы вычисляли значения ключей. Хорошие результаты дает использование алгоритмов нумерации, которые перечисляют все варианты заданий данного алгоритма генерации.

2. Проверка семантики задания это важный шаг, поскольку не всякая комбинация параметров может привести к решению задачи. В простейшем случае необходимо проверить некоторую формулу, например, подкоренное выражение не должно быть меньше нуля. В более сложных случаях необходимо разрабатывать специальный алгоритм семантического контроля.

3. Формулировка задания — это шаг, на котором производится преобразования внутреннего представления тестового задания на некоторый декларативный язык. Поскольку текст задания может содержать формулы, рисунки, таблицы и прочее, то нужно использовать язык описания документа для какого либо конкретного пакета.

Необходимо также отметить, что если задание разбивается на части, то приведенный обобщенный алгоритм (рис. 2.4) можно применить к каждой отдельно части, поэтому весь генератор может представлять собой многократную итерационную схему.

Перечислим основные свойства генератора:

- 1) мощность генератора;
- 2) управление последовательностью генерации;
- 3) управление сложностью тестового задания.

Мощность генератора это количество возможных вариантов заданий, которые может получить генератор. При разработке генератора нужно оценить его мощность; в некоторых случаях это довольно трудно сделать, поскольку может оказаться сложная связь между отдельными генерируемыми элементами задания.

При проведении тестирования в компьютерных классах встает задача получения тестовых заданий на двух соседних компьютерах. Здесь необходимо получить задания, которые, по крайней мере, не были похожи друг на друга. Вследствие этого встает задача установки семантического расстояния между двумя сгенерированными заданиями и управления генерацией с учетом этого расстояния.

Еще одним важным свойством генератора является возможность управления сложностью задания. В простейших случаях

генератор выдает задания одной и той же сложности. В некоторых технологиях требуется генерировать задания с разной сложностью, начиная, как правило, с простых и заканчивая самыми сложными заданиями.

Генераторы становятся одним из важных классов компьютерных учебных программ, поднимающие компьютерные технологии обучения на более высокий уровень.

### **2.2.5. Тестирующие программы**

Тестирующие программы — это класс компьютерных учебных программ, который в автоматизированном режиме производит тестирование студентов. В основе данного класса компьютерных учебных программ лежит понятие теста, как некоторого измерительного инструмента

Тестирующие программы можно разделить на следующие:

- 1) программы, предназначенные для определения уровня знаний, реализующие тесты достижений;
- 2) программы определения состояния обучаемого;
- 3) программы определения способностей обучаемого, реализующие тесты интеллекта, креативности и т.д.

Компьютерные учебные программы, реализующие тесты достижений классифицируются на:

1. Входные, проверяющие готовность к проведению обучения. Например, входное тестирование для проведения лабораторной работе.

2. Текущего контроля, для определения уровня знаний студента в процессе обучения. Например, тестовые контрольные работы в ТМЦДО.

3. Рубежного контроля, например, для проведения зачетов по курсу разделенного на два семестра.

4. Итоговые, как правило, это экзамены, проводящийся в тестовой форме.

Тесты определения состояния обучаемого необходимы для определения параметров здоровья обучаемого в данный момент времени. Например, с помощью теста определяют характеристики зрения обучаемого или утомляемости и т.д.

Тесты способностей предназначены для определения индивидуальных особенностей обучаемого (навыки, познавательные способности, особенности восприятия, креативность, интеллект). Например, тест, определяющий IQ (коэффициент интеллекта).

Тестирующая компьютерная учебная программы содержит базу вопросов, которая заранее подготовлена или получается с помощью генератора. Эти вопросы записываются в определенном формате, который зависит от инструментальной среды разработки или некоторого стандарта представления. Обобщенный алгоритм программы следующий:

- по определенному алгоритму формируется вопрос и задается студенту;
- студент вводит ответ;
- программа анализирует ответ;
- на основе анализа ставится оценка за вопрос;
- вопрос, ответ и оценка заносится в протокол;
- процесс повторяется, пока вопросы не закончатся или не выйдет время тестирования;
- на основе анализа ответов формируются некоторые выводы или определяется итоговая отметка.

Конкретных алгоритмов работы тестирующих программ огромное количество. С точки зрения реализации обобщенного алгоритма можно выделить тестирующие программы частичным выполнением, например, программа выполняет только фазу опроса, а фазу оценивания производит преподаватель и с полным выполнением.

Если формирование текущего вопроса зависит от ответов на предыдущие вопросы, то говорят об адаптивном тестировании. Стратегии адаптивного тестирования могут быть различными.

В случае, когда тестовые задания независимы, то может быть реализованы следующие варианты: задания случайно выбираются из банка заданий; задания группируются по темам и в каждой теме задания выбираются случайно; задания выдаются в некоторой заданной последовательности.

Следующий, важный с точки зрения программирования, шаг является предъявление вопроса студенту. Информационная часть вопроса может содержать текст, таблицы, графики, изображения, формулы и т.д. Здесь проблема заключается в том, что тести-

рующая программа должна иметь свою собственную систему отображения или воспользоваться уже имеющейся, например, браузером или системой отображения какого-нибудь известного редактора или издательского пакета.

Важным элементом тестирующей программы является форма ввода ответа, это может быть указание элемента меню, кнопка, ввод строки или числа и т.д. По способу ввода ответа тестирующие программы можно разделить на программы: с свободной формой ответа (когда студент вводит строку в поле ввода), с закрытой формой ответа (когда студент выбирает ответ из некоторого множества) и смешанной формой, (когда при опросе студента используются обе предыдущие формы).

Анализ ответа студента зависит от формы ввода ответа, если для закрытой формы анализ ответа является достаточно простой задачей, то для открытой формы это в общем случае задача пока неразрешимая, поскольку требует анализа текста записанном на естественном языке. В этом случае, множество возможных ответов на вопрос сужают до слова или словосочетания, тогда ответ можно анализировать на основе сравнения с эталонным ответом, записанном в программе.

После проведения опроса и анализа ответов тестирующая программа производит оценивание. Здесь используются различные подходы: оценивание с использованием теории размытых множеств и нечеткой логики, статистическая теория оценивания, оценивание на основе эвристических методов,

Важным вопросом при организации аттестации с помощью тестирующих программ является формирование протокола и решение проблемы несанкционированного доступа.

Необходимо отметить, что тестирующие программы являются одним из важнейших элементов современного образовательного процесса.

### **2.2.6. Тренажерные программы**

Тренажерные программы можно разделить на программы, предназначенные для практического закрепления учебного материала, и программы, предназначенные для тренажа операторов в

различных технических системах. Синонимом тренажерных программ можно считать термин «компьютерные практикумы».

Тренажерные программы первого типа (или их можно назвать тренировочно-закрепительными) обычно генерируют некоторую последовательность упражнений или задач, которые необходимо выполнить учащимся для закрепления учебного материала. Достоинством ЭВМ перед учебником является возможность генерации достаточно большого числа вариантов заданий. Обычно тренировочные программы осуществляют управление учебной деятельностью. Управление со стороны учащихся допускается на уровне выбора типа упражнений. Тренировочные упражнения обычно сочетаются с АУК, часто входят в состав комплекта систем обучения, реализующих разные методы. Примерами таких программ являются система ZGEOM, предназначенная для генерации и решения геометрических задач, GOLEM — Тренажер по терапии и постановки диагноза больных, находящихся в критическом состоянии, тренажер на доказательство теорем по курсу «Дифференциальное исчисление». Тренажер по методам оптимизации в учебной САП.

Тренажерные программы второго типа моделируют некоторую реальную, быстро меняющуюся обстановку, в которой необходимо управлять некоторым устройством. Например, управление автомобилем или управление некоторой технологической установкой. При этом программа должна моделировать изменение обстановки в зависимости от действий оператора и генерировать реалистичное изображение на экране компьютера. Примерами таких тренажеров являются программа TRNIMAGE, предназначенная для диспетчеров железнодорожного транспорта, тренажеры для операторов технологических установок НПЗ.

### **2.2.7. Моделирующие программы**

Компьютерное моделирование является мощным инструментом исследования сложных систем. Компьютерное моделирование в обучении имеет два аспекта:

- как содержание, которое должно быть усвоено студентом в процессе обучения, как способ познания, которым они должны овладеть;

- как одно из основных учебных действий, которое является составным элементом учебной деятельности.

С одной стороны студент должен освоить методологию моделирования в данной предметной области. С другой стороны освоить содержание предмета, который изучает с помощью компьютерного моделирования.

Еще одним важным аспектом компьютерного моделирования с точки зрения обучения является изучение процесс и явлений, которые трудно изучать в реальной жизни, например, протяженные во времени или в пространстве, опасные или дорогостоящие процессы (ядерный реактор, превращение куколки в бабочку и т.д.). В этой упрощенной среде студент решает задачи, наблюдает и вмешивается в процесс моделирования, приходит к пониманию особенностей изучаемого явления или объекта, учится управлять им и т.д.

Моделирующие программы отличаются от компьютерных учебников и тренажеров тем, что в них студент обучается в результате самостоятельной деятельности в условиях, приближенных к условиям реальной действительности. Обычно моделирующие программы используют при проведении лабораторных работ, когда вместо лабораторной установки для изучения процесса или явления реализуется моделирующая программа.

Особенно много такого рода программ разработано для проведения лабораторных работ по физике. Другим примером таких программ является программа для проведения лабораторных работ по курсу «Теория автоматического управления». Эти программы требуют от студента ввода типа модели, далее ввода параметров модели и тип проведения исследований, проводит моделирование и выводит графики на экран дисплея.

Имеются также моделирующие программы по химии такие как «Химия для всех», «1С Репетитор.Химия», «Chem3DPro», «ChemLand». В таких программах на экране дисплея задаются реактивы и неизвестное вещество в пробирке, необходимо определить это вещество. Обучаемый может производить химические опыты (самы различные), а программы будет моделировать результаты этих опытов.

В настоящее время в учебном процессе стали широко использовать системы моделирования, которые обеспечивают:

- 1) построение модели;
- 2) анализ модели;
- 3) проведение моделирования
- 4) обработка результатов моделирования;
- 5) визуализация результатов моделирования.

Примерами таких систем являются программы моделирования электронных схем WorkBanch Electronics, ASIMEK.

К преимуществам моделирующих программ относятся следующие:

- 1) усиление мотивации обучения;
- 2) возможность переноса приобретенных знаний, умений и навыков в реальную жизнь;
- 3) эффективность обучения;
- 4) надежность, безопасность и удобство;
- 5) управляемость со стороны студентов;
- 6) доступность;
- 7) возможность последовательного изучения объекта от упрощенной формы до более высокой степени детализации.

Особое внимание в последнее время уделяется организации лабораторных работ на реальных установках с удаленным доступом и использованию устройств виртуальной реальности в обучении.

### **2.2.8. Компьютерные учебные игры**

Компьютерные учебные игры — класс компьютерных учебных программ, реализующих процесс обучения в игровой форме. Известно, что игры позволяют реализовать активные методы обучения. Учебные игры очень близки к имитационным программам (наличие среды участников и правил определения результатов). Задача учебных игр — создание среды, облегчающей усвоение обучаемым знаний и приобретение ими умений и навыков. Игры не обязательно имитируют реальную жизнь, но они обязательно должны быть занимательны. Основное отличие игр от программ моделирования — состязательность, наличие противника (человека или программы). Задача учебной игры — привить навык, задача участника — выиграть.

### 2.2.9. Прикладные программы, используемые при обучении

По мере расширения сферы использования ЭВМ в обучении учебные заведения начали применять прикладные программы общего назначения, названные инструментальными средствами студентов. Компьютер может использоваться как счетно-решающее устройство для проведения расчетов, приводить в систему данные, полученные в результате лабораторных работ. Студенты могут учиться основам управления роботами и аппаратами, составляя соответствующие программы. Компьютеры могут использоваться для хранения и поиска информации, необходимой им в процессе обучения. К прикладным программам, используемым учащимися, относятся редакторы, базы данных и файловые системы, электронные динамические системы. Программы редактирования используются в учебных заведениях следующим образом: управление курсором на ранних стадиях обучения; редактирование документов по образцу; использование редактора как системы хранения данных (студенты вводят свои списки слов, правописание которых вызывает у них затруднения); обучение последовательному изложению мыслей; составление докладов, писем, сообщений, поздравлений, благодарностей, жалоб и т.д.

#### *Программы управления процессом обучения*

Программы управления процессом обучения (УПО) используются в качестве администратора индивидуальной программы обучения и предназначены для эффективного использования ресурсов учебного заведения. Основные функции программ УПО следующие:

1. Контроль — система должна оценивать достижения обучаемого.
2. Выдача предписаний — система должна координировать назначение очередных учебных мероприятий.
3. Содействие усвоению материала — система должна предоставить обучаемому обратную связь, с помощью которой он может узнать о качестве усвоения им учебного материала.
4. Оценивание результатов обучения — система должна предоставить обучаемому и преподавателю текущую информа-

цию об успеваемости обучаемого. Кроме этого, сведения о прошлых успехах обучаемого могут быть сопоставлены с текущей успеваемостью с целью правильного выбора очередной учебной процедуры.

5. Сбор данных и составление отчетов — система должна обеспечить хранение и выдачу данных: об отдельных учениках, об отдельных группах, о мероприятиях по индивидуальному и групповому обучению, об эффективности предпринимаемых действий обучаемого.

6. Распределение ресурсов и учебных помещений, составление расписаний занятий и другая информация о ресурсах.

7. Информационное обеспечение и взаимодействие через электронную почту.

Дальнейшее развитие этого направления связано с созданием систем управления учебным процессом (СУУП) в среде Интернет. Однако данный класс систем относится к классу интегрированных систем.

#### *Программы поддержки текущей деятельности преподавателя*

Программы поддержки текущей деятельности преподавателя необходимы для автоматизации рутинной деятельности преподавателя. К ней относится ведение дневников и журналов, составление отчетов и расписаний и т.д. Примером может служить программа, разработанная в институте программных средств обучения РАО (г. Новосибирск) и предназначенная для составления школьных расписаний. Эта программа может планировать учебный процесс, представленный в виде таблицы расписаний на 1 неделю. При этом необходимо выполнить следующие операции: ввод программ-планов по предметам для каждого класса с указанием тем уроков, формирование списка преподавателей, ввод списка помещений школы, ввод списка учебных предметов, разметка (ввод частично заполненной таблицы под расписание).

Другим примером является программный комплекс, предназначенный для расчета часов по кафедре, распределения нагрузки профессорско-преподавательского состава и формирования на их основе рабочего учебного плана.

### 2.2.10. Инструментальные системы

Под инструментальными системами будем понимать педагогические программные средства, которые позволяют автоматизировать процесс разработки компьютерных учебных программ. В некоторых публикациях используется термин «авторская система»[1]. Под авторской системой будем понимать систему, предназначенную для разработки некоторых классов компьютерных учебных программ.

Инструментальные системы для разработки КУП можно разделить на следующие классы:

- 1) универсальные языки программирования;
- 2) языки систем искусственного интеллекта;
- 3) специальные системы для разработки определенного класса КУП;
- 4) языки для создания КУП;
- 5) языки имитационного моделирования;
- 6) гибридные системы.

Универсальные языки программирования, такие как Си++, Паскаль, Basic, VisualBasic, Java, довольно часто используются при создании ППС. Однако для создания качественного продукта они требуют значительных усилий квалифицированных программистов.

Для создания интеллектуальных КУП, основанных на идеях экспертных систем и разнообразных решателей, необходимо использовать соответствующий инструментарий. Это могут быть языки ИИ — Лисп, Пролог, гибридные экспертные системы и т.д. Кроме этого также можно воспользоваться универсальными языками программирования. Однако использование этих средств также сопряжено с достаточно большими затратами материальных и трудовых ресурсов.

В настоящее время широкое распространение получили специальные инструментальные системы, позволяющие существенно сократить сроки и затраты на разработку определенного класса КУП. Это отечественные системы: «Урок», «Сценарий», «Адонис», «Аосмикро», «Слоистая машина», «Наставник», «Тамас», «Учитель и ученик», «Симтех», «Кадис», «Costoc», «Кобра», «SHELP», данные в обзоре. Рассмотрим некоторые из них.

Инструментальная система «Урок» — универсальный редактор обучающих курсов, предназначен для создания компьютерных учебников. В «Урок» входят:

1. Графическая система SPEN, состоящая из развитого графического редактора, редактора шрифтов и базы графических изображений.

2. Редактор динамических модулей, предназначенный для организации работы обучаемого с моделями явлений и технологических систем.

3. Модуль организации сценария обучения.

4. Редактор эталонов системы контроля знаний.

5. Текстовый редактор.

6. Редактор справочной информации.

7. Программы сервисного обслуживания.

8. Система обучаемого — автономный модуль, предназначенный для запуска обучающих программ.

Система «Урок» реализована для ОС MS-DOS.

Следующая система — AOSMICRO. Разработана в Московском техническом университете и состоит:

1) из графического редактора PICMAKE, который обеспечивает представление учебной информации (создает статические и динамические изображения, а также использует звук);

2) редактора сценариев, который в режиме меню:

а) формирует кадры, состоящие из некоторого множества команд;

б) осуществляет ввод эталонов для анализа ответов;

в) производит тестирование и отладку;

3) исполнителя, обеспечивающего выполнение ППС, разработанного данной системой.

Система AOSMICRO реализована в среде MS-DOS.

В настоящее время разработана новая версия системы AOSMICRO-W, предназначенная для работы под ОС WINDOWS

Эта версия системы ориентирована на три уровня подготовки:

1. Первый уровень предназначен для пользователей, знающих язык программирования ТурбоПаскаль. Здесь используются библиотека функций системы: анализ ответов, чтение изображений редактора PICMAKE и т.д.

2. Второй уровень предназначен для пользователей, знающих диалоговый язык системы.

3. Третий уровень предполагает использование готовых типовых блоков.

Система «Сценарий» предназначена для разработки КУП с использованием элементов визуального программирования[14]. Основная идея системы заключается в построении КУП, представленной некоторой блок-схемой. Блок-схема КУП строится из некоторого множества структурных единиц, следующих типов:

- информационный (представление учебной информации);
- принимающий (ввод ответа обучаемым);
- анализирующий (анализ ответа);
- специальный (запись в файл, вызов программ и т.д.).

Система состоит из набора редакторов, модуля проверки функционирования и исполнения, модуля помощи и программы начальной установки.

Система «Сценарий» реализована для ОС MS-DOS. Дальнейшее развитие этой системы связано с переходом на ОС Windows.

Система Strutum Computer(SC), или «Слоистый компьютер», разработана для моделирования различного класса систем. Система SC используется для автоматизации построения компьютерных учебников и тренажеров. Система состоит из трех слоев: визуального(VS), математического(MS) и алгоритмического(AS).

Система КАДИС (система Комплексов Автоматизированных Дидактических Средств) разработана и развивается в Самарском государственном аэрокосмическом университете с начала 90-х годов.

Эта система обеспечивает представление теоретического материала в виде гипертекста, построение тренажеров и контроля знаний.

Система «Ракель» разработана в Московском авиационном институте и предназначена для реализации компьютерной технологии обучения на основе использования экспертных систем. Система состоит:

1. Из базы знаний, которая наполняется в зависимости от учебного курса.
2. Модуля генерации учебного задания.

3. Модуля управления, который обеспечивает переход от одного этапа обучения к другому, в зависимости от уровня знаний обучаемого.

4. Модуля логического вывода, который обеспечивает оценивание текущего состояния процесса обучения.

5. Модуля приобретения знаний, который на основе диалога с преподавателем строит базу знаний по конкретному курсу.

В основе базы знаний системы «Ракель» используется фреймовая модель.

Система «I Know» разработана в центре внедрения медицинских технологий «ЦЕВМА» г. Иркутска и предназначена для создания различных КУП контроля знаний. Основные функции системы следующие:

1. Запись и корректировка тестовых заданий в базу данных.
2. Сортировка тестовых заданий, записанных в базу данных.
3. Получение экспресс-информации по базе данных.
4. Статистическая обработка результатов контроля по темам и группам.

5. Проведение тестирования на основе разработанной базы данных.

6. Гибкая настройка параметров контроля в зависимости от потребностей преподавателя.

В Томском университете систем управления и радиоэлектроники разработано четыре инструментальных системы создания КУП. Это системы Фея, Sidney, EduCAD, система разработки конспекта лекций и компьютерных тренажеров для естественно научных дисциплин.

Система Фея имеет три версии реализации: первая версия предназначена для создания кадровых КУП различного назначения для операционной системы DOS, вторая версия предназначена для разработки КУП для системы Windows, третья — ориентирована на создание тестирующих программ с использованием генераторов тестовых заданий.

Система Sydney предназначена для разработки различного класса КУП и обучения студентов с помощью данных КУП в локальном компьютерном классе. Достоинством данной системы является возможность визуальной разработки виртуальных лабораторных работ с использованием трехмерной графики.

Система EduCAD предназначена для разработки мультимедийных учебников. Отличительной особенностью данной системы является возможность представления учебного материала в аудио и видео форматах.

Система разработки конспекта лекций и тренажеров предназначена для создания тренажерных КУП для обучения решению задач. Особенностью данной ИС, является язык, описывающий КУП как конечный автомат, моделирующий деятельность студента при решении задачи.

Можно также отметить систему TeachPro для создания электронных учебников и автоматизированную инструментальную систему тестирования АИСТЕСТ, разработанную в Тульский педагогическом университете.

Для разработки электронных учебников используются различные системы, начиная с издательских, такие как Microsoft Word, Latex, Adobe Acrobat.

Из зарубежных ИС можно отметить следующие: LinkWay, TeachCAD, MediaVerse 1.5, Toolbook 4.0, Quest 5.1. Dreamweaver.

### **2.2.11. Интегрированные системы**

Интегрированные системы представляют собой программные комплексы, состоящие из нескольких типов компьютерных учебных программ. Например, в рамках некоторого учебного курса может быть разработано целое семейство компьютерных учебных программ. Например, такой комплекс может содержать: компьютерный учебник, некоторый экзаменатор, несколько тренажеров для тем, требующих особого внимания, программы моделирования для проведения компьютерных лабораторных работ, компьютерные справочники по теме и т.д.

Синонимом термину «интегрированные системы» может служить термин «программно-методический комплекс (ПМК)». Однако для ПМК характерно кроме программного обеспечения наличие обычной методической литературы. Это учебники и учебные пособия, описания лабораторных работ и практикумы и т.д. Примером такого подхода может служить программно-методический комплекс по иностранному языку для начинающих, разработанный в Московском государственном инженерно-

физическом институте [36]. Другим примером интегрированной системы является ПМК по микропроцессорной технике, разработанный в Томском университете систем управления и радиоэлектроники [37].

С развитием технологий обучения в среде Интернет появляются сетевые программно-методические комплексы. Одним из таких является сетевой учебно-методический информационный комплекс (СУМИК). Этот комплекс кроме учебно-методического материала (учебного пособия, руководства по изучению дисциплины, тестов, практикумов и т.п.) включает в себя возможность дистанционного взаимодействия между участниками учебного процесса (электронную почту, форумы).

Еще одним классом интегрированных систем обучения являются системы дистанционного обучения или LMS (Learning Management System).

### **2.3. Программное обеспечение дистанционного обучения**

Внедрение и развитие дистанционных технологий обучения становится определяющим фактором развития образования в мире. Для России с ее огромными территориями это имеет первостепенное значение. В настоящее время дистанционные технологии обучения используются практически каждым вузом России. Признанными лидерами и первопроходцами здесь являются: Московский экономико-статистический университет (МЭСИ), Современный гуманитарный университет (СГУ), Международный институт менеджмента ЛИНК. В области инженерного образования можно выделить Томский государственный университет систем управления и радиоэлектроники. Московский университет электроники и математики (МИЭМ).

В методике применения дистанционных образовательных технологий, утвержденной приказом Минобразования России № 4452 от 18.12.02 определяется, что дистанционное обучение есть технология обучения, при которой целенаправленное опосредованное или не полностью опосредованное взаимодействие обучающегося и преподавателя осуществляется на основе информационных технологий.

Решением коллегии Минобразования РФ № 16/1 от 26.06.2002 «Об итогах эксперимента в области дистанционного обучения и перспективах развития дистанционных образовательных технологий» определено три основные дистанционные образовательные технологии: кейс-технологии; интернет-технологии; технологии спутникового телевидения. Допускается сочетание основных видов технологий. Выбор технологии обучения зависит от возможностей и предпочтений вуза. Самой простой с точки зрения затрат на разработку, является кейс-технология, которая предполагает формирование пакета учебник и учебно-методических пособий и доставку их студенту по месту проживания. Общение с вузом может осуществляться по обычной или электронной почте. Прием экзамена или зачета обычно осуществляется в представительстве данного вуза в регионе.

Технологии спутникового телевидения также являются средством эффективного обучения, однако требуют наличие телевизионных каналов, что является достаточно затратным. Эта технология окупается, если контингент вуза достаточно велик. Например, в СГУ где учится более 200 тысяч студентов.

Интернет-технология наиболее затратная, поскольку требует существенной переработки всего методического материала и изменения парадигмы преподавания. В этой технологии общение между студентом и преподавателем опосредовано и осуществляется через компьютерную сеть. Важнейшим элементом данной технологии становится программное обеспечение и в частности педагогические программные средства.

Практика дистанционного обучения показывает, что реально в вузе используется технология, основанная на сочетании указанных видов. Важным требованием для студентов является наличие персонального компьютера с доступом, по крайней мере, к электронной почте. Например, в студенческий пакет кроме учебных пособий вкладываются компакт-диски с компьютерными учебниками и тренажерами и т.д. Студент, используя интерактивные мультимедийные учебники, изучает теорию, с помощью тренажерных программ осуществляет практические занятия, с помощью моделирующих программ выполняет лабораторные работы, запуская тестирующие программы, осуществляет самокон-

троль и если протоколы выполнения тестовых заданий высылаются в вуз, то вуз может осуществлять промежуточный контроль.

Такая стратегия позволяет:

1) проводить политику эволюционного развития технологии дистанционного обучения;

2) эффективно использовать ресурсы вуза для развития дистанционной технологии обучения;

3) использовать разнообразные виды технологий дистанционного обучения в зависимости от развития информационных сетей того или иного региона.

4) использовать различные инструментальные средства и системы для разработки педагогических программных средств.

5) использовать разработки других вузов и производителей программного обеспечения.

Для реализации данной технологии дистанционного обучения вузу необходимо иметь:

1. Библиотеку печатных и электронных учебников и учебно-методических пособий.

2. Ресурсный центр компьютерных учебно-методических комплексов.

3. Компьютерную систему контроля знаний.

4. Информационную систему электронного документооборота.

5. Систему мониторинга за процессом обучения.

6. Учебный портал.

При современной оснащенности и доступности персональных компьютеров создать электронную версию учебника с помощью систем Microsoft Word или Latex не представляет большой сложности. Поэтому через небольшой промежуток времени организуется соответствующая библиотека. Причем учебное пособие может храниться в двух вариантах: печатном и электронном.

Ресурсный центр компьютерных учебно-методических комплексов наиболее сложный, затратный и длительный по времени создания элемент дистанционной технологии обучения. Для каждой дисциплины или курса необходимо разработать: компьютерный учебник, тренажеры, моделирующие и тестирующие программы, возможно, все это объединить в интеллектуальную обучающую систему. Причем необходимо учитывать, что для орга-

низации дистанционного обучения по одной специальности необходимо представить приблизительно 60—65 дисциплин общим объемом более 8000 часов.

При решении данной задачи необходимо использовать инструментальные системы и опыт, накопленный на кафедрах вуза при создании ППС.

Компьютерная система контроля знаний является необходимым элементом дистанционной технологии обучения и включает:

- инструментальную систему построения тестов;
- систему проведения входного тестирования;
- систему проведения промежуточного контроля;
- аттестационную систему;
- систему мероприятий направленную на защиту системы тестирования от несанкционированного доступа.

Информационная система должна обеспечить электронный документооборот вуза по организации учебного процесса по дистанционной технологии обучения. Как правило, она включает следующие функции:

Более подробно ИС должна обеспечивать:

- автоматизацию существующего документооборота связанного с набором студентов и организацией учебного процесса;
- автоматизацию технологических процессов связанных с организацией и администрированием учебного процесса;
- организацию хранения всей информации связанной с деятельностью вуза в процессе набора и обеспечения учебного процесса студентов, а так же авторизированного доступа к этой информации всех сотрудников, студентов и преподавателей;
- автоматизацию внутренних бизнес процессов вуза (внутренние распоряжения, приказы, кадровые вопросы, и т.д.).

Разработка информационной системы для вуза является довольно сложной и трудоемкой научно-технической задачей.

Система мониторинга за учебным процессом является необходимым элементом для организации учебного процесса по дистанционной технологии обучения. Эта служба может быть организована различными способами: если общение преподавателя и студента осуществляется посредством электронной почты, то эта служба может контролировать поток писем; если используется более формализованный инструмент, то процесс мониторинга

можно осуществлять, используя соответствующие возможности информационной системы.

Сеть Интернет с каждым годом становится все мощнее и сложнее в том числе и в России. Практически все вузы имеют возможности выхода в Интернет. Поэтому портал вуза ставится важным инструментом в технологии дистанционного обучения. Основные функции учебного портала:

- Информационная. Осуществлять информационную поддержку целевой аудитории сайта (основная подзадача здесь это оперативное предоставление студентам информации о реальном состоянии их процесса обучения).

- Технологическая. Предоставление целевой аудитории сайта функциональных возможностей необходимых для решения их конкретных задач (основная подзадача здесь это функциональная поддержка процесса обучения студентов, поддержание обратной связи со студентами).

- Маркетинговая. Привлечение новых студентов для обучения в вузе (продвижение услуг центра в Интернет, проведение маркетинговых исследований и рекламных компаний).

- Обучающая. Предоставление студентам интернет-услуг по обучению в режиме реального времени (предоставление сетевых курсов, проведение on-line экзаменов, виртуальное библиотечное обслуживание, online-лекции, слайд-лекции, виртуальные лабораторные работы, проведение семинаров и консультаций с использованием чатов и форумов).

Объединение перечисленных систем в одну единую интегрированную систему приводит к понятию Learning Management System(LMS). В отечественной литературе используются термины: система организации дистанционного обучения (СДО), система управления процессом обучения (СУУП), интегрированные средства разработки и использования сетевых курсов (ИСРИСК).

### 3. НАУЧНЫЙ ПОИСК В ИНТЕРНЕТ

**Интернет** — всемирная паутина — всемирная информационная компьютерная сеть, в которой информация представлена на языке HTML. Вся сеть состоит из WWW серверов, на которых хранятся HTML-страницы. Все компьютеры, подключенные к Интернет, имеют специальные физические адреса (IP-адрес). Все www-сервера имеют еще доменные имена, например, [www.tusur.ru](http://www.tusur.ru). В настоящее время в Интернет имеется миллионы серверов, на которых хранятся миллиарды страниц.

#### 3.1. Универсальная поисковая система

Для организации контекстного поиска информации в Интернет существуют специальные www сервера, называемые поисковиками. Основные компоненты поисковиков следующие:

**1. Паук** — программа, предназначенная для скачивания HTML страниц с WWW-серверов.

**2. «Гусеничный трактор»** — программа, которая обрабатывает найденные ссылки и ведет паука по ссылкам.

**3. Indexer** — программа занимается анализом текста скаченных HTML-страниц и производит индексацию.

**4. Database** — база данных всех найденных и обработанных html страниц.

**5. Поисковая машина** — именно данная программа решает какая страница соответствует введенному запросу, в каком порядке должны быть отсортированы html страницы. Все расчеты происходят исходя из внутреннего алгоритма, с которым и имеет дело оптимизатор.

Для поиска с большим количеством данных разработали **алгоритм поиска** обратных индексов или предварительного индексирования. При данном алгоритме поиска, поисковая система очищает html страницу от всего «мусора» и превращает ее в текстовый документ, помечая все найденные слова множеством значений. Структура документа приблизительно имеет вид содержания книги, где указан основной раздел и номер страницы или ос-

новой список слов с номерами страниц где они находятся. И так задача любой поисковой системы это поиск html страниц в своей базе их сортировка и выдача в порядке релевантности.

Основные поисковые системы и их использование распределялось следующим образом:

- Google — 83,87 %;
- Yahoo! — 6,20 %;
- Baidu — 4,22 %;
- Bing — 3,69 %;
- Yandex — 1,7 %;
- Ask — 0,57 %;
- AOL — 0,36 %.

### **3.2. Высшая аттестационная комиссия**

Высшая аттестационная комиссия (ВАК) является важным элементом системы аттестации научных кадров. Ниже представлены выдержки из Положения о высшей аттестационной комиссии при министерстве образования и науки российской федерации:

ВАК при Министерстве образования и науки Российской Федерации (далее — Комиссия) создана в целях обеспечения государственной аттестации научных и научно-педагогических работников.

ВАК осуществляет следующие функции:

а) дает заключения Министерству образования и науки Российской Федерации:

- на создание советов по защите диссертаций на соискание ученой степени кандидата наук, на соискание ученой степени доктора наук (далее — диссертационные советы), установление и изменение состава этих советов, установление полномочий советов, приостановление, возобновление и прекращение деятельности таких советов;

- о результатах экспертизы диссертаций соискателей ученой степени доктора наук;

- о присвоении ученых званий профессора по специальности и доцента по специальности, профессора по кафедре и доцента по кафедре;

- о представлении к защите на соискание ученой степени доктора наук, диссертации на соискание ученой степени кандидата наук по ходатайству диссертационного совета;

- о признании и установлении эквивалентности документов иностранных государств об ученых степенях и ученых званиях на территории Российской Федерации;

- по апелляциям, поданным на решения диссертационных советов по вопросам присуждения, лишения (восстановления) ученых степеней, выдачи дипломов доктора наук, кандидата наук, присвоения, лишения (восстановления) ученых званий профессора по специальности и доцента по специальности, профессора по кафедре и доцента по кафедре, признания и установления эквивалентности документов иностранных государств об ученых степенях и ученых званиях на территории Российской Федерации;

б) дает рекомендации Министерству образования и науки Российской Федерации:

- о перечне кандидатских экзаменов;

- о перечне рецензируемых научных журналов и изданий для опубликования основных научных результатов диссертаций;

- о программах кандидатских экзаменов;

в) принимает решения о продлении сроков проведения экспертизы аттестационных дел и диссертаций на соискание ученой степени доктора наук экспертными советами;

г) проводит анализ аттестационных дел, представляет заинтересованным федеральным органам государственной власти и организациям, в которые представляется обязательный экземпляр диссертации, соответствующую информацию;

д) участвует в разработке проектов актов по вопросам присуждения ученых степеней и присвоения ученых званий;

е) по поручениям Министерства образования и науки Российской Федерации:

- дает рекомендации по вопросам установления требований к обязательному минимуму содержания основных образователь-

ных программ послевузовского профессионального образования, номенклатуры специальностей научных работников;

- проверяет деятельность диссертационных советов.

Сайт ВАК <http://vak.ed.gov.ru>

На сайте ВАК имеется:

- 1) нормативные документы (Положение о ВАК, положение о диссертационном совете и др.);
- 2) решения президиума ВАК;
- 3) объявление о защите докторских диссертаций с авторефератами;
- 4) справочная информация (Перечень ведущих научных журналов, номенклатура специальностей, паспорта специальностей и др.).

### **3.3. Федеральное государственное бюджетное учреждение науки Всероссийский институт научной и технической информации Российской академии наук**

**ВИНИТИ РАН** — крупнейший информационный центр, обеспечивающий с 1952 г. российское и мировое сообщество научно-технической информацией по проблемам точных, естественных и технических наук <http://www2.viniti.ru>

Основные направления деятельности ВИНИТИ РАН:

- научно-информационное и аналитическое обеспечение исследований по естественным и техническим наукам в области науки, национальной экономики, образования, а также федеральных и региональных программ и проектов РФ;
- развитие методологии информатики, информатизация общества;
- создание интегрированных и интеллектуальных информационно-телекоммуникационных систем в области естественных и технических наук с возможностью мониторинга и навигации в рамках национальных и международных сетей.

РЖ — периодическое информационное издание, в котором публикуются рефераты, аннотации и библиографические описа-

ния публикаций в области естественных, точных и технических наук, экономики и медицины.

Ежегодно обрабатывается около миллиона научных публикаций из российских и зарубежных источников. Ежемесячно издается 232 выпуска РЖ, посвященных различным отраслям науки и техники, а также некоторым межотраслевым проблемам.

РЖ состоит из сводных томов, в которые входят выпуски, издающиеся также самостоятельно, и из отдельных, не входящих в сводные тома. Сводные тома и отдельные выпуски РЖ имеют авторские и предметные годовые указатели. Некоторые сводные тома имеют специализированные указатели (патентный, формульный, фактографический, символный).

### **3.4. Федеральное государственное научное учреждение «Центр информационных технологий и систем органов исполнительной власти»**

ЦИТиС осуществляет формирование и поддержку национального библиотечно-информационного фонда Российской Федерации в части открытых неопубликованных источников научной и технической информации — отчётов о НИОКР, кандидатских и докторских диссертаций, описаний алгоритмов и программ. В ЦИТиС функционирует комплекс, состоящий:

- из автоматизированной системы информации по науке и технике (АСИНИТ) для выполнения задач по комплектованию обязательного экземпляра неопубликованных документов, его государственной регистрации и учёту, выпуску информационных изданий и информированию в Интернете о нём, обеспечению его постоянного хранения и использования;

- единого реестра результатов научно-технической деятельности (ЕР РНТД) с утверждёнными формами учёта РНТД;

- информационно-аналитической системы регистрации и учёта НИОКР, выполненных ФГУП и ОАО, контрольный пакет акций которых находится в федеральной собственности, за счёт собственных средств (ИАС НИОКР, выполненных за счёт собственных средств).

<http://www.rntd.citis.ru/>

Статистика федерального фонда (1982—2012 г.)	Всего	2009	2010	2011	2012
Информационные карты диссертаций	671835	28260	24700	24800	4000
Информационные карты НИР и ОКР	1307811	12590	15300	19600	3500
Регистрационные карты НИР и ОКР	1156862	23283	21100	31000	1800
Информационные карты алгоритмов и программ (с 1996 г.)	17079	1242	2020	1101	260
Объекты учета РНТД (с 2007 г.)	9735	1654	1292	3844	106

### 3.5. arXiv.org

**arXiv.org** — крупнейший бесплатный *архив* электронных публикаций научных статей и их *препринтов* по *физике, математике, астрономии, информатике* и *биологии*.

Большинство публикаций архива доступно в исходном виде в формате *TeX*, но можно также скачать автоматически генерирующиеся документы в форматах *PostScript* и *PDF*.

Существует возможность оформить *e-mail-подписку* на список новых статей с их аннотациями. Можно подписаться либо на все статьи, либо на статьи только по интересующей тематике, например: вычислительная геометрия, дискретная математика и др.

При добавлении в архив публикация автоматически добавляется в базу цитирования *Citebase*. Это позволяет оценить *индекс цитирования*, то есть формальный признак значимости статьи.

**Препринт** (реже **предпубликация**) — научное издание (обычно небольшого объёма), посвящённое какой-либо теме, с которой автор хочет ознакомить заинтересованных лиц и специалистов (для обсуждения и/или уточнения полученных результатов работы), выпускаемое в свет до публикации статьи в рецензируемом *научном журнале* или до выхода полноценной *монографии*.

Как правило, препринты не рецензируются перед выходом в свет, поэтому они могут содержать ошибки и поэтому зачастую не учитываются в отчётах в качестве публикаций.

Электронный препринт (препринт, размещаемый в *Интернете*) иногда называют **е-принтом**.

### 3.6. Специализированная научная поисковая система Scirus

Специализированная научная поисковая система Scirus предназначена для поиска в интернет страниц, относящихся к научной деятельности. В настоящее время в базе Scirus содержится информация о более чем 440 000 000.

#### *Точное определение научной информации*

Scirus имеет широкий спектр специальных функций, которые помогут вам определить научную информацию, что вам нужно. С Scirus, вы можете осуществлять:

- Выбор поиска в различных предметных областях, включая здравоохранение, жизнь, физическую и социальных наук.
- Ограниченный поиск по определенному автору, журнал или статью.
- Ограничение на результаты поиска в указанный диапазон дат.
- Поиск научные конференции, тезисы и патентов.
- Настройки и сохранение поисковых запросов и результатов поиска.

Система Scirus имеет доступ и индексирует следующие специализированные интернет источники:

- 496 000 статей из *Американского физического общества*;
- 739 000 электронных статей с *ArXiv.org*;
- 91 000 полнотекстовых статей из *BioMed Central*;
- 42 300 документов из *Калифорнийского технологического института Coda*;
- 3500 электронных статей *Cogprints*;
- 104 000 полнотекстовых статей из *журналов кристаллографии Интернет*;
- 85 100 документов из *КУРАТОР*;
- 5 100 000 документов из *цифровых архивов*;
- 86 000 документов из *Diva*;
- 114 000 полнотекстовых статей из *проекта Евклида*;
- 47 500 полнотекстовых статей из *Hindawi издательская корпорация*;
- 6800 полнотекстовых документов *HKUST Институциональные репозитории*;

- 90 200 полнотекстовых документов из *Университета Гонконга*;
- 38 800 полнотекстовых документов, доступных из *IISc*;
- 24 600 полнотекстовых документов, доступных из *Университета Гумбольдта*;
- 370 000 полнотекстовых статей из *ВГД Издательский*;
- 24900000 патентных записей от *LexisNexis*;
- 1100000 полнотекстовых статей из *Lippincott Williams & Wilkins*;
- 35 300 полнотекстовых статей из *Издательский Мэни*;
- 348 000 полнотекстовых документов *MD Обратитесь*;
- 743 000 полнотекстовых статей из *Nature Group Publishing*;
- 21 000 000 *Medline* цитаты через PubMed;
- 77 800 полнотекстовых документов из *MIT OpenCourseWare*;
- 25 700 технических докладов от *NASA*;
- 2 100 000 полнотекстовых диссертаций по *NDLTD*;
- 21 000 полнотекстовых документов *OncologySTAT*;
- 10 400 полнотекстовых документов из *органических Eprints*;
- 2700 полнотекстовых документов *PsyDok*;
- 2 100 000 статей *PubMed Central*;
- 1 100 000 документов *RePEc*;
- 68 900 полнотекстовых статей из *Издательский Королевского общества*;
- 325 000 полнотекстовых статей из *PKK Издательский*;
- 766 000 полнотекстовых статей из *Sage Publications*;
- 11 000 000 полнотекстовых статей из *ScienceDirect*;
- 538 000 полнотекстовых статей из *Scitation*;
- 9100 статей из the Society for Industrial and Applied Mathematics *SIAM*;
- 3 500 000 статей из *Springer*;
- 32 200 полнотекстовых документов из Университета Торонто *T-Space*;
- 165 200 полнотекстовых документов *The Wageningen UR Publications database*;
- 2 200 000 статей из *Wiley-Blackwell (Blackwell Heritage Collection)*.

### 3.7. Академия Google

Академия Google позволяет без труда выполнять обширный поиск научной литературы. Используя единую форму запроса, можно выполнять поиск в различных дисциплинах и по разным источникам, включая прошедшие рецензирование статьи, диссертации, книги, рефераты и отчеты, опубликованные издательствами научной литературы, профессиональными ассоциациями, высшими учебными заведениями и другими научными организациями. Академия Google позволяет найти исследование, наиболее точно соответствующее вашему запросу, среди огромного количества научных трудов.

Функции Академии Google:

- **Поиск** по различным источникам с одной удобной страницы.
- **Поиск** статей, рефератов и библиографических ссылок.
- **Поиск** полного текста документа в библиотеке или сети.
- **Получение информации** об основных работах в любой области исследований.

Определение рейтинга статей.

Академия Google классифицирует статьи так же, как и ученые, оценивая весь текст каждой статьи, ее автора, издание, в котором статья появилась, и частоту цитирования данной работы в научной литературе. Наиболее релевантные результаты всегда отображаются на первой странице.

### 3.8. Фонд содействия развитию малых форм предприятий в научно-технической сфере

Основными задачами Фонда являются:

- проведение государственной политики развития и поддержки малых предприятий в научно-технической сфере;
- оказание прямой финансовой, информационной и иной помощи малым инновационным предприятиям, реализующим проекты по разработке и освоению новых видов наукоемкой продукции и технологий на основе принадлежащей этим предприятиям интеллектуальной собственности;

- создание и развитие инфраструктуры поддержки малого инновационного предпринимательства;
- содействие созданию новых рабочих мест для эффективного использования, имеющегося в Российской Федерации научно-технического потенциала;
- привлечение внебюджетных инвестиций в сферу малого инновационного предпринимательства;
- подготовка кадров (в том числе вовлечение молодежи в инновационную деятельность).

В настоящее время Фонд реализует программы инновационного развития, которые направлены на создание новых и развитие действующих высокотехнологических компаний, коммерциализацию результатов научно-технической деятельности, привлечение инвестиций в сферу малого инновационного предпринимательства, создание новых рабочих мест.

Реализуя данные задачи, Фонд ежегодно оказывает финансовую поддержку более чем 1500 малым инновационным предприятиям, в более чем 150 городах Российской Федерации.

За время деятельности Фонда по всем программам было подано порядка 21 000 заявок на выполнение НИОКР и поддержано свыше 9 700 проектов из 75 субъектов Российской Федерации.

Участие малых инновационных предприятий в программах Фонда позволяет довести разработку от научной идеи до создания устойчивого бизнеса, привлекательного для отечественных и зарубежных инвесторов. Инновационные проекты предприятий проходят независимую экспертизу на научно-техническую новизну, финансово-экономическую обоснованность, перспективу рыночной реализации продукции.

### **3.9. Российский фонд фундаментальных исследований**

Основная задача Фонда, в соответствии с Уставом, — это проведение конкурсного отбора лучших научных проектов из числа тех, что представлены Фонду учеными в инициативном порядке, и последующее организационно-финансовое обеспечение поддержанных проектов.

Российский фонд фундаментальных исследований поддерживает фундаментальные исследования по следующим основным областям знаний:

- 1) математике, механике и информатике (01);
- 2) физике и астрономии (02);
- 3) химии и наукам о материалах (03);
- 4) биологии и медицинской науке (04);
- 5) наукам о Земле (05);
- 6) наукам о человеке и обществе (06)
- 7) информационным технологиям и вычислительным системам (07);
- 8) фундаментальным основам инженерных наук (08).

Для достижения основной цели — поддержки фундаментальных научных исследований — Фонд:

- а) проводит отбор на конкурсной основе проектов;
- б) разрабатывает и утверждает порядок рассмотрения представляемых на конкурс проектов, порядок проведения экспертизы проектов и предложений;
- в) осуществляет финансирование отобранных проектов и мероприятий, а также контролирует использование выделенных средств;
- г) поддерживает международное научное сотрудничество в области фундаментальных научных исследований, включая финансирование совместных научно-исследовательских проектов;
- д) осуществляет подготовку, выпуск и распространение информационных и других материалов о деятельности Фонда;
- е) участвует в выработке предложений по формированию государственной научно-технической политики в области фундаментальных научных исследований.

Объявления о проведении конкурсов, результаты конкурсного отбора, решения Фонда о финансировании проектов и мероприятий, а также другие материалы о деятельности Фонда публикуются в печати и распространяются через электронные средства информации.

### **3.10. Федеральное государственное бюджетное учреждение «Федеральный институт промышленной собственности» (ФИПС)**

<http://www.fips.ru>

Прием и экспертиза заявок на объекты патентного права, в т.ч. международных заявок на изобретения и полезные модели, рассмотрение предусмотренных законодательством Российской Федерации документов, представленных на государственную регистрацию договоров о распоряжении исключительным правом, и сделок, предусматривающих использование единой технологии за пределами Российской Федерации, выпуск официальных бюллетеней о зарегистрированных объектах патентных прав, поданных заявках и выданных по ним патентах, рассмотрение заявлений, ходатайств, касающихся продления срока действия исключительного права, рассмотрение возражений, касающихся решений по результатам экспертизы заявок на объекты патентного права, признания недействительным предоставления или досрочного прекращения действия правовой охраны, подготовка проектов решений.

Прием и экспертиза заявок на государственную регистрацию средств индивидуализации, выполнение работ по международным заявкам на товарный знак, рассмотрение заявлений о признании товарного знака или обозначения общеизвестным в Российской Федерации, рассмотрение предусмотренных законодательством Российской Федерации документов, представленных на государственную регистрацию договоров о распоряжении исключительным правом, средствах индивидуализации, выпуск официальных бюллетеней о зарегистрированных средствах индивидуализации; рассмотрение заявлений, ходатайств, касающихся продления срока действия исключительного права на средства индивидуализации, рассмотрение возражений и заявлений, касающихся решений по результатам экспертизы заявок на средства индивидуализации, признания недействительным предоставления или досрочного прекращения действия правовой охраны средств индивидуализации, подготовка проектов решений.

Прием и проверка заявок на государственную регистрацию программы для электронных вычислительных машин, базы данных и топологии интегральных микросхем, выпуск официальных бюллетеней о зарегистрированных свидетельствах, рассмотрение предусмотренных законодательством Российской Федерации документов, представленных на государственную регистрацию договоров о распоряжении исключительным правом, и сделок, предусматривающих использование единой технологии за пределами Российской Федерации, подготовка проектов решений.

Библиотечное, библиографическое, справочно-информационное, научно-методическое обслуживание на базе государственного патентного фонда и единой системы автоматизированных банков данных

### **3.11. Роспатент**

Федеральная служба по интеллектуальной собственности является правопреемником Федеральной службы по интеллектуальной собственности, патентам и товарным знакам, а также правопреемником Министерства юстиции Российской Федерации в части, касающейся правовой защиты интересов государства в процессе экономического и гражданско-правового оборота результатов научно-исследовательских, опытно-конструкторских и технологических работ военного, специального и двойного назначения, в том числе по обязательствам, возникающим в результате исполнения судебных решений.

Роспатент находится в ведении *Правительства Российской Федерации*.

Основными функциями Федеральной службы по интеллектуальной собственности являются:

а) контроль и надзор в сфере правовой охраны и использования результатов интеллектуальной деятельности гражданского, военного, специального и двойного назначения, созданных за счёт бюджетных ассигнований федерального бюджета, а также контроль и надзор в установленной сфере деятельности в отношении государственных заказчиков и организаций исполнителей

государственных контрактов, предусматривающих проведение научно-исследовательских, опытно-конструкторских и технологических работ;

б) оказание государственных услуг в установленной сфере деятельности;

в) нормативно-правовое регулирование вопросов, касающихся контроля, надзора и оказания государственных услуг в установленной сфере деятельности.

## 4. СИСТЕМА ПРЕДСТАВЛЕНИЯ УЧЕБНОЙ И НАУЧНОЙ ИНФОРМАЦИИ (LATEX)

### 4.1. Общие сведения. История. Текущее состояние

#### 4.1.1. Система компьютерной верстки TeX

Система компьютерной верстки TeX была создана выдающимся американским математиком и программистом Дональдом Кнудом в конце 70-х годов XX века. В 1979 году была выпущена первая версия системы.

Предпосылкой к созданию компьютерной типографии стало недовольство Кнута существующими на тот момент типографскими системами. Кнут выпустил первый том своей книги «Искусство программирования» в 1969 году. Издание было напечатано методом монотипии, технологии XIX века, которая давала на выходе издание в «хорошем классическом стиле». Второе издание публиковалось несколько позже, в 1976 году. Всю книгу пришлось перенабирать, поскольку монотипия повсеместно была замещена фотографической техникой. Когда 30 марта 1977 года Кнут получил новые оттиски, он был сильно удручен их качеством. Примерно в это же время Кнут впервые увидел результат работы высококачественной цифровой типографической системы и заинтересовался возможностями цифровой типографии. Не оправдавшие ожиданий оттиски дали ему дополнительный толчок к тому, чтобы разработав свою типографическую систему. 13 мая 1977 года он написал заметку самому себе, описывающую базовые возможности TeX. Примерно через 10 лет после начала работы над проектом а TeX был стабилизирован (версия 3.1). Кнут отошел от активной разработки. В дальнейшем система TeX модифицировалась только для целей исправления ошибок. На текущий момент рабочая версия TeX — 3.141592. Кнут завещал, что после его смерти версия будет заморожена и равна числу  $\pi$ , а все неисправленные ошибки будут считаться особенностями реализации.

Сегодня TeX — самый «безошибочный» программный пакет. За обнаружение ошибки в своей программе Кнут выплачивает вознаграждение, не большое, но невероятно ценное.

В отличие от обыкновенных текстовых процессоров и систем компьютерной верстки, построенных по принципу WYSIWYG («что вижу, то и получаю»), в TeX пользователь лишь задает текст и его структуру, а TeX самостоятельно на основе выбранного пользователем шаблона форматирует документ, заменяя при этом дизайнера и верстальщика. Документы набираются на собственном языке разметки в виде обычных текстовых (ASCII или Unicode) файлов, содержащих информацию о форматировании текста или выводе изображений. Эти файлы (обычно имеющие расширение «.tex») транслируются специальной программой в файлы «.dvi» (device independent — «независимые от устройства»), которые могут быть отображены на экране или напечатаны. DVI-файлы можно специальными программами преобразовать в PostScript, PDF или другой электронный формат.

Ядро TeX'a представляет собой язык низкоуровневой разметки, содержащий команды отступа и смены шрифта. Огромные возможности в TeX'e предоставляют готовые наборы макросов и расширений.

Можно выделить основные достоинства данной системы компьютерной верстки:

- Ни одна из существующих в настоящее время издательских систем не может сравниться с TeX'ом в полиграфическом качестве текстов с математическими формулами.
- Система TeX реализована на всех современных компьютерных платформах, и все эти реализации работают одинаково.
- Благодаря этому TeX стал международным языком для обмена математическими и физическими статьями: набрав свою статью в TeX'e, математик может послать ее по электронной почте своему коллеге, даже если отправитель работает под Windows, а получатель — с UNIX'ом или, допустим, на Макинтоше.
- Указав с помощью простых средств логическую структуру текста, автор может не вникать в детали оформления, причем эти детали при необходимости нетрудно изменить (для того, чтобы сменить шрифт, которым печатаются заголовки, не надо искать по всему тексту, а достаточно заменить одну строчку в так называемом стилевом файле). Такие вещи, как нумерация разделов, ссылки, оглавление и т.п. получаются почти что сами собой.

- Основные реализации TeX'a для всех платформ распространяются бесплатно.

Разумеется, у TeX'a есть и недостатки. Главный из них — в том, что с помощью TeX'a достаточно тяжело (хотя в принципе и возможно) готовить тексты со сложным расположением материала на странице (например, рекламных буклетов). Для таких приложений, практически не встречающихся в научно-технической литературе, TeX не предназначен.

### *LaTeX*

LaTeX (произносится по-русски — *латех*) — наиболее популярный набор макрорасширений (или макропакет) системы компьютерной *вёрстки* TeX, который облегчает набор сложных документов. Всё, что можно сделать в LaTeX'e, можно сделать и в обычном TeX'e. Однако благодаря различным упрощениям, использование макропакетов зачастую позволяет избежать весьма изощрённого программирования. Кроме базового набора существует множество пакетов расширения LaTeX.

Первая версия LaTeX была выпущена Лесли Лампортом в 1984 году, текущая версия — LaTeX2 $\epsilon$ , после создания в 1994 году испытывала некоторый период нестабильности, окончившийся к концу 90-х годов, а в настоящее время стабилизировалась (хотя раз в год выходит новая версия). Создатели LaTeX2 $\epsilon$  — Йоханнес Браамс (Johannes Braams), Михаэль Гуссенс (Michael Goossens), Алан Джефффри (Alan Jeffrey), Дэвид Карлайл (David Carlisle), Франк Миттельбах (Frank Mittelbach), Крис Роули (Chris Rowley) и Райнер Шёпф (Rainer Schopf).

LaTeX за годы своего существования завоевал серьёзное уважение издателей и стал мировым стандартом компьютерной вёрстки технических и научных текстов. С его помощью формируется техническая документация крупных проектов, пишутся статьи и книги по математике, физике, химии и пр. Во многих развитых компьютерных аналитических системах, например, Maple, Mathematica, Maxima возможен экспорт документов в формат \*.tex. И даже Википедия поддерживает TeX-нотацию для представления формул.

## *Лицензирование LaTeX*

LaTeX — свободное программное обеспечение, распространяется на условиях LaTeX Project Public License (LPPL). LPPL не совместима с GNU GPL, так как она требует, чтобы изменённые файлы были явно различимы с оригиналами (обычно, имели другие имена); это было сделано для того, чтобы быть уверенным, что зависимости между существующими файлами не будут нарушены, и для того, чтобы избежать проблем с совместимостью. Начиная с версии 1.3, LPPL совместима с критериями Debian по определению свободного программного обеспечения (DFSG). Так как LaTeX является свободным программным обеспечением, то он доступен для многих операционных систем включая Linux, Unix (включая ветку BSD), Windows, Mac OS X, RISC OS и AmigaOS.

Обычно LaTeX распространяется вместе с обычным TeX'ом. Будучи макропакетом, LaTeX предоставляет набор макросов TeX'а. Существуют и другие макропакеты, такие как Plain TeX, GNU Texinfo, AMSTeX и ConTeXt.

## **4.2. Быстрое введение. Документ и его структура**

### **4.2.1. Входной файл**

Исходный файл для системы LaTeX представляет собой собственно текст документа вместе со *специфики* и *командами*, с помощью которых системе передаются указания касательно размещения текста. Этот файл можно создать в любом текстовом редакторе, но при этом необходимо, чтобы в итоге получился так называемый чистый текстовый файл. Это означает, что текст не должен содержать шрифтовых выделений, разбивки на страницы и т.п.

Исходный текст документа не должен содержать переносов (TeX сделает их сам). Слова отделяются друг от друга пробелами, при этом TeX не различает, сколько именно пробелов вы оставили между словами (для управления пробелами вручную, есть специальные команды). Конец строки также воспринимается как пробел. Соседние абзацы должны быть отделены друг от друга пустыми строками (все равно, сколько именно пустых строк стоит между абзацами, важно, чтоб была хоть одна).

### 4.2.2. Преамбула

Входной файл должен начинаться с преамбулы. Преамбула начинается с декларации

```
\documentclass[options]{class}[release-date]
```

которая задаёт класс документа: LaTeX читает файл *class.cls*, содержащий определения команд, специфических для выбранного типа документа. Необязательный аргумент *options* позволяет изменить значения ряда параметров и некоторые правила форматирования, принятые по умолчанию для этого класса. Опции в списке *options* перечисляются через запятую. Необязательный аргумент *release-date* позволяет указать дату наиболее старой пригодной версии файла *class.cls*. Дата задаётся в формате «год/месяц/день». Пример:

```
\documentclass[a4paper,12pt]{article}[2000/05/19]
```

В LaTeX включены так называемые стандартные классы для поддержки наиболее популярных типов документа: *article* (статья), *proc* (доклад), *book* (книга), *report* (отчёт), *letter* (письмо) и *slides* (слайды).

После `\documentclass` для расширения базовой версии LaTeX используются декларации

```
\usepackage[options]{package}[release-date]
```

Декларация `\usepackage` стимулирует LaTeX читать файл *package.sty*, содержащий переопределения уже имеющихся команд и определения новых команд. Аргументы *options* и *release-date* имеют то же предназначение, что и у `\documentclass`. Количество деклараций `\usepackage` не ограничено. Одной декларацией можно загрузить сразу несколько пакетов, если, конечно, для каждого из них требуются одинаковые опции. Пример:

```
\usepackage[dvips]{graphicx,color}
```

Опции пакетов можно указывать также в аргументе *options* команды `\documentclass`.

Кроме описанных выше деклараций, в преамбуле обычно размещают всё то, что само ничего не печатает. Например, только в преамбуле документа может находиться декларация

$$\backslash\text{nofiles}$$

которая запрещает создавать любые служебные файлы.

### 4.2.3. Текст документа

Текст документа размещается за преамбулой в командных скобках

$$\backslash\text{begin}\{\text{document}\} \dots \backslash\text{end}\{\text{document}\}$$

Всё, что следует после  $\backslash\text{end}\{\text{document}\}$ , LaTeX игнорирует.

Существует возможность включения в документ текста из других файлов. Любая часть документа может храниться не только во входном файле. Команда

$$\backslash\text{input}\{file\}$$

позволяет включить в документ содержимое файла *file*. По умолчанию подразумевается расширение *tex*. LaTeX читает файл *file* от начала до конца или до команды

$$\backslash\text{endinput}$$

Команду  $\backslash\text{input}$  можно использовать и в преамбуле входного файла. В частности, сама преамбула может находиться в файле *file*.

Наряду с  $\backslash\text{input}$ , имеется команда

$$\backslash\text{include}\{file\}$$

которая также позволяет включить в документ содержимое файла *file.tex*. Команды  $\backslash\text{include}$  должны находиться внутри окружения *document*. Перед и после вставки в документ содержимого файла LaTeX начинает новую страницу (исполняется команда  $\backslash\text{clearpage}$ ). LaTeX читает файл только тогда, когда его имя *file* указано в аргументе декларации

$$\backslash\text{includeonly}\{files\}$$

которая должна находиться в преамбуле. Имена файлов в списке *files* перечисляются через запятую. Если для какой-нибудь ко-

манды `\include` имени файла нет в этом списке, то LaTeX просто переходит на новую страницу.

Список файлов, которые читает LaTeX при обработке входного файла можно вывести в файл протокола (имеет расширение `log`), поместив в преамбуле команду

```
\listfiles
```

#### 4.2.4. Поддержка русского языка

В декабре 1998 года в рамках проекта LaTeX3 реализована поддержка русского языка в соответствии со стандартом. Все необходимые средства распространяются сейчас вместе с LaTeX'ом.

Пакет **inputenc** надо подключать с опцией, соответствующей кодировке символов во входном файле. Так, в среде MS Windows, в которой используется кодовая страница 1251, в преамбулу входного файла надо включить декларацию

```
\usepackage[cp1251]{inputenc}
```

Кроме пакета **inputenc**, надо подключить пакет **babel** с опцией `russian`:

```
\usepackage[russian]{babel}
```

Будет установлена кодировка текстовых шрифтов T2A с русскими буквами, включены правила переноса русских слов, переопределены стандартные заголовки и введены новые команды для набора символов, специфических для русского языка.

Если документ на русском языке содержит целые абзацы английского текста, то перед ними надо ставить любую из двух эквивалентных команд

```
\English \Eng
```

Они включают правила переноса слов английского языка, что даёт более аккуратную вёрстку абзацев. После окончания английского текста надо поставить любую из двух эквивалентных команд

```
\Russian \Rus
```

Они восстановят правила переноса для русского языка. Приведённые выше команды переключения языка становятся доступными после подключения пакета **babel** с опцией `russian`.

Если документ на английском языке содержит русский текст, то вместо пакета **babel** можно ограничиться подключением пакета `fontenc` с опцией `T2A`:

```
\usepackage[T2A]{fontenc}
```

Будет установлена кодировка текстовых шрифтов `T2A` и `LaTeX` сверстает русский текст, но без переносов в словах, что может ухудшить качество вёрстки. Для небольших фрагментов русского текста качество вёрстки можно улучшить, задавая места переноса в русских словах командой `\-` или в декларации `\hyphenation`.

Если же документ на английском языке содержит большие фрагменты русского текста, то лучше воспользоваться пакетом **babel** с двумя опциями:

```
\usepackage[russian,english]{babel}
```

Поскольку опция `english` стоит последней, основным языком документа будет английский и все стандартные заголовки будут печататься по-английски. Перед абзацами на русском языке надо ставить команду `\Russian` или `\Rus`, а после них — `\English` или `\Eng`. Для вставки коротких фраз на русском языке можно воспользоваться командой

```
\textcyrillic{text}
```

#### 4.2.5. Комментарии

Всё, что следует в строке за символом `%`, `LaTeX` игнорирует. Пакет **verbatim** из коллекции `tools` вводит командные скобки

```
\begin{comment} ... \end{comment}
```

Всё, что находится в них, `LaTeX` игнорирует.

## 4.3. Печатный документ

### 4.3.1. Параметры страницы

Страница печатного документа состоит из верхнего и нижнего колонтитулов и области, в которой размещается содержание документа: текст и подстрочные примечания. Кроме того, на боковых полях страницы могут размещаться заметки на полях, которые печатает команда `\marginpar`. Размер и расположение колонтитулов, области с содержанием документа и заметок на полях задаются нерастяжимыми командными длинами, приведёнными на рис. 4.1. Их значения, установленные по умолчанию, можно изменить в преамбуле документа декларациями `\setlength` и `\addtolength`.

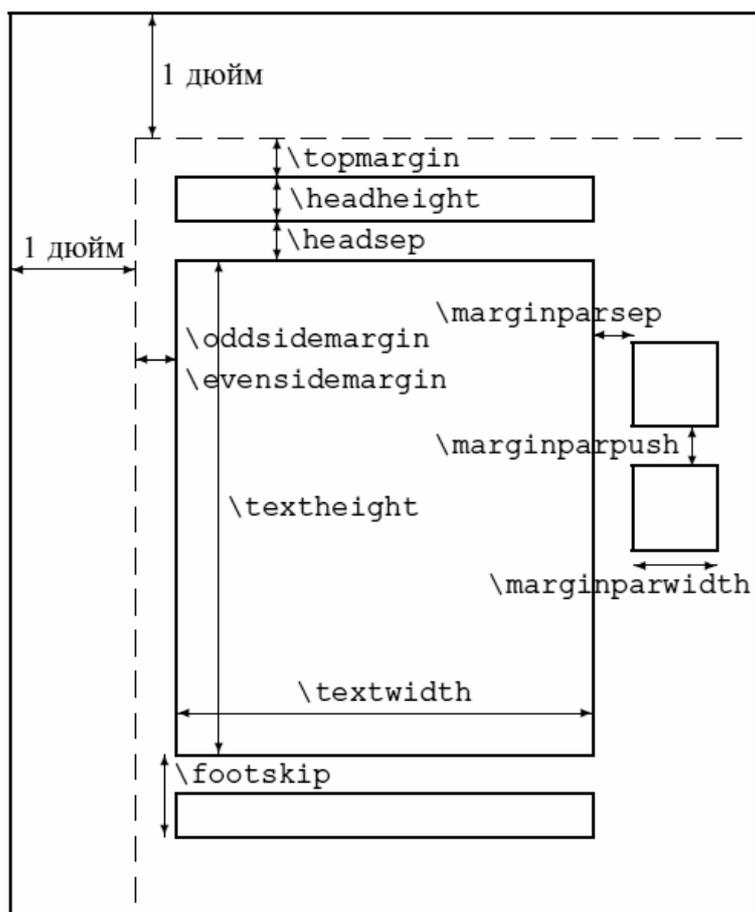


Рис. 4.1 — Макет страницы с командами, задающими размер и расположение верхнего и нижнего колонтитулов, области с содержанием документа и заметок на полях

Команды `\oddsidemargin` и `\evensidemargin` задают левое поле для нечётных (odd) и чётных (even) страниц, соответственно.

Текущие значения параметров компоновки страницы можно узнать с помощью пакета **layout** из коллекции `tools`. Команда `\layout` из этого пакета печатает макет страницы, на которой она находится, с указанием значений всех параметров. Команда различает правые и левые страницы, одно- и двухколоночный режимы печати.

### 4.3.2. Титульная страница и аннотация

Стандартный заголовок, состоящий из названия, имён авторов и даты создания документа, печатает команда

```
\maketitle
```

Ей должны предшествовать две команды

```
\title{title}
```

```
\author{author(s)}
```

содержащие название документа *title* и имена авторов *author(s)*. Для разбиения длинного названия или списка авторов на строки используется команда `\\`. Аргументы обеих команд могут быть пустыми.

Перед `\maketitle` можно с помощью команды

```
\date{date}
```

указать дату создания документа *date*. Если команда `\date` отсутствует, то печатается текущая дата. Если дата не нужна, то надо использовать команду `\date` с пустым аргументом `{}`. Аргументы команд `\title`, `\author` и `\date` могут содержать команду

```
\thanks{text}
```

которая печатает *text* как подстрочное примечание.

В стандартных классах команда `\maketitle` печатает заголовок на отдельной странице, если действует опция `titlepage`. Страница, следующая за титульной, нумеруется как 1-ая. Если действует опция `notitlepage`, то заголовок печатается с новой страницы прямо перед содержанием документа. В классе `article` по умолча-

нию используется `notitlepage`, а в классах `book`, `report` и `slides` — `titlepage`.

В аргументе команды `\author` можно использовать команду

$$\text{\and}$$

для разделения *author(s)* на боксы. LaTeX, формируя из этих боксов строку, отделяет их друг от друга большими пробелами. Каждый бокс может сам состоять из нескольких строк.

Если формат стандартного заголовка не соответствует требуемому, то надо использовать командные скобки

$$\text{\begin\{titlepage\} \dots \end\{titlepage\}}$$

для создания титульной страницы. На этой странице печатается содержание окружения `titlepage`. Страница, следующая за титульной, нумеруется как 1-ая.

В классах `article` и `report` определены командные скобки

$$\text{\begin\{abstract\} \dots \end\{abstract\}}$$

для печати аннотации к статье. Она печатается на отдельной странице, если действует опция `titlepage`. Перед аннотацией LaTeX печатает заголовок `Abstract`. Он хранится в команде

$$\text{\abstractname}$$

которую можно переопределить с помощью `\renewcommand`. Пакет **babel** с опцией `russian` переопределяет её как Аннотация.

При наличии опции `twocolumn` аннотация, как и сам документ, печатается в двухколоночном режиме. Пакет **abstract** позволяет напечатать её в одноколоночном.

### 4.3.3. Секционирование документа

В стандартных классах, за исключением `letter`, определены команды для разделения документа на секции. Все команды имеют по два аргумента, один из которых является обязательным. Обязательный аргумент печатается как название секции и, если отсутствует необязательный аргумент, включается в оглавление и используется при оформлении колонтитулов. Для оглавления и колонтитулов можно задать более компактное название секции в виде необязательного аргумента. Неустойчивые коман-

ды в подвижных аргументах должны быть защищены командой `\protect`. Для разбиения длинного названия на строки внутри обязательного аргумента можно использовать команду `\`.

LaTeX автоматически нумерует секции, уровень которых не превышает число, которое хранится в счётчике

`secnumdepth`

Его значение можно изменить декларацией `\setcounter`. Секциям, созданным командой `\section`, присвоен уровень 1. Счётчики секций более младшего уровня определены как внутренние по отношению к счётчику секций старшего уровня, поэтому в каждой секции подсекции нумеруются независимо друг от друга.

Все команды секционирования имеют \*-форму. Она имеет только обязательный аргумент, который печатается как название секции. Такие секции не нумеруются и не заносятся в оглавление и колонтитулы.

#### 4.3.4. Части

Команды

`\part[toc]{head}`    `\part*{head}`

позволяют разделить документ на части. В классах `book` и `report` команда `\part` печатает заголовок в центре отдельной страницы: сначала слово `Part` и порядковый номер части прописными римскими цифрами, а затем с новой строки название части `head`. В классе `article` заголовок печатается прямо перед содержанием части. При наличии опции, для оглавления и колонтитулов используется не `head`, а `toc`. Команда `\part*` печатает только название части.

Команды `\part` и `\part*` являются необязательными, поэтому они не влияют на порядок нумерации более младших секций. Слово `Part` хранится в команде

`\partname`

Её можно переопределить с помощью `\renewcommand`. Пакет **babel** с опцией `russian` переопределяет её как `Часть`.

### 4.3.5. Главы

В классах `book` и `report` определены команды для разделения документа на главы:

```
\chapter[toc]{head}   \chapter*{head}
```

Каждая глава печатается с новой страницы: правой, если действует опция `openright`, или любой, если действует опция `openany`. В классе `book` по умолчанию используется `openright`, а в классе `report` — `openany`.

Команда `\chapter` печатает сначала слово `Chapter` и порядковый номер главы арабскими цифрами. Название главы *head* печатается с новой строки. При наличии опции, для оглавления и колонтитулов используется не *head*, а *toc*. Команда `\chapter*` печатает только название главы. Слово `Chapter` хранится в команде

```
\chaptername
```

Её можно переопределить с помощью `\renewcommand`. Пакет **babel** с опцией `russian` переопределяет её как Глава.

### 4.3.6. Разделы

Для деления документа на разделы, подразделы и подподразделы определены команды:

```
\section[toc]{head}
```

```
\subsection[toc]{head}
```

```
\subsubsection[ toc ]{head}
```

```
\section*{head}
```

```
\subsection*{head}
```

```
\subsubsection*{head}
```

Они печатают перед названием секции *head* её порядковый номер. При наличии опции, для оглавления и колонтитулов используется не *head*, а *toc*. \*-форма команд печатает только название секции.

### 4.3.7. Параграфы

Команды

```
\paragraph[toc]{head}
```

```
\subparagraph[toc]{head}
```

```
\paragraph*{head}
```

```
\subparagraph*{head}
```

печатают в начале первого абзаца секции шрифтом полужирной насыщенности её название *head*. При этом, команда `\paragraph` подавляет абзацный отступ, а `\subparagraph` — нет. Между секциями вставляется дополнительный вертикальный промежуток. При наличии опции, для оглавления и колонтитулов используется не *head*, а *toc*. \*-форма команд печатает только название секции.

### 4.3.8. Приложения

Если в документе имеются приложения, то перед ними надо поставить декларацию

```
\appendix
```

После неё новая глава начинается не со слова *Chapter*, а со слова *Appendix*. Кроме того, меняется формат нумерации глав: вместо арабских цифр используются заглавные латинские буквы *A*, *B*, *C* и т. д. В классе *article*, где нет глав, буквами нумеруются разделы, созданные командой `\section`. Слово *Appendix* хранится в команде

```
\appendixname
```

Её можно переопределить с помощью `\renewcommand`. Пакет **babel** с опцией *russian* переопределяет её как Приложение.

## 4.4. Основные команды

### 4.4.1. Переключение в математическую моду

В исходном тексте математические выражения, а также верхние и нижние индексы, выделяются специальными командными скобками для переключения TeX'a в математическую моду

вёрстки. В математической моде TeX игнорирует *все* пробелы между символами в исходном тексте и расставляет промежутки сам. Формулы *не* должны содержать пустых строк. Кроме латинских букв a..z, A..Z и цифр 0..9, распознаются и печатаются следующие символы:

$$+ - = * / < > ( ) [ ] | . , ; ? ! : ` ' " @$$

Любая буква считается именем переменной и печатается шрифтом «математический курсив». Символы  $\wedge$  и  $\_$  являются служебными: они используются для набора верхних и нижних индексов, соответственно. Математические символы, а также греческие буквы, которых нет на клавиатуре, печатаются специальными командами. Имена команд обычно совпадают с названиями символов.

После подключения пакета **mathtext** согласно общим правилам

$$\backslash usepackage\{mathtext\}$$

в математической моде можно использовать русские буквы, набирая их прямо на клавиатуре.

#### 4.4.2. Формулы внутри текста

Формулы внутри текста надо размещать между командами  $\($  и  $\)$  или в окружении `math` (между  $\begin\{math\}$  и  $\end\{math\}$ ), или выделять с обеих сторон знаками доллара  $\$$ :

Катеты  $\$a\$$  и  $\$b\$$  треугольника связаны с гипотенузой  $\$c\$$  формулой

$$\backslash(c^2=a^2+ b^2\backslash).$$

При вёрстке абзаца TeX может разорвать формулу для переноса её части на другую строку. Переносы возможны после символов бинарных отношений типа знака равенства и символов бинарных операций типа знака сложения, причём последний знак в строке, вопреки российской традиции, не дублируется в начале следующей. В остальных местах формулы не разрываются. Поэтому надо избегать длинных неразрываемых фрагментов в формулах, иначе могут возникнуть проблемы с вёрсткой абзаца.

Имеется специальная команда `\*`, указывающая возможное место для переноса формулы с одной строки на другую по знаку умножения:

$$$(x+y)\*(z+t)$.$$

## 4.5. Выделенные формулы

Выделенные в отдельную строку формулы TeX печатает с временным прерыванием текущего абзаца. Это означает, что если после формулы нет пустой строки или команды `\par`, то следующая за формулой строка печатается без абзацного отступа.

### 4.5.1. Однострочные уравнения

Формулы, которые должны печататься без порядкового номера на отдельной строке, располагают между командами `\[` и `\]` или в окружении `displaymath`:

Катеты  $a$  и  $b$  треугольника связаны с гипотенузой  $c$  формулой

$$\[ c^2=a^2+b^2.\]$$

Знаки пунктуации после выделенной формулы ставятся в конце самой формулы! Иначе этот знак появится в начале новой строки.

Если математическое выражение в строке состоит не из одной формулы, а из нескольких, то пробелы между ними надо составлять вручную. Обычно используется команда `\qquad`:

$$\[F_n=F_{n-1}+F_{n-2},\qquad n\geq 2.\]$$

В математических текстах формулы обычно нумеруются для того, чтобы на них можно было ссылаться по ходу документа. TeX позволяет организовать нумерацию таким образом, чтобы номера формул и ссылки на них создавались автоматически. Чтобы TeX пронумеровал формулу, её надо поместить в окружение `equation`. Каждая такая формула на печати автоматически получает свой порядковый номер:

Катеты  $a$  и  $b$  треугольника связаны с гипотенузой  $c$  формулой

```
\begin{equation}
```

$$c^2 = a^2 + b^2.$$

```
\end{equation}
```

Чтобы на присвоенный формуле номер можно было ссылаться в тексте, надо формулу пометить: в любом месте между `\begin{equation}` и `\end{equation}` поставить команду `\label`, аргументом которой будет «имя» формулы, и после этого команда `\ref` будет печатать номер этой формулы.

Пример:

```
\begin{equation}\label{eq:exp}
```

$$e^x = 1 + x$$

```
\end{equation}
```

Уравнение~(`\ref{eq:exp}`) справедливо очень малых  $x$ , только при очень малых  $x$ .

Отметим, что скобки вокруг номера формулы, созданного командой `\ref`, автоматически *не* ставятся.

То, как именно выглядит на печати номер формулы, зависит от класса документа: например, в классе `article` формулы имеют сплошную нумерацию, а в классе `book` нумерация формул начинается заново в каждой главе и номер, скажем, 5-ой по счёту пронумерованной формулы из главы 3, имеет вид (3.5).

#### 4.5.2. Системы уравнений

Для набора систем уравнений в TeX'e предусмотрены окружения `eqnarray` для нумерованных уравнений и `eqnarray*` для нумерованных формул. Внутри окружения уравнения, которые должны размещаться на отдельных строках, отделяются друг от друга командами `\\`. Выражение в пределах одной строки должно состоять из *трёх* частей (возможно пустых), разделённых амперсантами `&`. Каждая часть помещается в свой столбец. В левом столбце формулы прижимаются к правому краю, в среднем — центрируются, а в правом столбце — прижимаются к левому

краю. Другими словами, символы `&` задают точки выравнивания уравнений из разных строк по вертикали. Каждая строка в окружении `eqnarray` получает при печати свой номер. Подавить нумерацию любой строки можно командой `\nonumber`. Пример:

```
\begin{eqnarray}
      I & = & U + pV \\
\Psi & = & U - TS \\
\Psi + PV & = & \Phi \nonumber
\end{eqnarray}
```

Окружения `equation` и `eqnarray` используют один и тот же счётчик `equation`, значение которого и печатается как номер формулы. Следовательно, все нумерованные уравнения в документе будут иметь единую нумерацию.

Для печати системы уравнений без выравнивания по вертикали пакет **amsmath** вводит окружение `gather`. Оно даёт нумерованные уравнения. Для нумерованных формул надо использовать окружение `gather*`. Точки переноса строк задаются командой `\`. Все строки центрируются.

Пример:

```
\begin{gather}
a_x = -Hy \\
A_y = 0
\end{gather}
```

Для набора систем уравнений с выравниванием по вертикали, пакет **amsmath** предлагает окружения `align`, `alignat` и `flalign` для нумерованных уравнений и `align*`, `alignat*` и `flalign*` для нумерованных уравнений. Точки переноса строк задаются командой `\`. Точки выравнивания уравнений из разных строк по вертикали задаются амперсантом `&`. В отличие от окружения `eqnarray`, уравнения не надо разбивать на три части.

Пример:

```
\begin{align}
\Psi &= U - TS \\
\Psi + PV &= \Phi
\end{align}
```

В окружениях `align` и `align*` между столбцами уравнений, а также перед первым столбцом и после последнего столбца, автоматически вставляются равные пробелы. При расчёте величины промежутка наличие номера у строки не учитывается.

В окружениях `alignat` и `alignat*` автоматически вставляются равные пробелы только перед первым столбцом и после последнего столбца. Промежутки между столбцами уравнений автоматически не вставляются. Их надо задавать самому командами, которые вставляют горизонтальные промежутки в математической моде. Окружения `alignat` и `alignat*` имеют обязательный аргумент, значение которого указывает количество столбцов. Число символов `&` в каждой строке не должно превышать необходимого для создания этого количества столбцов.

## 4.6. Списки

LaTeX поддерживает списки разного типа. Во всех случаях каждый элемент списка должен начинаться с команды

```
\item[label]
```

Необязательный аргумент *label* печатается перед содержанием элемента, заменяя собой маркёр или номер, принятые для этого элемента по умолчанию. Каждый элемент одного списка может включать в себя другие списки.

### 4.6.1. Маркированные списки

Командные скобки

```
\begin{itemize} items \end{itemize}
```



Элементы списка первого уровня нумеруются арабскими цифрами с точкой: 1., 2., 3. и т. д., второго уровня — строчными латинскими буквами в круглых скобках: (a), (b), (c) и т.д., третьего уровня — строчными римскими цифрами с точкой: i., ii., iii. и т.д. и четвёртого уровня — прописными латинскими буквами с точкой: A., B., C. и т.д.

Чтобы изменить формат, в котором будет напечатано значение счётчика, надо переопределить сам счётчик. Например, в документах на русском языке после переопределения

```
\renewcommand{\theenumii}{\asbuk{enumii}}
```

значение счётчика элементов списка второго уровня будет печататься строчными русскими буквами.

#### 4.7. Позиционирование текста в строке

Все строки в командных скобках

```
\begin{center} ... \end{center}
```

или в области действия декларации `\centering` центрируются.

Строки в командных скобках

```
\begin{flushleft} ... \end{flushleft}
```

или в области действия декларации `\raggedright` прижимаются к левому краю внешнего текста.

Строки в командных скобках

```
\begin{flushright} ... \end{flushright}
```

или в области действия декларации `\raggedleft` прижимаются к правому краю внешнего текста.

Во всех случаях: если текст не помещается в одной строке, то он автоматически переносится на следующую строку без переносов в словах. Промежутки между словами фиксированы. Для разбиения текста на строки надо использовать команду `\\`.

Первая строка абзаца, следующего за окружениями `center`, `flushleft` и `flushright`, начинается со стандартного отступа, если между ними вставлена пустая строка или команда `\par`.

## 4.8. Таблицы

Командные скобки

```
\begin{tabular}[align]{keys}
strings
\end{tabular}
```

создают таблицу, которая является боксом и, следовательно, должна уместиться на одной странице (её обычно помещают в окружение `table`, которое создаёт нумерованные плавающие объекты).

Аргумент *align* задаёт расположение таблицы по вертикали в текущей строке. Его допустимые значения: *t*, *c* (по умолчанию) и *b*. По умолчанию таблица позиционируется по центру текущей строки. В случае *t* и *b* — таблица позиционируется так, чтобы базовая линия её соответственно первой и последней строки совпала с базовой линией текущей строки.

Количество колонок в таблице задаётся в обязательном аргументе *keys*. Колонки создаются ключами *l*, *c*, *r* и *p{width}*: по одной колонки на каждый ключ. В случае *l*, *c* и *r* — текст в ячейке печатается в строковой моде, причём ширина колонки устанавливается равной ширине самой широкой ячейки. В остальных ячейках этой колонки текст прижимается к левому (*l*) или к правому (*r*) краю или центрируется (*c*). Ключ *p{width}* создаёт колонку, в ячейках которой текст печатается в виде текстового бокса шириной *width*.

Кроме описанных выше ключей *l*, *c*, *r* и *p{width}*, в аргументе *keys* используются ключи `|` и `@{text}`. Ключ `|` задаёт вертикальную разделительную линию между колонками на всю высоту таблицы. Ключ `@{text}` отменяет вставку пробела между колонками: вместо него во всех строках таблицы печатается *text* в качестве разделителя колонок. Этот ключ с пустым аргументом `@{ }` можно использовать для подавления пробела перед первой и после последней колонки.

Несколько ячеек можно слить в одну с помощью команды

```
\multicolumn{n}{keys}{text}
```

Здесь  $n$  — количество ячеек ( $n = 1, 2, \dots$ ).

Расстояние между строками таблицы можно изменить переопределением с помощью декларации `\renewcommand` междустрочного интервала `\arraystretch`.

#### 4.8.1. Таблицы заданной ширины

Командные скобки

```
\begin{tabular*}{width}[align]{keys}
strings
\end{tabular*}
```

создают таблицу шириной *width*: между колонками должен быть вставлен эластичный пробел, способный растянуть таблицу до заданной ширины. В строке под такие таблицы отводится место согласно заказанной ширине. Если истинная ширина таблицы (определяется содержанием ячеек) больше, чем заказанная, то таблица наедет на соседним с ним текст. Эластичный пробел обычно вставляется с помощью `@{\extracolsep{\fill}}`. В остальном окружение `tabular*` идентично стандартной версии `tabular`.

В пакете **tabularx** из коллекции `tools` определены командные скобки

```
\begin{tabularx}{width}[align]{keys}
strings
\end{tabularx}
```

которые создают таблицу шириной *width*, но не за счёт увеличения расстояния между колонками, как в случае `tabular*`, а за счёт подбора ширины колонок.

#### 4.8.2. Размещение таблицы на нескольких страницах

В пакете **longtable** определены командные скобки

```
\begin{longtable}[position]{keys}
strings
\end{longtable}
```

для печати таблиц на нескольких страницах. Печать начинается с новой строки на текущей странице. Положение таблицы по горизонтали задаётся аргументом *position* с допустимыми значениями l, c (по умолчанию) и r. По умолчанию таблица центрируется, а в случае l и r — прижимается соответственно к левому и правому полю страницы. Колонки и разделители между ними задаются в аргументе *keys*, причём используются те же ключи, что и в стандартной версии *tabular*.

Обычно первые строки таблицы содержат заголовки колонок. Если таблица расположена на нескольких страницах, то такие строки, как правило, должны дублироваться в начале каждой страницы. Кроме того, в конце страницы, где происходит разрыв таблицы, целесообразно указать, что она продолжается на следующей странице. Поддержка описанных выше требований к длинным таблицам реализована в **longtable** следующим образом. В начале окружения надо задать четыре группы строк, причём любая группа может быть опущена. Группа строк, после которой стоит команда

`\endfirsthead`

будет напечатана только в самом начале таблицы. Строки этой группы содержат обычно как заголовок самой таблицы, так и заголовки колонок. Группа строк, после которой стоит команда

`\endhead`

будет напечатана в верхней части таблицы на всех страницах, кроме первой. Группа строк, после которой стоит команда

`\endfoot`

будет напечатана в нижней части таблицы на всех страницах, кроме последней. И, наконец, группа строк, после которой стоит команда

`\endlastfoot`

будет напечатана только в самом конце таблицы. Подпись к таблице можно напечатать командой

`\caption[entry]{head}`

Она, как и подпись, созданная командой `\caption` в окружении `table`, начинается с ключевого слова `Table`, за которым следует порядковый номер таблицы, двоеточие и текст *head*. При наличии опции, текст *entry*, а не *head* включается в список таблиц. Можно также воспользоваться командой

$$\backslash\text{caption}*\{head\}$$

Она не нумерует таблицу, а её аргумент не заносится в список таблиц. По умолчанию таблица центрируется по горизонтали.

## 4.9. Вставка графики

Ответственным за создание «бокса» для размещения картинки является пакет **graphicx**, а точнее команда `\includegraphics`:

$$\backslash\text{includegraphics}[\text{width}=\backslash\text{textwidth}]\{title.eps\}$$

`\textwidth` — ширина тела текста. В команде `\includegraphics` есть один обязательный параметр — вставляемая картинка. Необязательные параметры передаются с помощью пар «ключ» = «значение», разделяемых запятой.

## 4.10. Плавающие объекты

### 4.10.1. Рисунки и таблицы

Чтобы избежать полупустых страниц, любой материал (рисунки, таблицы и т.п.), который не подлежит переносу на следующую страницу по частям, надо оформлять как плавающий объект: `LaTeX` сам подыщет для него подходящее место, передвигая на следующие страницы. Плавающие объекты создают командные скобки

$$\backslash\text{begin}\{figure\}[\textit{placement}] \dots \backslash\text{end}\{figure\}$$

$$\backslash\text{begin}\{table\}[\textit{placement}] \dots \backslash\text{end}\{table\}$$

При печати в две колонки каждый объект размещается в одной из колонок. Плавающие объекты, созданные командными скобками

$$\backslash\text{begin}\{figure^*\}[\textit{placement}] \dots \backslash\text{end}\{figure^*\}$$

$$\backslash\text{begin}\{table^*\}[\textit{placement}] \dots \backslash\text{end}\{table^*\}$$

занимают обе колонки при двухколоночном режиме печати. Опция *placement* уточняет правила размещения объекта. Она может состоять из любой последовательности следующих ключей:

- *h* — разрешает размещение объекта сразу после заполнения текущей строки текста. Не используется в \*-формах при двухколоночной печати.
- *t* — разрешает размещение объекта в верхней части страницы над текстом.
- *b* — разрешает размещение объекта в нижней части страницы под текстом.
- *p* — разрешает размещение объекта на специальной странице, содержащей только плавающие объекты.
- *!* — снимает ряд описанных ниже ограничений, которые могут препятствовать размещению объекта. Используется только в комбинации с другими ключами.

По умолчанию в стандартных классах используется последовательность *tbr*.

Подпись к рисунку или таблице печатает команда

$$\backslash\text{caption}[entry]\{head\}$$

Подпись начинается с ключевого слова: *Figure* в окружениях *figure* и *Table* в окружениях *table*, за которым следует порядковый номер рисунка или таблицы, и затем после двоеточия печатается текст *head*. При наличии опции, *entry*, а не *head* включается в список рисунков и таблиц.

Слова *Figure* и *Table* хранятся соответственно в командах

$$\backslash\text{figurename} \quad \backslash\text{tablename}$$

Их можно переопределить с помощью *\renewcommand*. Пакет **babel** с опцией *russian* переопределяет их как *Рис.* и *Таблица*. В России после номера рисунка или таблицы принято ставить точку, а не двоеточие, как в стандартном TeX'e. Необходимую замену позволяет сделать пакет **caption2**. Для этого достаточно переопределить введённую в этом пакете команду

$$\backslash\text{captionlabeldelim}$$

следующим образом

$$\backslash\text{renewcommand}\{\backslash\text{captionlabeldelim}\}\{.\}$$

#### 4.10.2. Рисунки и таблицы, обтекаемые текстом

Пакет **floatflt** вводит окружения `floatingfigure` и `floatingtable` для размещения небольших рисунков и таблиц так, чтобы текст мог обтекать их. Рисунок или таблица располагаются у левого или правого поля страницы, причём их верхний край совпадает с первой строкой того абзаца, в начале которого они размещены в исходном тексте. Пакет имеет три опции: `rflt`, `lflt` и `vflt`. По умолчанию действует `vflt` и обтекаемый объект располагается у левого поля на чётных страницах и у правого поля на нечётных страницах. Опции `rflt` и `lflt` задают размещение всех объектов соответственно только у правого и у левого поля страницы.

Окружение `floatingfigure` имеет следующий синтаксис:

```
\begin{floatingfigure}[position]{width}
figure
\end{floatingfigure}
```

Аргумент *width* задаёт ширину бокса, который отводится под рисунок *figure*. Опция *position* позволяет изменить заданное опцией пакета расположение рисунка на странице. Значения `r`, `l` и `v` задают те же правила, что и опции пакета `rflt`, `lflt` и `vflt`. Значение `v` указывает, что для текущего объекта надо использовать опцию пакета.

Подпись к рисунку, как и в окружении `figure`, можно напечатать командой `\caption`.

Окружение `floatingtable` имеет следующий синтаксис:

```
\begin{floatingtable}[position]{tabular environment}
[\caption{text}]
\end{floatingtable}
```

Здесь обязательным аргументом является окружение `tabular` (или `\parbox`) и ширина обтекаемого объекта определяется естественной шириной таблицы. Опция *position* имеет то же предназначение, что и у `floatingfigure`.

Подпись к таблице, как и в окружении `table`, можно напечатать командой `\caption`.

### 4.10.3. Управление плавающими объектами

Если плавающих объектов в документе немного, то всё будет хорошо без какого-либо вмешательства человека. Но если их много, то, так или иначе, надо будет управлять их размещением.

Если LaTeX не справляется с размещением картинок, то он переносит их на следующую страницу. В какой-то момент может накопиться целая «толпа» таких перенесённых картинок и возникнет необходимость в их «насильственном» выводе в каком-то определённом месте. Для этого существует команда

```
\clearpage
```

При вызове команды `\clearpage` завершается текущая страница, выводятся все отложенные плавающие объекты, и только потом продолжается обычный вывод текста. Единственная проблема этой команды в том, что текущая страница по ней обрывается. Чтобы избежать обрыва, можно воспользоваться пакетом **afterpage**, точнее одноимённой командой из него:

```
\afterpage {\clearpage}
```

Команда `\afterpage` откладывает выполнение указанных в ней инструкций до конца текущей страницы.

Команда

```
\suppressfloats
```

полностью подавляет размещение плавающих объектов на текущей странице. В качестве необязательного параметра ей можно передать `t` или `b` — в этом случае запрет распространяется только на размещение плавающих объектов вверху или внизу страницы соответственно.

Пакет **placeins** не даёт «утекать» плавающим объектам за установленные пределы. Барьер устанавливается с помощью команды

```
\FloatBarrier
```

Это бывает полезно, когда хочется, чтобы все картинки не выходили за пределы своего раздела. В этом случае следует переопределить нужную команду секционирования для установки пе-

ред ней барьеров. В случае команды секционирования раздела (section) достаточно передать опцию [section] пакету при загрузке:

```
\usepackage [ section ]{placeins}
```

Часто при подготовке статей требуют их размещения после текста на отдельных страницах, предваряя эту галерею списком иллюстраций. Пакет **endfloat** именно это и делает. Достаточно его загрузить.

## 4.11. Обзор пакетов

### 4.11.1. Интернационализация и локализация

Пакеты **fontenc**, **inputenc**, **babel**, **indentfirst** — это обязательная часть программы, если вы собираетесь создавать документ на русском языке:

**fontenc** — стандартный пакет для выбора внутренней кодировки LaTeX;

**inputenc** — стандартный пакет для указания, в какой кодировке набран текст;

**babel** — стандартный пакет локализации или выбора языка документа. Пакет поддерживает сорок три языка без учёта диалектов.

Чтобы использовать **babel** с русскими текстами, необходимо включить в преамбулу документа строки:

```
\usepackage[...]{inputenc}
```

```
\usepackage[russian]{babel}
```

В квадратных скобках вместо ... нужно указать кодировку, в которой будет набираться русский текст, например: cp866, cp1251, koï8-r или utf8 (рекомендуется). Эти строки должны находиться в преамбуле до вызова каких-либо пакетов, использующие русские буквы. После этого в тексте можно набирать русские буквы в указанной кодировке;

**braille** — поддержка набора с помощью шрифта Брайля (для слепых).

### 4.11.2. Рубрикация

**titlesec** — данный пакет предоставляет альтернативный способ формирования заголовков разделов, сильно отличающийся от стандартной схемы. Для работы с оглавлением в том же стиле следует использовать пакет **titletoc**;

**appendix** — дополнительные возможности по работе с приложением.

### 4.11.3. Пакеты для работы с рисунками

**figsize** — специализируется на автоматическом вычислении размеров картинок для размещения их в указанных пределах;

**graphicx** — добавление иллюстраций в документ;

**epstopdf** из пакета **oberdiek** — позволяет подключать eps-файлы при компиляции с помощью `pdflatex`, вызывая внешнюю программу `epstopdf`;

**ncopic** из пакета **ncctools** — расширение возможности пакета **graphicx** при работе с растровыми изображениями;

**hilowres** — позволяет определить при вставке два файла для одного и того же рисунка, например, с низким и высоким разрешением. В зависимости от выбора опций при компиляции подключается либо один, либо другой файл;

**rotating** — позволяет поворачивать любые боксы;

**watermark** из пакета **ncctools** — создание «водяных знаков», иначе говоря, вывод графических объектов или текста на подложке страницы. Схожую функциональность предоставляют пакеты **bophook**, **eso-pic** и **wallpaper**;

**XY-pic** — это пакет для создания графов и диаграмм. Графы строятся в виде матрицы, где каждый элемент матрицы соответствует вершине графа. Рёбра графа строятся с помощью специальных команд.

Для подключения пакета в преамбуле документа пишем

```
\usepackage[all]{xy}
```

#### 4.11.4. Пакеты для работы с таблицами

**booktabs** — набор макросов для тонкой настройки параметров таблиц. Цель пакета — помочь автору напечатать идеальную таблицу;

**cellspace** — гарантирует, что между текстом и горизонтальной разделительной линией всегда будет промежуток;

**colortbl** — раскраска строк, колонок и клеток таблицы;

**makecell** — пакет для тонкой настройки параметров и структуры таблицы. Для создания клеток, занимающих несколько строк, в пакете определена команда `\makecell`;

**multirow** — клетки, занимающие несколько строк. В пакет также входят стилевые файлы **bigdelim** (разделители высотой в несколько строк) и **bigstrut** (высокие подпорки);

**slashbox** — формирование клетки на перекрестии заголовка и боковика, разделённой косой чертой;

**longtable** — многостраничная таблица с заголовками по умолчанию. Не работает при многоколоночной вёрстке;

**ltxtable** из пакета **carlisle** — «смесь» **longtable** и **tabularx**;

**supertabular** — чуть более сложный в использовании и менее гибкий аналог **longtable**, зато можно применять в случае многоколоночной вёрстки;

**xtab** — современное расширение возможностей пакета **supertabular** с оглядкой на **longtable**.

#### 4.11.5. Пакеты для работы с презентациями

Великолепный и очень мощный пакет для подготовки презентаций **beamer**. Ниже перечислены лишь некоторые ключевые возможности:

- презентация генерируется в виде документа формата \*.pdf;
- слайд может выводиться как целиком, так и порциями, что легко задаётся в исходном коде;
- на всех слайдах генерируется двухуровневое оглавление презентации, в котором подсвечивается текущий слайд;
- имеется множество вариантов профессионального, хорошо продуманного дизайна презентаций, которые подключаются одной командой.

- слайды можно сопровождать пояснительным текстом, который будет показываться на дополнительном мониторе.

Пакет входит в состав многих дистрибутивов, в частности, MiKTeX.

#### 4.11.6. Пакеты для точных наук

LaTeX создавался под математику, а так как математика — язык всех точных наук, то любая из них легко может быть изложена с помощью LaTeX.

##### *Математика*

**amsmath** — базовый стиль AMS-LaTeX. Определяет дополнительные окружения для выключенных формул. При загрузке автоматически подгружает пакеты **amsbsy** (жирные математические символы), **amsopn** (создание новых математических операторов) и **amstext** (определяет команду `\text`, которая позволяет набирать обычный текст в математическом режиме). Если в документе есть формулы, то этот стиль обязателен для использования;

**amssymb** — даёт имена всем символам из шрифтов семейства AMS. Автоматически загружает пакет **amfonts** (дополнительные математические символы и шрифты). Входит в коллекцию AMS;

**amscd** — коммутативные диаграммы. Входит в коллекцию AMS;

**amsthm** — улучшенные теоремы. Определяет окружение `proof`;

**mh** — множество улучшений и дополнений для пакета **amsmath**. Пакет содержит стили **mathtools** — улучшение вывода формул, **ntheorem** — улучшение для окружения `theorem` и **empheq** — дополнительные средства визуализации для выключенных формул;

**mhequ** — печать формул в несколько колонок;

**breqn** — автоматический перенос в многострочных выключенных формулах. Одно из решений проблемы открывающей и закрывающей скобок при переносах формулы. При взаимодействии с другими пакетами случаются коллизии;

**commath** — улучшенное отображение дифференциалов, частных производных, пределов;

**deleq** — предоставляет гибкий способ нумерации выражений;

**pst-3dplot** — графики трёхмерных математических функций. Возможна загрузка данных из внешнего файла для печати трёхмерных поверхностей;

**pst-bar** — рисование гистограмм;

**pst-eucl** — построение простых и не очень геометрических чертежей;

**pst-func** — построение математических функций. В пакете предусмотрены полиномы, суммы Фурье, функции Бесселя, гауссовское распределение и многое другое;

**venn** — создание диаграмм Эйлера—Венна (MetaPost);

**wasysym** — дополнительные символы. Загрузка пакета с опцией `integrals` переопределяет символы интегралов с наклонных на прямые.

### *Информатика*

**ascii** — шрифты IBM ASCII.

Стандартные пакеты **algorithms**, **algorithmic**, **algorithmicx**, **algorithm2e**, **clrscode** и **pseudocode** специализируются на представлении алгоритмов.

Пакет **algorithm** предоставляет набор команд для записи алгоритмов в виде псевдокода, который стал стандартом де-факто за последние годы.

Пакет **algorithmic** реализует важную дополнительную возможность — оформляет алгоритмы в виде плавающих иллюстраций. Без этого запись алгоритма может быть разорвана между страницами, что очень плохо воспринимается.

Подключение пакетов осуществляется командами

```
\usepackage[ruled,section]{algorithm}
```

```
\usepackage[noend]{algorithmic}
```

Пакеты входят в состав многих дистрибутивов, в частности, MiKTeX;

**basix** — интерпретатор языка BASIC, написанный на TeX;

**cursor** — рисует L-образный курсор в математической моде;

**examplep** — печать примеров кода на LaTeX и на MetaPost. Пакет состоит из стилей **codep** (определяет окружение code, позволяющее выводить код слева, а результат компиляции справа) и **examplep** (определяет окружения и команды для печати и вывода в файл неформатированного текста);

**keystroke** — изображение клавиш клавиатуры;

**listings** — оформление программного кода;

**method** — формальное описание функции;

**pst-dbicons** — описание и прототипирование баз данных с помощью ER-диаграмм;

**pst-uml** — создание не сильно сложных UML-диаграмм;

**semantic** — нотация для описания семантического разбора и компиляции программного кода, включая T-диаграммы.

### *Физика*

Пакеты шрифтов **astro** и **cmastro** предоставляют доступ к астрономическим символам, включая знаки зодиака.

**braket** — бра-кет нотация Дирака;

**circ** — создание не сильно сложных электрических схем средствами LaTeX и Metafont;

**feyn** — простые фейнмановские диаграммы;

**feynmf** — набор макросов для создания сложных фейнмановских диаграмм средствами LaTeX и Metafont/MetaPost;

**hepparticles** — набор макросов для печати названий частиц для Физики высоких энергий;

**isotope** — макрос для печати названий изотопов химических элементов;

**pst-circ** — простые электрические схемы;

**pst-optic** — создание иллюстрации для геометрической оптики;

**pst-osci** — иллюстрации осциллограмм;

**SIstyle** — единицы физических величин по правилам СИ. Схожую функциональность предоставляет пакет **SIunits**;

**slashed** из пакета **carlisle** — слэш-нотация Фейнмана. Пакет определяет макрос `\slashed`, который перечёркивает переданный ему символ в математической моде;

**timing** — рисование простых временных диаграмм средствами LaTeX и Metafont;

**units** — печать единиц физических величин. Есть альтернативный способ представления дроби, а также проблемы со шрифтами;

**unitsdef** — печать единиц физических величин в том числе и по правилам СИ.

### *Химия*

**bpchem** — печать химических элементов, названий, формул и нумерация химических соединений;

**chemarr** из пакета **oberdiek** — дополнительный набор стрелок для химиков;

**chemarrow** — ещё один дополнительный набор стрелок;

**chemcompounds** — простая последовательная нумерация химических соединений. Возможно присвоение соединению своего уникального имени;

**chemcono** — нумерация химических соединений по принципу организации библиографических ссылок;

**cryst** — набор символов для описания типа симметрии в кристаллографии;

**mhchem** — продвинутый пакет для печати названий химических элементов, формул и значков для маркировки опасных химических соединений (возможна локализация);

**pst-labo** — создание иллюстраций для лабораторных работ с помощью пакета **pstricks**;

**r-und-s** — маркировка опасных химических соединений (возможна локализация);

**xymtex** — пакет для отображения средствами LaTeX множества химических структурных формул. Превосходит устаревший пакет **chemtex** со схожей функциональностью по всем параметрам.

### *Биология*

**biocon** — создание коротких имён для биологических видов;

**dichokey** — создание дихотомических идентификационных ключей для целей классификации вида;

**dnaseq** — вывод простых ДНК-последовательностей;

**pst-pdgr** — родословное дерево для нужд медиков;

**texshade** — выделение одинаковых последовательностей нуклеотидов и пептидов (nucleotide and peptide alignments);

**textopo** — расцветка и подписи для рисунков, отображающих геометрическую структуру структурных белков (membrane protein topology plots);

**labbook** — класс для создания рабочего журнала для биологов с хронологическими метками о проведении эксперимента.

#### 4.11.7. Пакеты для гуманитарных направлений

*Поэзия, пьесы и критика*

**dramatist** — вывод пьесы в стихах или прозе;

**ednotes** — печать комментариев;

**ledmac** — набор макросов для формирования издания с комментариями. Базируется на TeX-пакете **edmac**. Имеется расширение в виде пакета **ledpar** для печати параллельного текста;

**play** — простой набор окружений и макросов для набора пьес;

**parrun** — пакет, позволяющий печатать текст и его перевод один над другим;

**poemscol** — пакет, ориентированный на создание сборников стихов и критических заметок;

**sides** — класс с небольшим набором специализированных макросов для оформления текста пьесы. Базируется на более простом классе **plari**;

**stage** — класс для создания пьес;

**verse** — сочиняем стихи.

*Языкознание*

Для рисования различного вида деревьев имеет смысл взглянуть на **pst-tree** — набор макросов для **pstricks**;

**arcs** — позволяет рисовать дуги под (`\underarc`) и над (`\overarc`) текстом;

**cbcoptic** — набор текстов на коптском;

**covington** — набор макросов, позволяющих создавать сложные множественные акценты, трансляцию текста слово-в-слово, нумерацию текстовых примеров, описывать структуру предложения и множество других мелких, но полезных для лингвиста вещей;

**engpron** — набор макросов, позволяющих набирать транскрипцию слов, как это сделано в словаре Даниэля Джонса (Daniel Jones «English Pronouncing Dictionary»);

**lexikon** — набор простых макросов для создания двуязычного словаря;

**linguex** — оформление примеров;

**phonetic** — дополнительные фонетические символы;

**rst** — анализ структуры речи;

**synttree** — синтаксическое дерево;

**tipa** — шрифты и набор макросов для отображения международной фонетической транскрипции (International Phonetic Alphabet);

**teubner** — расширение **babel**, загруженного с опцией `greek` для набора на классическом греческом;

**xytree** — формирование лингвистического синтаксического дерева.

### *Музыка*

**abc** — поддержка LaTeX ABC-нотации (запись музыки ASCII-символами). Требуется установки сторонней программы **abcm2ps**;

**ConcProg** — класс для написания концертных программ. Класс входит в состав пакета **concprog**;

**cd** и **cd-cover** — классы для создания обложек CD-дисков;

**guitar** — подпись гитарных аккордов над текстом песен;

**Musictex** — очень мощный пакет, ориентированный на набор нот средствами LaTeX;

**musixlyr** — расширение **musixtex** для добавления текстов песен;

**songbook** — формирование песенных сборников с возможностью добавления аккордов над текстом.

## 4.12. Редактор TEXSTUDIO

TeXstudio — оболочка для LaTeX позволяющая с легкостью создавать LaTeX документы и управлять ими. Она имеет современный текстовый редактор с автоматической проверкой орфографии, группировку кода и подсветку синтаксиса. Оболочка также обеспечивает взаимодействие со всеми инструментами LaTeX. Ниже приведено описание TeXstudio 2.2. Интерфейс данной версии программы изображен на рисунке 4.2.

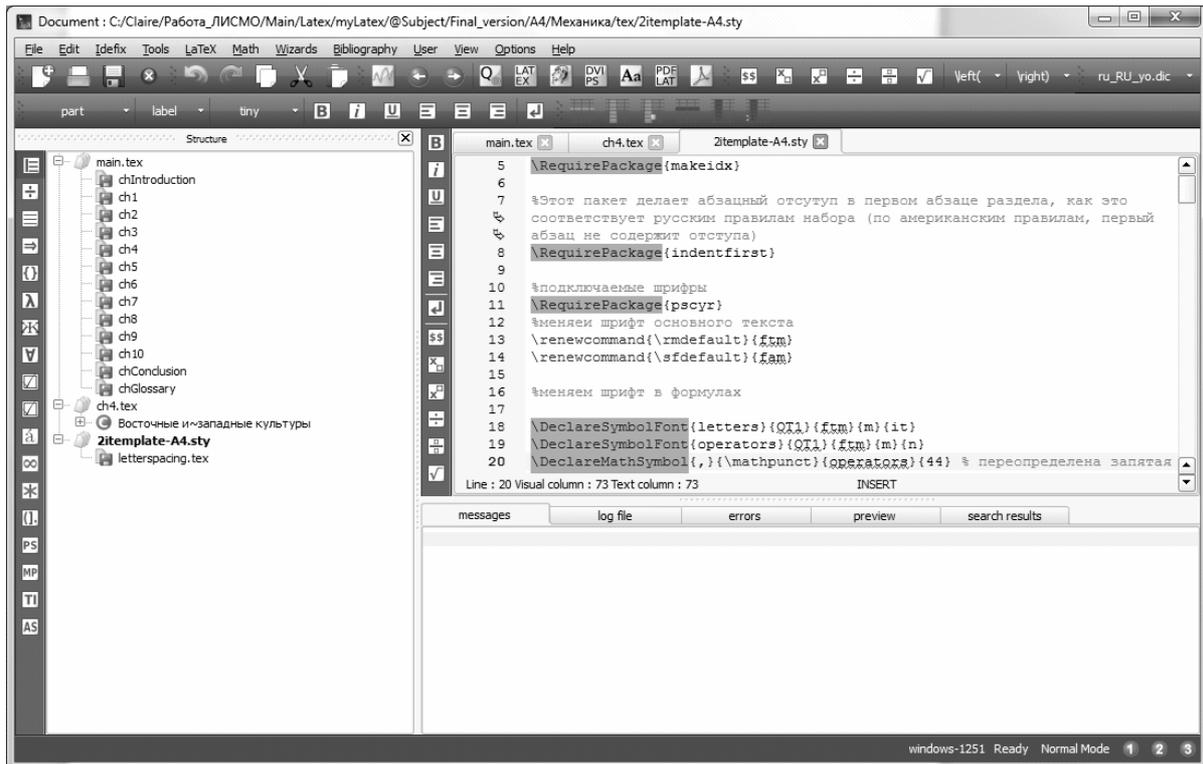


Рис. 4.2 — Интерфейс программы TexStudio

Кросс-платформенность позволяет использовать TeXstudio в операционных системах Windows, Linux, BSD и MacOSX.

Изначально TeXstudio называлась TexMakerX и была модификацией Texmaker, расширяющей возможности оригинальной программы дополнительными функциями, но с неизменным визуальным оформлением.

#### 4.12.1. Редактирование документов TeX

##### *Обычные команды*

Стандартные команды (вырезать, скопировать, найти) можно выполнить через пункт меню «Edit» и панель инструментов «Edit» (см. рис. 4.3).



Рис. 4.3 — Панель инструментов «Edit»

### *Задание преамбулы для документов TeX*

Для задания преамбулы документа можно воспользоваться помощником «Quick start» (Меню «Wizards»). Этот диалог показан на рисунке 4.4. Он позволяет задать главные особенности документа (класс, размер бумаги, кодировку и т.д.).

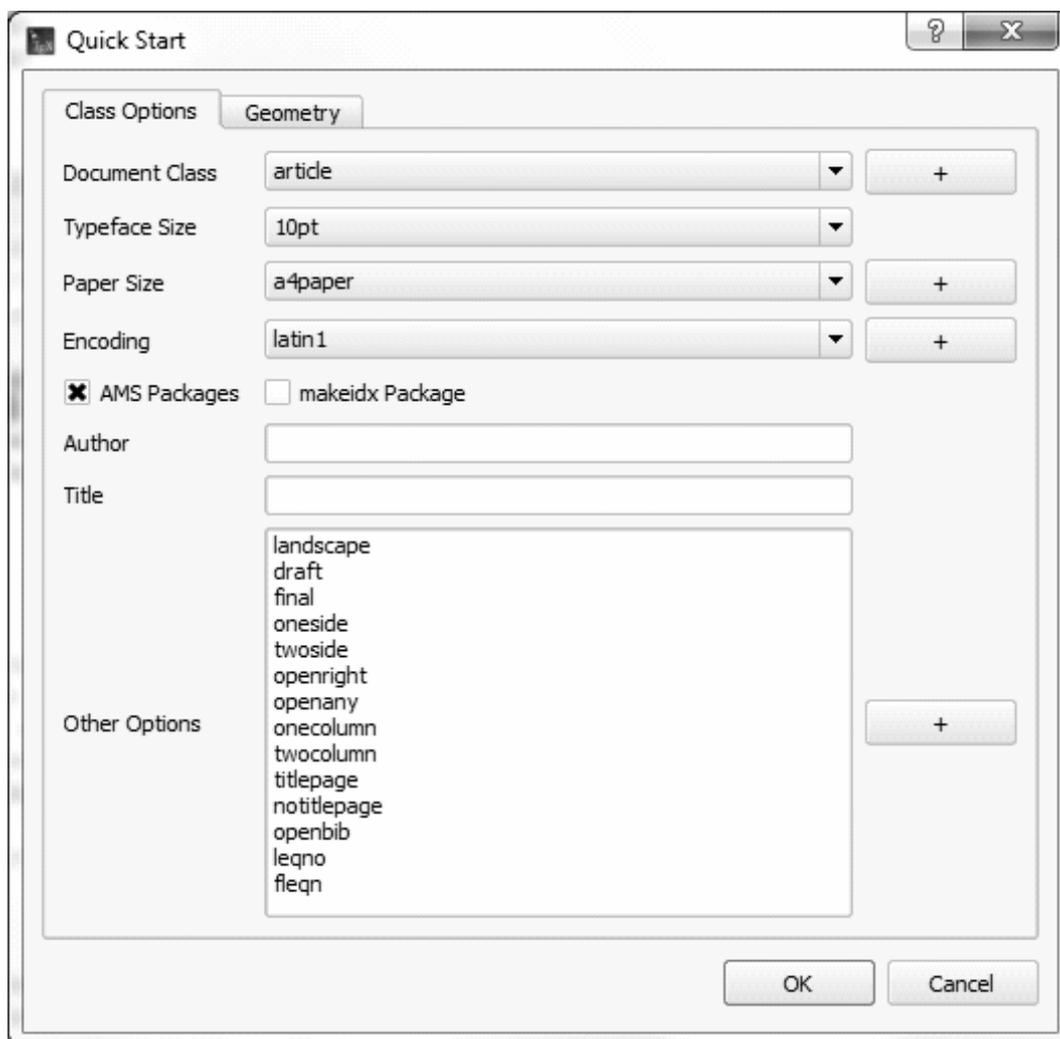


Рис. 4.4 — Создание преамбулы документа

### *Структура документа*

Для того чтобы задать новую часть документа (раздел, подраздел и т.д.) с помощью TeXstudio, можно использовать выпадающий список в панели инструментов (см. рис. 4.5).



Рис. 4.5 — Выбор раздела документа

### Просмотр документа

Левая панель «Structure» позволяет быстро перемещаться между частями документа (см. рис. 4.6). Для перехода к началу желаемой области в редакторе необходимо кликнуть на соответствующий пункт (метку, раздел и т.п.) на панели «Structure».

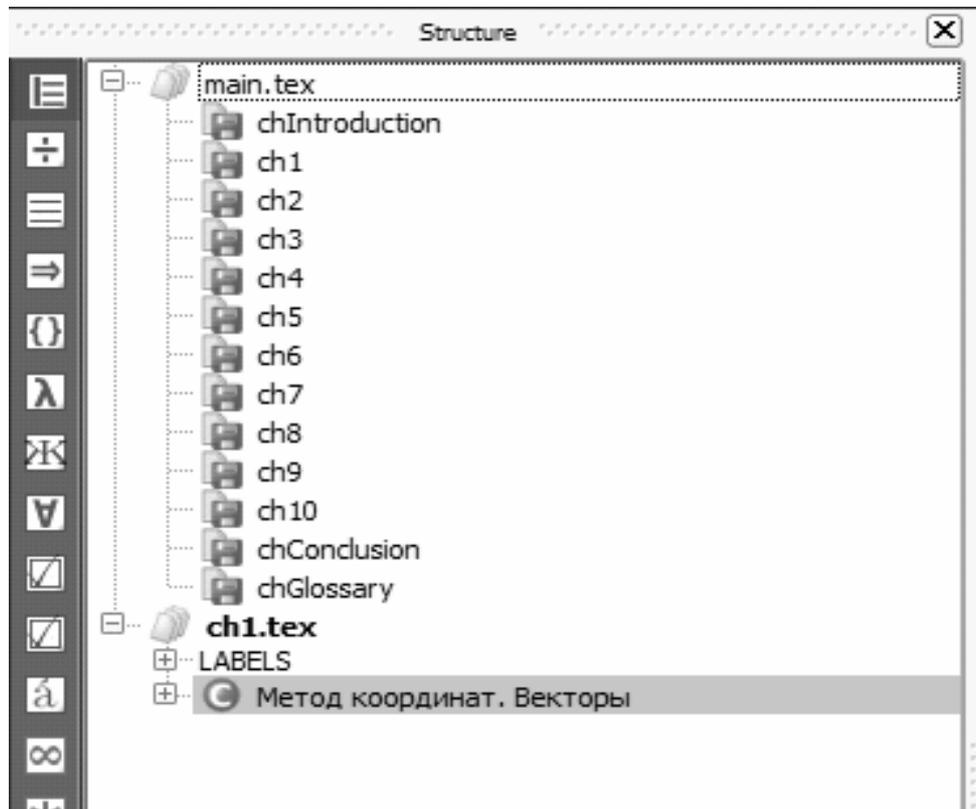


Рис. 4.6 — Панель «Structure»

Панель «Structure» автоматически обновляется при наборе документа. Существует возможность обновить данную панель в любой момент времени (меню «Idefix — Refresh Structure»).

В каждом файле, для ускоренной навигации можно задавать до трех закладок. Для того чтобы добавить или удалить закладку необходимо просто кликнуть слева от номера строки (см. рис. 4.7). Если три закладки уже заданы, то для добавления новой, необходимо удалить одну из существующих. Чтобы перейти к строке, отмеченной закладкой в редакторе, можно просто кликнуть на кнопку с номером закладки в строке состояния, которая расположена в правом нижнем углу окна программы.

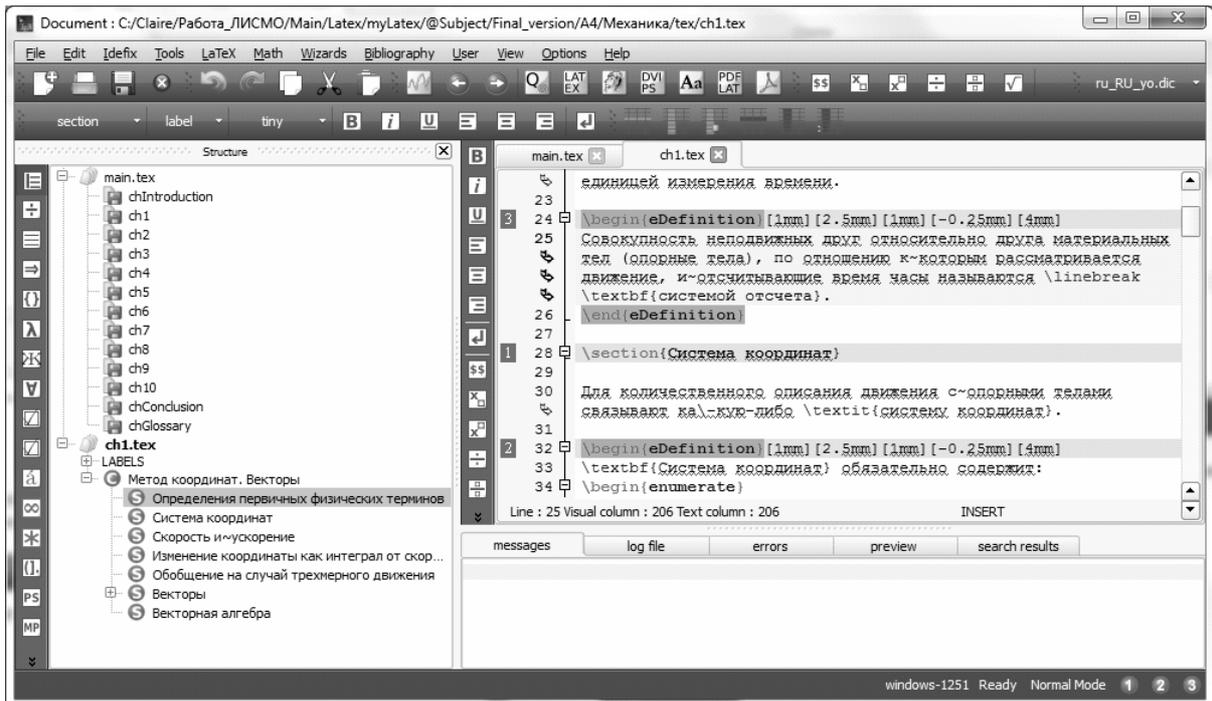


Рис. 4.7 — Использование закладок для навигации по документу

### Форматирование текста

Для быстрого форматирования текста можно воспользоваться панелями инструментов, которые изображены на рисунке 4.8.



Рис. 4.8 — Панель инструментов

Например, выделенный текст может быть быстро заключен в определенное окружение или команду. Кликнув на кнопку «B» после выделения слова «Здравствуй», получим код: `\textbf{Здравствуй}`. Эта возможность доступна для всех окружений помеченных «[selection]» в меню «LaTeX».

### Вставка списка

Код обычного окружения списка можно быстро вставить через меню «LaTeX — List Environments». Можно использовать горячие клавиши команды «\item» — `Ctrl+Alt+N`.

### *Вставка таблицы*

Окружение `tabular` можно быстро вставить, используя помощник «Quick Tabular» (меню «Wizards»). Данный диалог позволяет вставлять текст непосредственно в ячейки (см. рис. 4.9). Соответствующий код автоматически вставляется в редактор.

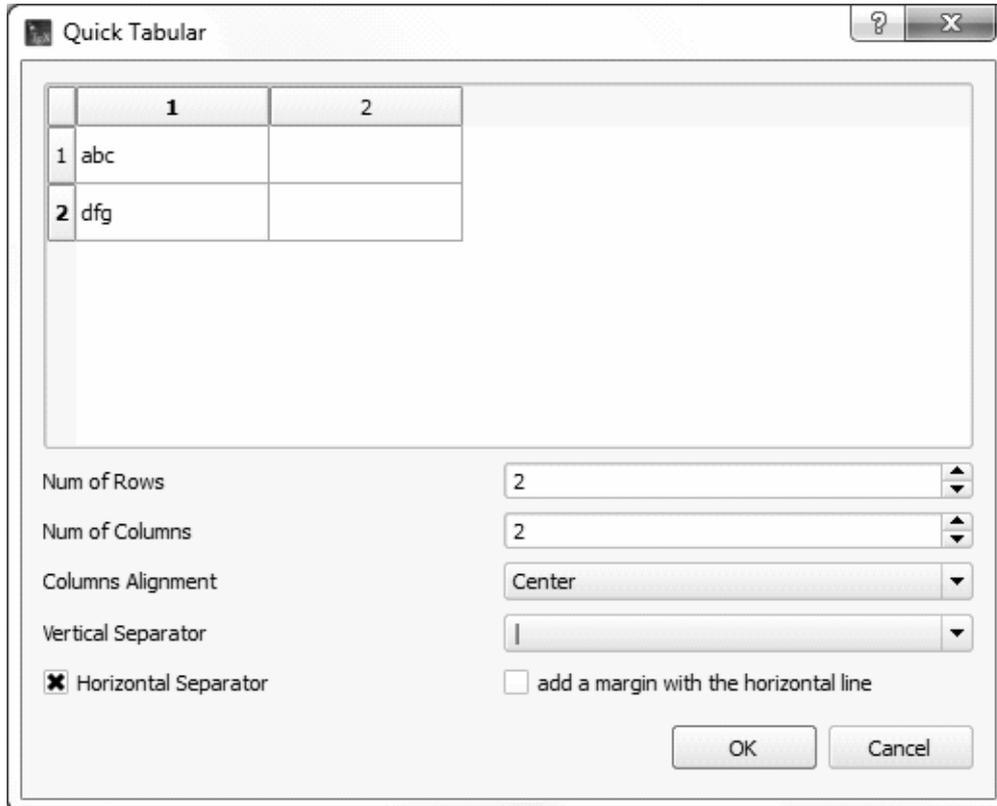


Рис. 4.9 — Помощник «Quick Tabular»

### *Вставка картинки*

Для того чтобы вставить картинку в документ, необходимо просто воспользоваться командой `\includegraphics` в меню «LaTeX». Появится диалоговое окно выбора файла (см. рис. 4.10).



Рис. 4.10 — Вставка изображения

### Перекрестные ссылки и сноски

Выпадающий список на панели инструментов, изображенный на рисунке 4.11, позволяет быстро вставить код `\label`, `\cite`, `\ref`, `\footnote` и др. Все метки (label) использовавшиеся в документе отображаются на панели «Structure». Для команды `\ref`, диалог позволяет непосредственно выбрать метку «label».



Рис. 4.11 — Вставка перекрёстных ссылок и сносок

### Вставка математических формул

Можно включать и выключать окружение «Inline math mode» кнопкой « $\$$ » на панели инструментов или в меню «Math» или использовать сочетание клавиш `Ctrl+Alt+M`. Сочетание клавиш для окружения «Display math mode» — `Alt+Shift+M`. Панель инструментов «Math» позволяет вставлять наиболее часто используемые математические формулы (см. рис. 4.12).

С помощью различных панелей символов, расположенных под панелью «Structure» можно вставить коды 400 математических символов и левые и правые скобки (см. рис. 4.13).

Панель инструментов «Favorites»  позволяет создать свою собственную панель символов. Чтобы добавить символ в панель инструментов «Favorites» необходимо кликнуть правой кнопкой мыши на символе в другой панели символов и выбрать «Add to favorites» в всплывающем меню. Чтобы удалить символ из панели инструментов «Favorites» необходимо кликнуть на нем правой кнопкой мыши и выбрать «Remove from favorites» во всплывающем меню.

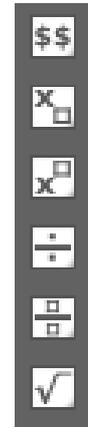


Рис. 4.12 — Панель инструментов «Math»

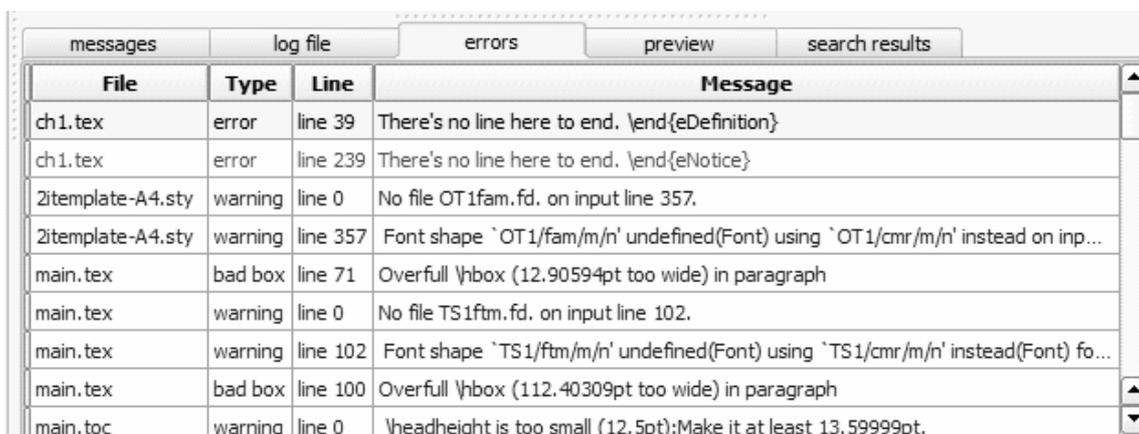


Рис. 4.13 — Вставка символов

### Компиляция документа

Самый простой способ скомпилировать документ — использовать команду «Quick Build»  на панели инструментов «Tools» или из меню «Tools — Quick Build». Можно задать последовательность команд используемую командой «Quick Build» (меню «Options — Configure TeXstudio» вкладка «Quick Build»).

При выполнении команды «Quick Build» все сообщения о ходе процесса отображаются в панели «Messages/Log file» (см. рис. 4.14). При щелчке на число в колонке «Line», курсор перемещается на соответствующую строку в редакторе и показывается сообщение об ошибке.



File	Type	Line	Message
ch1.tex	error	line 39	There's no line here to end. \end{eDefinition}
ch1.tex	error	line 239	There's no line here to end. \end{eNotice}
2itemplate-A4.sty	warning	line 0	No file OT1fam.fd. on input line 357.
2itemplate-A4.sty	warning	line 357	Font shape `OT1/fam/m/n' undefined(Font) using `OT1/cmr/m/n' instead on inp...
main.tex	bad box	line 71	Overfull \hbox (12.90594pt too wide) in paragraph
main.tex	warning	line 0	No file TS1ftm.fd. on input line 102.
main.tex	warning	line 102	Font shape `TS1/ftm/m/n' undefined(Font) using `TS1/cmr/m/n' instead(Font) fo...
main.tex	bad box	line 100	Overfull \hbox (112.40309pt too wide) in paragraph
main.toc	warning	line 0	\headheight is too small (12.5pt);Make it at least 13.59999pt.

Рис. 4.14 — Панель «Messages/Log file»

## 5. СИСТЕМА MAXIMA

### 5.1. Системы численных и символьных вычислений

Численные методы решения математических задач явились основой для применения и развития современных компьютеров. Однако решение задачи в численном виде во многих случаях является недостаточным, а в ряде случаев, и вовсе не приводит к решению. Поэтому наряду с численными методами решения задач развивались методы символьных вычислений. Простейшей задачей символьных вычислений является задача упрощения математических выражений. Например, сложение с нулем, умножение на ноль, умножение на единицу, сокращение подобных членов, раскрытие скобок. Для реализации таких алгоритмов используются механизмы представления математических выражений в виде абстрактных синтаксических деревьев и манипулирование ими. Дальнейшее развитие этого направления породило компьютерную алгебру — науку на стыке алгебры и теории вычислений.

Одной из первых систем компьютерной алгебры принято считать систему Reduce, предназначенная для решения физических задач. В настоящее время широко используются системы Reduce, Mathematica, Maple, Pari, Maxima и др.

Обычно эти пакеты поддерживают следующие действия:

- 1) упрощение математических выражений или приведение к стандартному виду;
- 2) подстановка символьных и численных значений в выражения;
- 3) преобразование выражений: раскрытие скобок, произведений и степеней, частичная и полная факторизация (разложение на множители);
- 4) разложение на простые дроби, запись тригонометрических функций через экспоненты и т.д.;
- 5) дифференцирование;
- 6) нахождение неопределённых и определённых интегралов
- 7) решение задач оптимизации: нахождение глобальных экстремумов, условных экстремумов и т.д. в виде выражений;
- 8) решение линейных и нелинейных уравнений;

- 9) алгебраическое решение дифференциальных и конечно-разностных уравнений;
- 10) нахождение пределов функций и последовательностей;
- 11) интегральные преобразования;
- 12) оперирование со степенными рядами: суммирование, умножение, суперпозиция и т.д.;
- 13) матричные операции: обращение, факторизация, решение спектральных задач и т.д.;
- 14) статистические вычисления;
- 15) автоматическое доказательство теорем;
- 16) операции над производящими функциями.

## 5.2. Об истории системы Maxima

Система Maxima является развитием известной системы Macsyma (MAC Symbolic MAnipulation), которая была разработана в Массачусетском технологическом институте, в рамках существовавшего в 60-е годы 20 столетия большого проекта MAC. В качестве языка для разработки системы был выбран Lisp. Принципы, положенные в основу системы Macsyma, позднее были заимствованы наиболее активно развивающимися ныне коммерческими программами — Mathematica и Maple. Система Macsyma была закрытым коммерческим проектом, который просуществовал до 1999г. В 1998 году профессор Уильям Шелтер получил права на публикацию кода по лицензии GPL Macsyma и стал развивать свой проект под названием Maxima. В настоящее время Maxima выпускается под две платформы: Unix-совместимые системы и MS Windows. Далее будем рассматривать wxMaxima (Maxima для MS Windows).

## 5.3. Графический интерфейс системы Maxima

В редакторе wxMaxima формулы вводятся в текстовом виде, а отображается в виде математических формул.

На рис. 5.1 показан пример ввода/вывода: (%i1) текстовый ввод матрицы  $x$ , (%o1) текстовый вывод матрицы  $x$ , (%i2) текстовый ввод матрицы  $y$ , (%o2) текстовый вывод матрицы  $y$ , (%i1)

текстовый ввод операции сложения матриц  $x+y$ , (%o3) текстовый вывод результирующей матрицы.

Для вызова помощи используется клавиша F1.

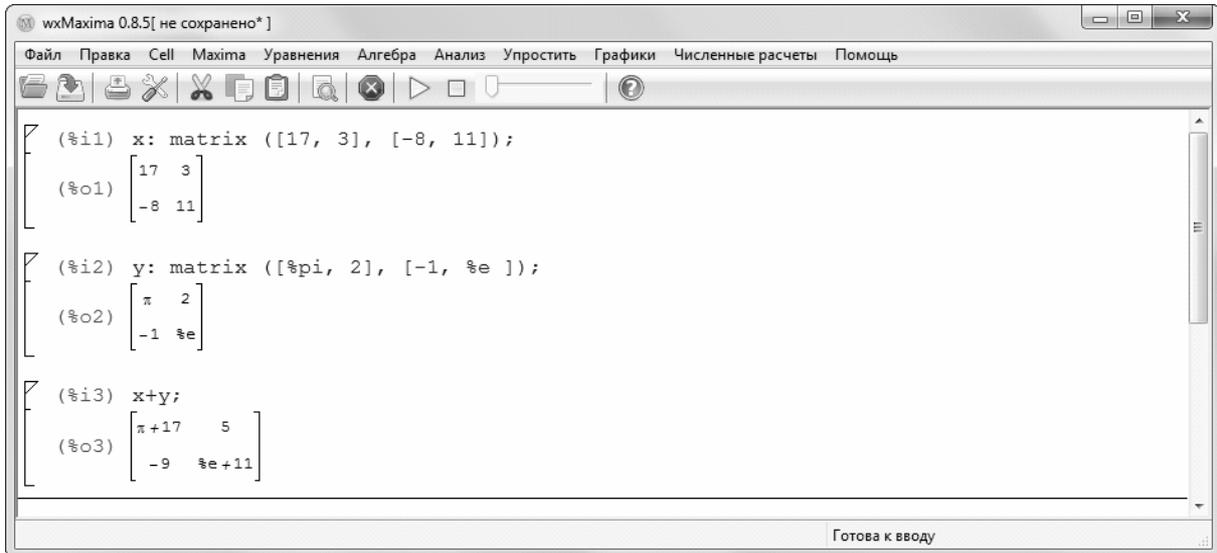


Рис. 5.1 — Окно системы Maxima

## 5.4. Выражения

Выражения в системе строятся из идентификаторов, констант, знаков операций, функций и разделителей. Ниже представлены зарезервированные ключевые слова, которые не должны быть в выражении.

integrate	next	from	diff
in	at	limit	sum
for	and	elseif	then
else	do	or	if
unless	product	while	thru
step			

Все выражения должны заканчиваться точкой с запятой или знаком  $\$$ . Ниже представлено примеры выражений.

```
(%i1) x: 3$
(%i2) (if (x > 17) then 2 else 4);
(%o2) 4
```

Переменной  $x$  присвоить значение 3 (%i1).

```
(%i1) y: (x: 1, for i from 1 thru 10 do (x: x*i))$
(%i2) y;
(%o2)                done
```

Вычислить значение  $x$ , используя цикл для  $i$  от 1 до 10. Результат присвоить переменной  $y$ . Идентификатор — это последовательность букв, цифр или знака «\_» начинающийся с буквы. Знаки сравнения  $>$  (больше),  $<$  (меньше),  $>=$  (больше или равно),  $<=$  (меньше или равно),  $=$  (равно),  $\neq$  (не равно). Строки символов помечаются двойными кавычками.

### Таблица констант

Название	Обозначение
$+\infty$ (плюс бесконечность)	inf
$-\infty$ (минус бесконечность)	minf
Комплексная бесконечность	infinity
$\pi$ (число Пи)	%pi
$e$ (экспонента)	%e
Мнимая единица ( $i$ )	%i
Истина	true
Ложь	false
Золотое сечение $(1 + \sqrt{5})/2$	%phi

## 5.5. Массивы

В системе можно создавать и использовать различные массивы, которые создаются как контейнеры. Обращение к элементам массива производится посредством указания индексов. Например, для матрицы — указание номеров строки и столбца. Массив можно создать с помощью функции системы `array`:

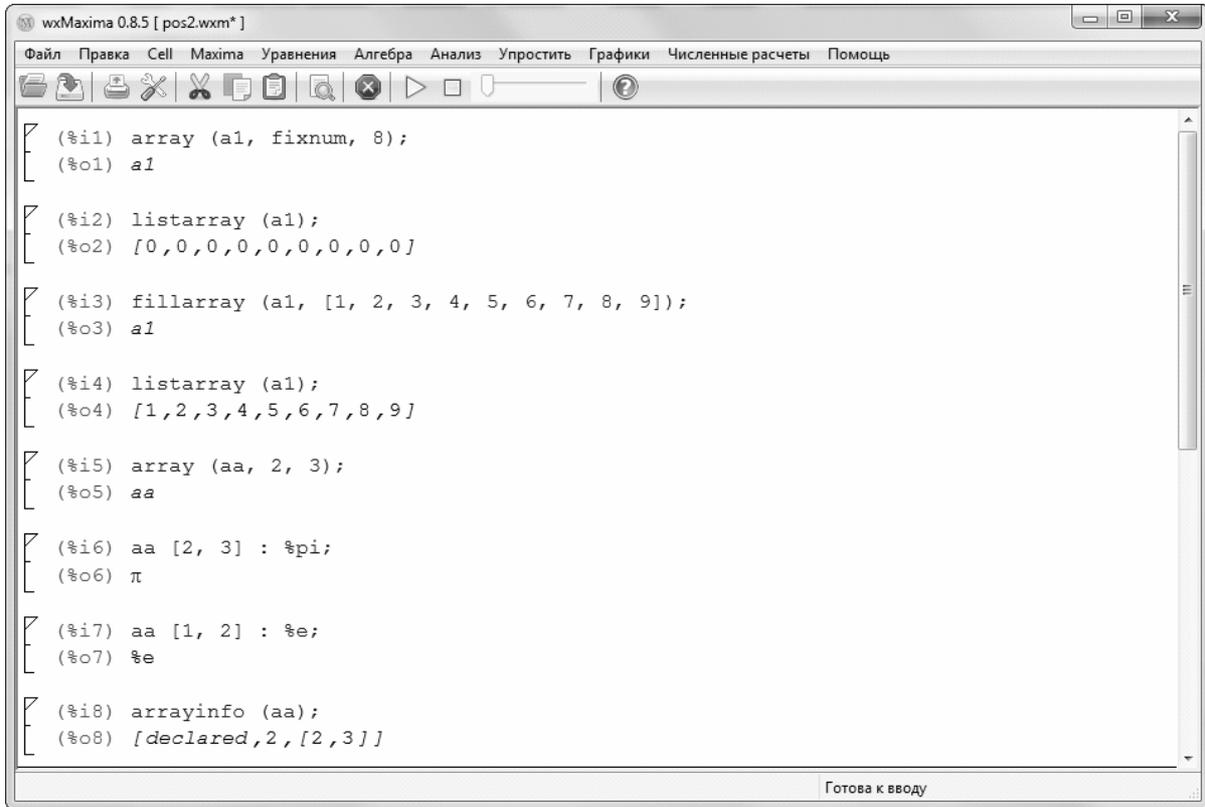
1) `array (name, dim_1, ..., dim_n)`

2) `array (name, type, dim_1, ..., dim_n)`

3) `array ([name_1, ..., name_m], dim_1, ..., dim_n)`

В первом случае создается обычный массив *name*, размерностью  $dim_1, \dots, dim_n$  (не более 5), элементами которого могут

быть любые объекты и выражения. Во втором случае создается специальные массивы, для `type= fixnum` хранит только целые, для `type= flonum` хранит только вещественные числа. В третьем случае создается `n` обычных массивов, одинаковой размерностью.



```

wxMaxima 0.8.5 [ pos2.wxmx* ]
Файл  Правка  Cell  Maxima  Уравнения  Алгебра  Анализ  Упростить  Графики  Численные расчеты  Помощь

[ (%i1) array (a1, fixnum, 8);
  (%o1) a1

[ (%i2) listarray (a1);
  (%o2) [0,0,0,0,0,0,0,0]

[ (%i3) fillarray (a1, [1, 2, 3, 4, 5, 6, 7, 8, 9]);
  (%o3) a1

[ (%i4) listarray (a1);
  (%o4) [1,2,3,4,5,6,7,8,9]

[ (%i5) array (aa, 2, 3);
  (%o5) aa

[ (%i6) aa [2, 3] : %pi;
  (%o6) π

[ (%i7) aa [1, 2] : %e;
  (%o7) %e

[ (%i8) arrayinfo (aa);
  (%o8) [declared,2,[2,3]]

Готова к вводу
  
```

Рис. 5.2

## 5.6. Списки

Списки в систем создаются функцией `makelist`:

1) `makelist (expr, i, i_0, i_1)`

2) `makelist (expr, x, list)`

В первом случае организуется цикл для переменной `i` от значений `i_0` до `i_1` с шагом 1 и вычисляемое значение `expr` заносится в список. Во втором случае переменная `x` последовательно принимает значение из списка `list`. Результатом работы функции является список. Примеры использования списков показаны на рис. 5.3.

```

wxMaxima 0.8.5 [ Pos1.wxm* ]
Файл  Правка  Cell  Maxima  Уравнения  Алгебра  Анализ  Упростить  Графики  Численные расчеты  Помощь

[ (%i1) makelist (concat (x, i), i, 1, 6);
  (%o1) [ x1, x2, x3, x4, x5, x6 ]

[ (%i2) L:makelist (fib(n), n, 1, 10);
  (%o2) [ 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 ]

[ (%i3) L[1]+L[2]+L[3];
  (%o3) 4

[ (%i4) makelist (x=y, y, [a, b, c]);
  (%o4) [ x=a, x=b, x=c ]

Готова к вводу

```

Рис. 5.3 — Примеры использования списков

## 5.7. Упрощение выражений

В системе имеется множество функций, предназначенных для упрощения выражений. Ниже представлено использование двух функций `expand` и `ratsimp`.

```

wxMaxima 0.8.5 [ не сохранено* ]
Файл  Правка  Cell  Maxima  Уравнения  Алгебра  Анализ  Упростить  Графики  Численные расчеты  Помощь

[ (%i6) expr: (x+1)^2*(y+1)^3;
  (%o6) (x+1)^2 (y+1)^3

[ (%i7) expand(expr);
  (%o7) x^2 y^3 + 2 x y^3 + y^3 + 3 x^2 y^2 + 6 x y^2 + 3 y^2 + 3 x^2 y + 6 x y + 3 y + x^2 + 2 x + 1

[ (%i8) expand(expr, 2);
  (%o8) x^2 (y+1)^3 + 2 x (y+1)^3 + (y+1)^3

[ (%i9) sin (x/(x^2 + x)) = exp ((log(x) + 1)^2 - log(x)^2);
  (%o9) sin( x / (x^2 + x) ) = %e^(log(x)+1)^2 - log(x)^2

[ (%i10) ratsimp(%);
  (%o10) sin( 1 / (x+1) ) = %e x^2

[ (%i11) ((x - 1)^(3/2) - (x + 1)*sqrt(x - 1))/sqrt((x - 1)*(x + 1));
  (%o11) (x-1)^(3/2) - sqrt(x-1)(x+1) / sqrt((x-1)(x+1))

[ (%i12) ratsimp (%);
  (%o12) -2*sqrt(x-1) / sqrt(x^2-1)

Готова к вводу

```

Рис. 5.4

## 5.8. Решение уравнений

Для решения уравнений используется функция `solve`. Примеры для решения квадратных и кубических уравнений, причем кубическое уравнение имеет параметр  $a$ .

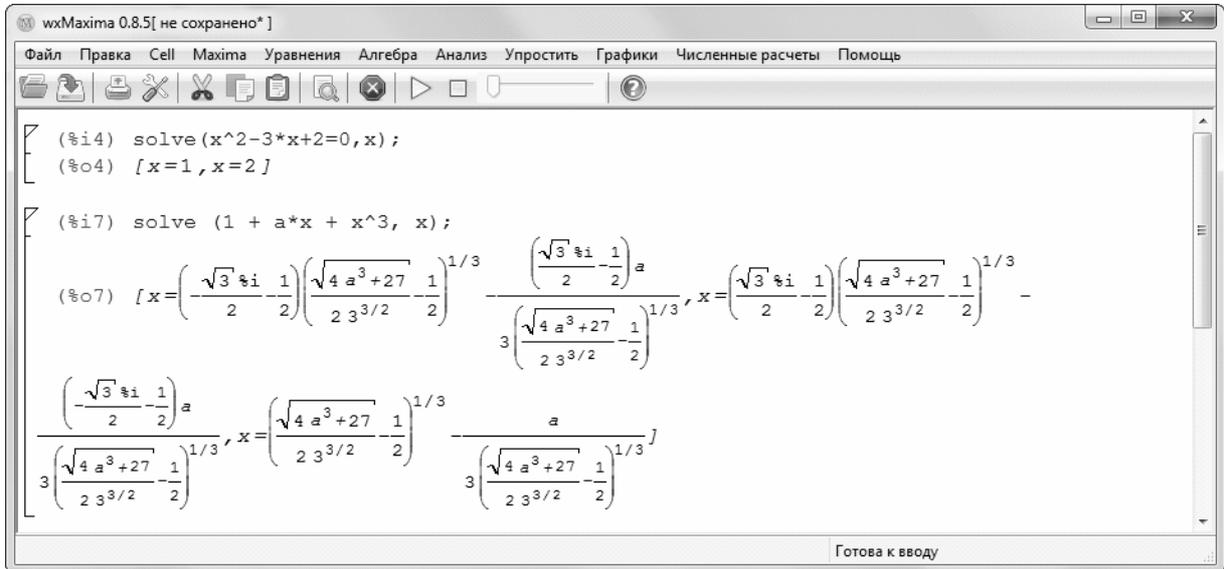


Рис. 5.5

## 5.9. Интегрирование

Для выполнения операций интегрирования в системе существует функция `integrate`:

- 1) `integrate (expr, x)`
- 2) `integrate (expr, x, a, b)`

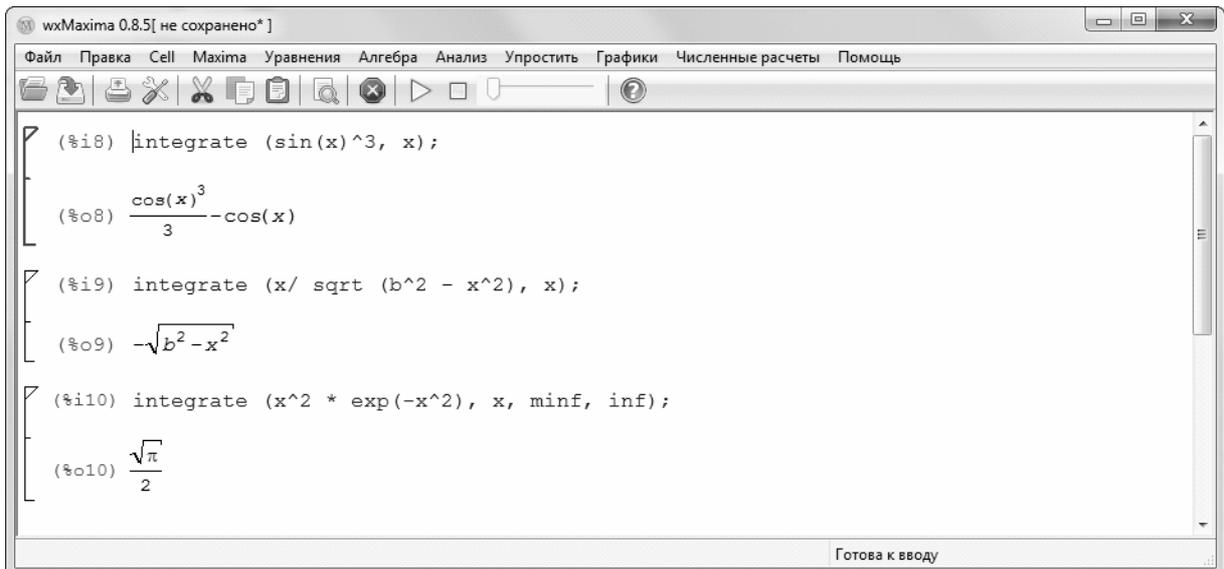


Рис. 5.6 — Примеры интегрирования

Первый вариант предназначен для получения выражения неопределенного интеграла, второе — для получения определенного интеграла, параметры  $a$  и  $b$  задают пределы интегрирования. На рис. 5.6 приведены примеры получения выражений определенных и неопределенных выражений.

## 5.10. Нахождение пределов

Для нахождения пределов используется функция `limit`:

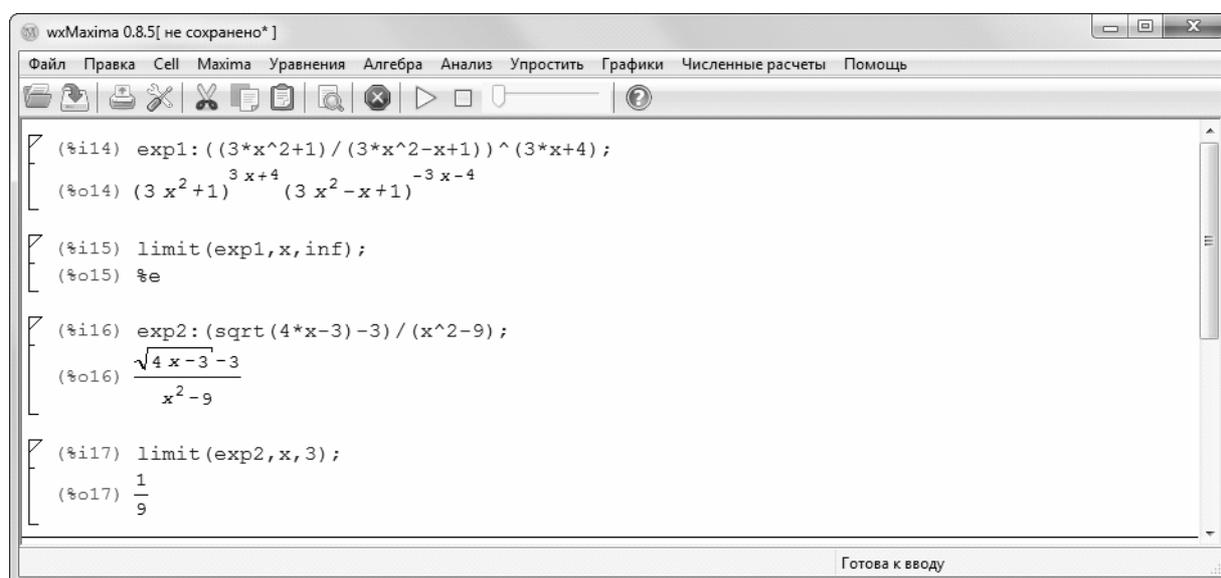


Рис. 5.7

## 5.11. Разложение функции в ряд

Для разложения функции в ряд Тейлора используется функция системы `taylor`:

**taylor** (*expr*, *x*, *a*, *n*), где *expr* — выражение функции, *x* — переменная функции, *a* — точка, в которой производится разложение, *n* — число членов ряда. Возвращает список, содержащий выражение членов. На рис. показаны примеры использования функции `taylor`. Функция `rade` предназначена для аппроксимации ряда Тейлора рациональной функцией.

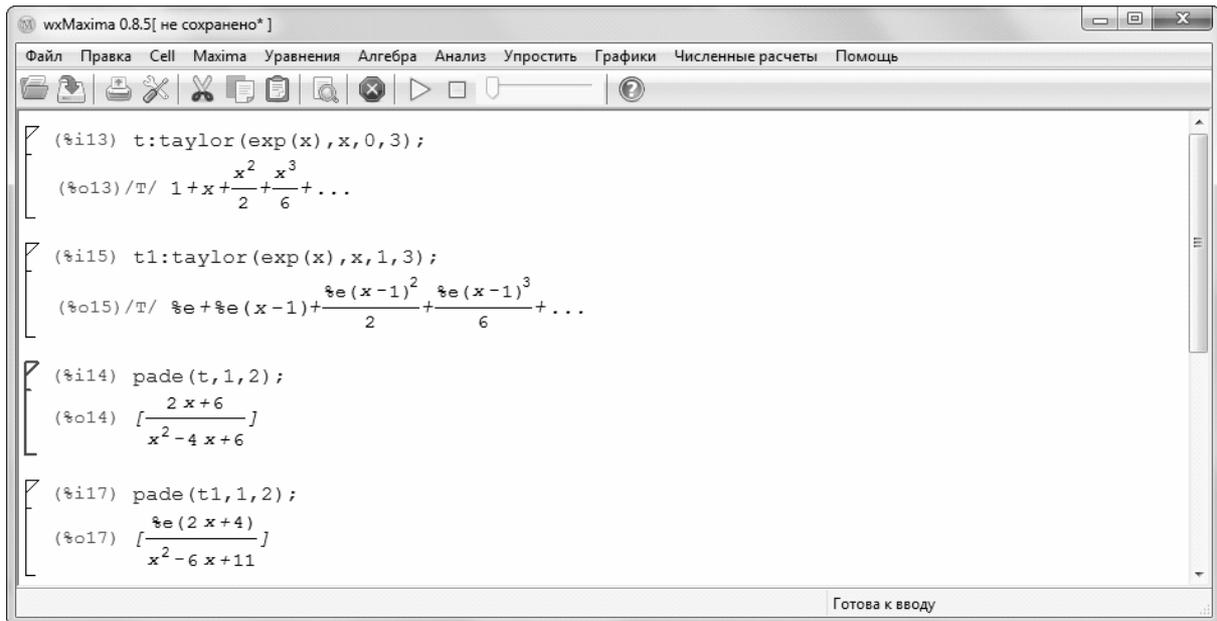


Рис. 5.8

## 5.12. Программирование в системе Maxima

### 5.12.1. Операторы цикла

В системе имеется три варианта организации цикла

- 1) **for** *variable*: *initial\_value* **step** *increment* **thru** *limit* **do** *body*
- 2) **for** *variable*: *initial\_value* **step** *increment* **while** *condition* **do** *body*
- 3) **for** *variable*: *initial\_value* **step** *increment* **unless** *condition* **do** *body*

### 5.12.2. Условный оператор

**if** *cond\_1* **then** *expr\_1* **else** *expr\_0* evaluates to *expr\_1* if *cond\_1* evaluates to true, otherwise the expression evaluates to *expr\_0*.

Operation	Symbol	Type
Меньше чем	<	сравнение
Меньше или равно	<=	сравнение
Равно	=	сравнение
Отрицание равно =	#	сравнение
Равенство	equal	функция отношения
Не равенство	notequal	функция отношения
Больше или равно	>=	сравнение
Больше чем	>	сравнение
and	and	логическая
or	or	логическая
not	not	логическая

### 5.12.3. Функция map

Оператор **map** предназначен для получения отображения. Правила записи следующие

```
(%i1) map(f, x+a*y+b*z);
(%o1) f(b z)+f(a y)+f(x)

(%i2) map(ratsimp, x/(x^2+x)+(y^2+y)/y);
(%o2) y + 1/(x+1) + 1

(%i3) map("=", [a,b], [-0.5,3]);
(%o3) [a=-0.5, b=3]
```

Рис. 5.9

### 5.12.4. Функция block

Функция **block** предназначена для объединения нескольких выражений  $\text{expr}_1 \dots \text{expr}_n$  в локальный блок. Правила записи следующие:

**block** ( $[v_1, \dots, v_m], expr_1, \dots, expr_n$ )

Здесь первым аргументом стоит список локальных переменных. При работе функции **block** формируется список локальных переменных, затем последовательно выполняются операторы, записанные в выражениях. Последовательный порядок выражений может быть нарушен операторами **goto** или **return**.

### 5.12.5. Создание собственных функций

Простые функции в системе создаются с помощью операции «:=». Например,

- 1)  $A(x) := x + x^2$ .
- 2)  $A(x) := \text{block}([s,i], s:0, \text{for } i:0 \text{ thru } 10 \text{ do}( s:s+1), \text{return}(s));$

### 5.12.6. Вывод графиков

```
(%i1) plot2d (sin(x), [x, -%pi, %pi])$
```

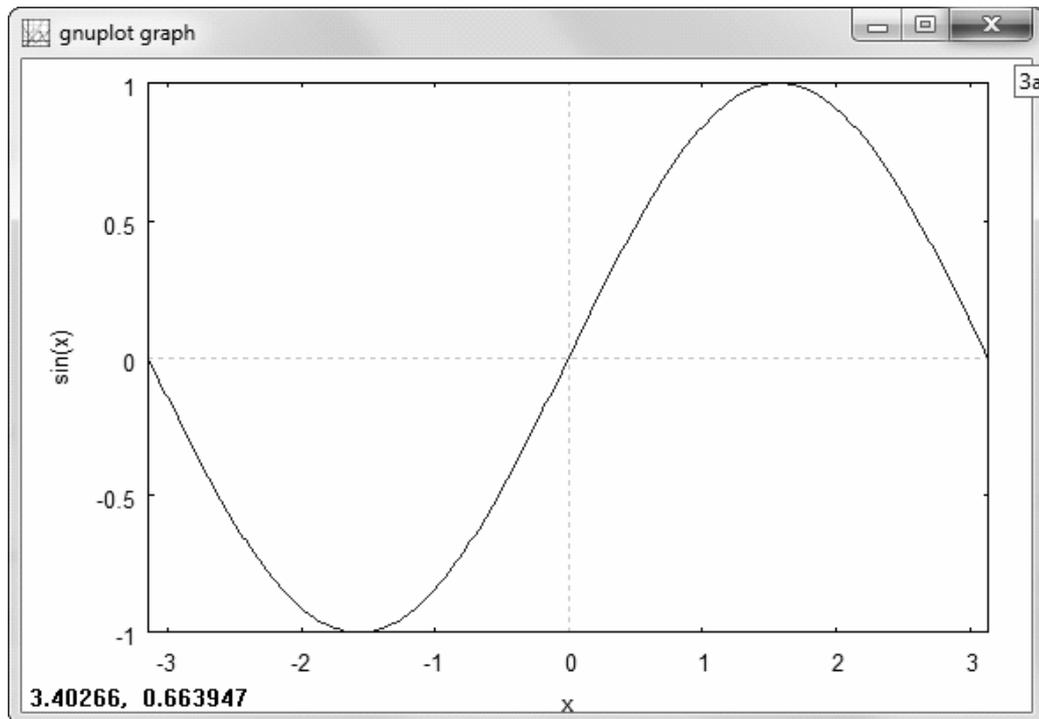


Рис. 5.10

```
(%i1) r: (exp(cos(t))-2*cos(4*t)-sin(t/12)^5)$
(%i2) plot2d([parametric, r*sin(t), r*cos(t), [t, -8*%pi, 8*%pi],
[nticks, 2000]])$
```

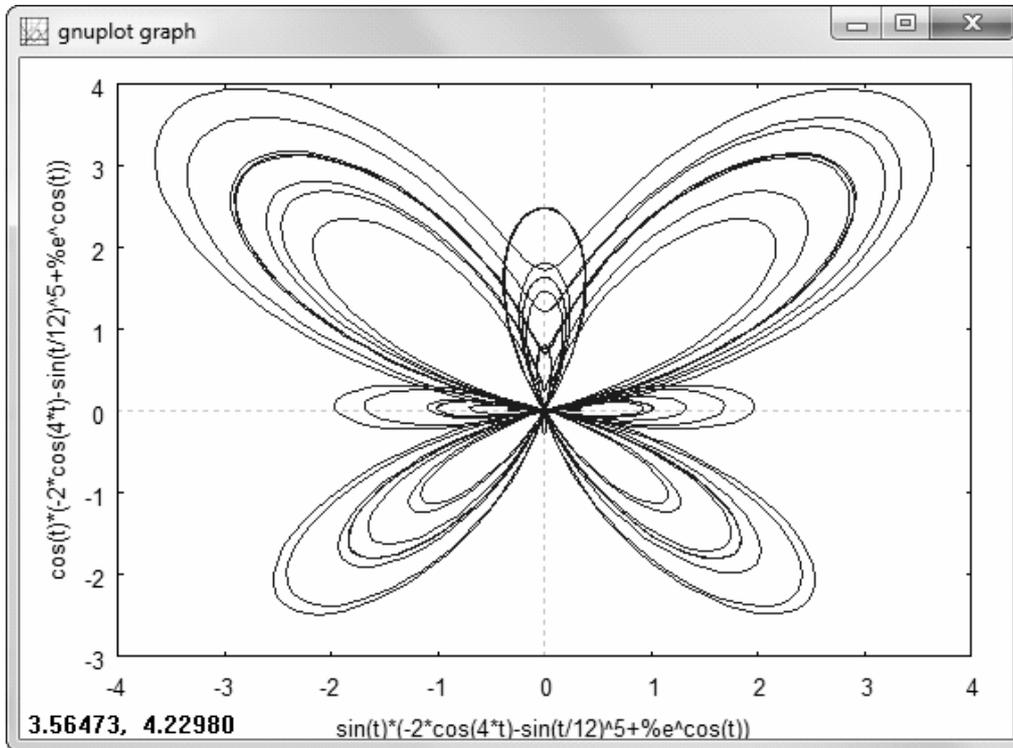


Рис. 5.11

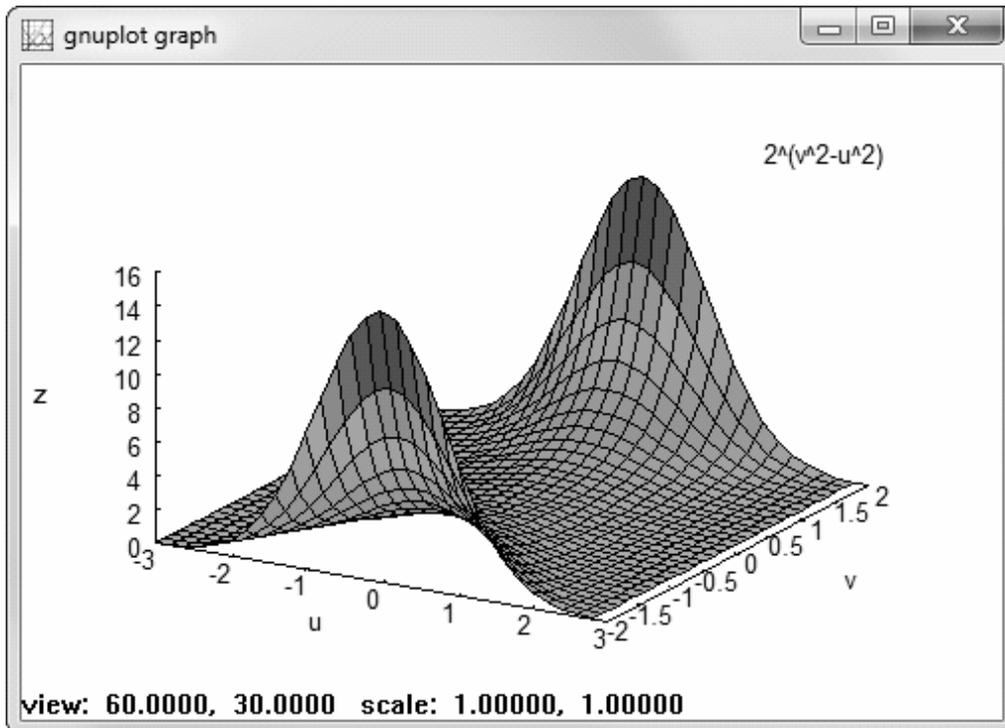


Рис. 5.12

## 6. СИСТЕМА МОДЕЛИРОВАНИЯ ASIMEC

### 6.1. Назначение

**ASIMEC (Advanced SIMulator of Electronic's Circuits)** — программа, для создания графических образов схем, их моделирования и визуализации результатов. Моделируемая схема может содержать такие компоненты как: сопротивления, индуктивности, емкости, диоды, биполярные транзисторы и полевые транзисторы с р-n-переходами, управляемые ключи и т.д., которые пользователь соединяет друг с другом посредством проводов. Схемы могут иметь «особенности» в виде индуктивных узлов и емкостных контуров. Модели полупроводниковых приборов — диодов, транзисторов, такие же как в SPICE-программах. Это значит, что вы можете использовать SPICE модели, поддерживаемых ASIMEC-ом типов, в своих схемах.

**ASIMEC** позволяет проводить анализ во временной области (решается задача Коши) и строить амплитудную и фазовую частотные характеристики. При решении задачи Коши используются либо неявная численная схема Эйлера либо схема трапеций либо схема Гира 2-го порядка.

Хотя **ASIMEC** не имеет встроенных ограничений на число элементов схем, следует помнить, что скорость моделирования, точность расчетов и возможность интерпретации результатов ухудшаются с ростом сложности схем.

**ASIMEC** предназначен для использования в научных исследованиях и для учебных целей. Он разрабатывался как альтернатива дорогим фирменным пакетам с акцентом на повышение качества моделирования — точность и скорость — схем малой и средней размерности. **ASIMEC** идеально подходит для обучения основам электротехники, электроники, основам цифровой схемотехники, моделированию систем автоматического регулирования.

**ASIMEC**, в отличие от SPICE программ, использует расширенный базис переменных состояний, что делает его намного более устойчивым в случае высоких требований к точности расчетов.

После инсталляции программы, рекомендуем запустить **ASIMEC**, визуально исследовать интерфейс программы, загрузить несколько примеров из каталога DEMO, запустить моделирование, изменять параметры компонентов, снова провести моделирование. После чего, если вопросы еще останутся, вернуться к изучению настоящего руководства.

## 6.2. Основное окно ASIMEC

После запуска **ASIMEC** открывается основное окно (см. рис. 6.1).

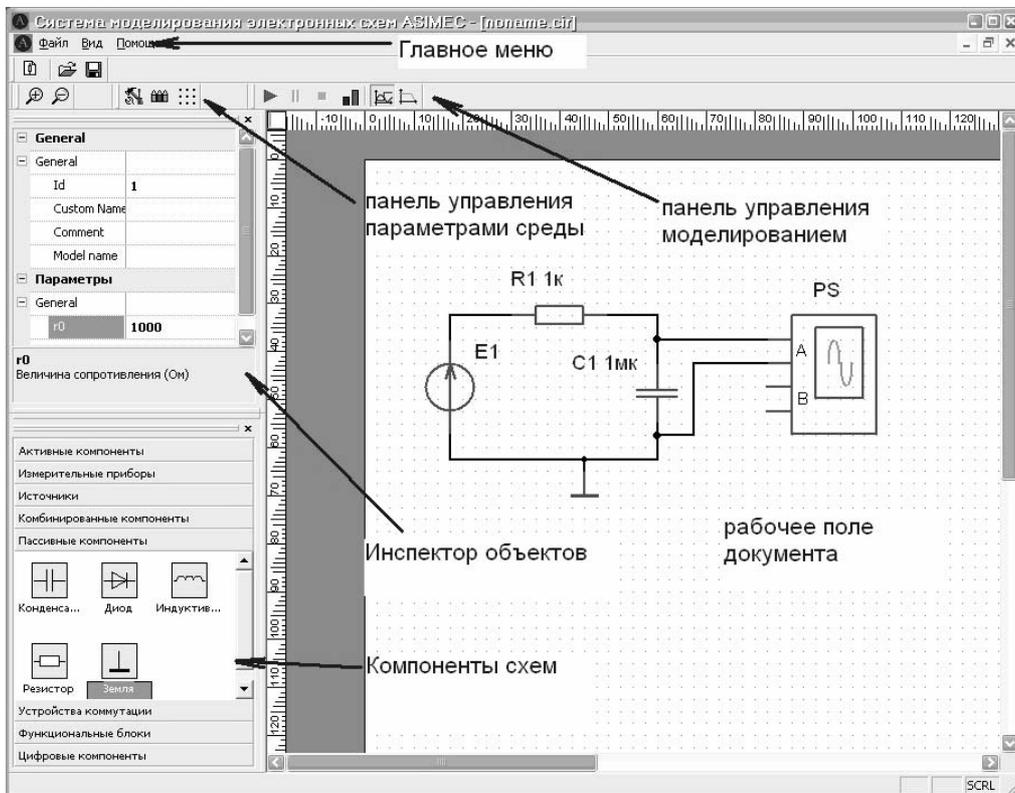


Рис. 6.1 — Основное окно

В верхней части окна **ASIMEC** расположена полоса главного меню. Раздел **Файл** позволяет управлять открытием/закрытием/сохранением редактируемой схемы. Раздел **Вид** позволяет показать/скрыть панели и Инспектор объектов. Раздел **Помощь** содержит команды для открывания настоящего руководства и показа информации о продукте.

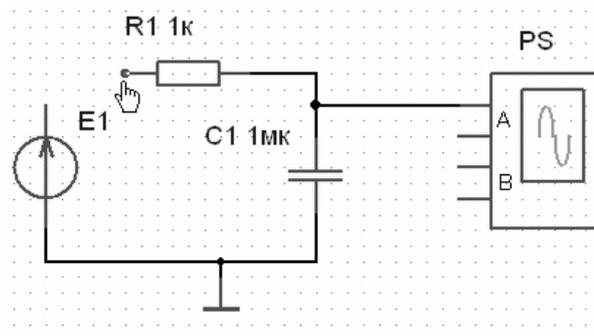
Основное («**рабочее поле документа**») предназначено для размещения компонентов и их соединения. Краткие подсказки по элементам управления можно получить во всплывающих окнах, если на некоторое время поместить на них курсор мыши.

### 6.2.1. Создание схемы

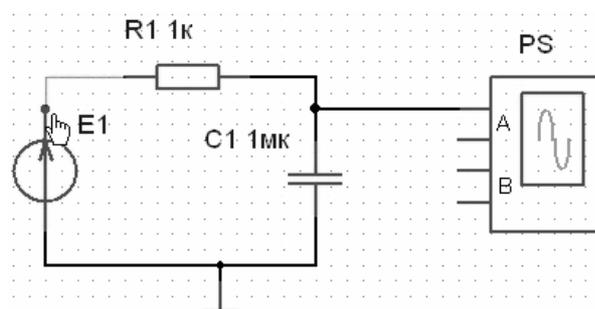
При создании схемы сначала на **рабочем поле** размещаются компоненты (например, резисторы, транзисторы), затем «проводами» соединяются их выводы, далее размещаются измерительные приборы и производится настройка параметров моделирования. Для того, чтобы поместить элемент в **рабочее поле**, необходимо переключить раздел на требуемую группу, например, «**Пассивные компоненты**». После открытия раздела находящиеся в нем элементы можно размещать, просто перетаскивая их мышкой на **рабочее поле**.

После того, как элементы размещены в **рабочем поле**, их можно объединить в группу (выделить). Для этого достаточно щелкнуть левой кнопкой мыши в **рабочей области** и, не отпуская, потянуть вправо вниз. При этом появится прерывистый прямоугольник, и после отпускания кнопки мыши элементы схемы, оказавшиеся внутри его, окажутся выделены. Повернуть, отразить, удалить, скопировать, вырезать, вставить схемный компонент можно выбором соответствующего пункта контекстного меню или нажатием горячих клавиш.

Для того чтобы соединить два вывода различных элементов, необходимо подвести курсор мыши к выводу одного из них до появления точки.

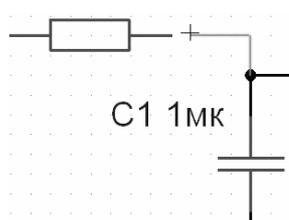


Далее щелкните мышью и тяните проводник до вывода второго элемента.



Когда на конце вывода, к которому нужно подсоединить проводник, появится точка, еще раз щелкните левой кнопкой мыши для создания связи.

Для удаления соединения между выводами, подведите курсор мыши к концу вывода элемента, в том месте, где он соединяется с проводником, и после появления синей точки (курсор при этом примет форму креста), нажмите левую клавишу мыши.

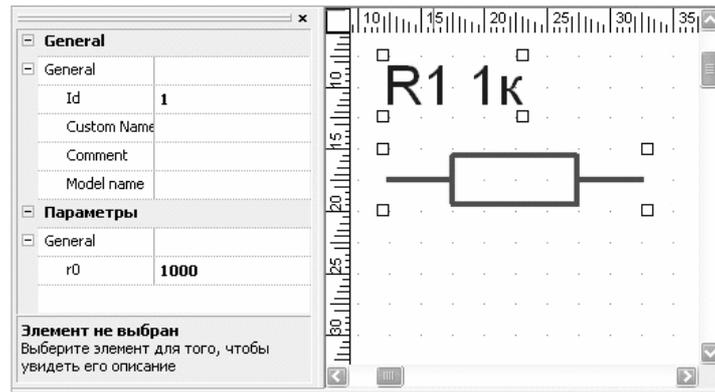


Далее, оттащите указатель соединения на пустое место в сторону от вывода элемента, и нажмите правую кнопку мыши. Если при протягивании соединения подвести курсор мыши к проводнику, и нажать левую кнопку мыши, автоматически создастся точка и к ней подсоединится связь.

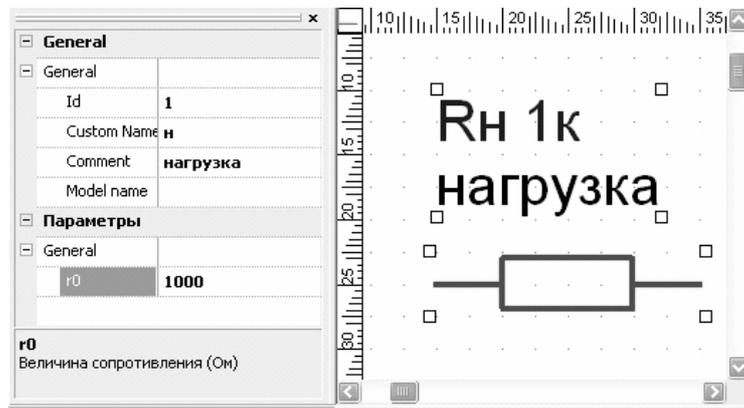
После создания провода, элементы, точки соединения, группы элементов можно удалять или перемещать по полю документа

### 6.2.2. Параметры электронных компонентов

Параметры компонента размещенного на рабочем поле можно изменять в настоящей версии **ASIMEC** только с помощью **инспектора объектов**. Для того чтобы свойства компонента отображались инспектором достаточно выделить его мышкой.



Раздел **General (Общее)** содержит Id, который позволяет управлять номером элемента R1 – Id=1, R2 – Id=2... Если в поле **Custom Name** находится какой либо текст, то он будет выведен на месте Id. См. ниже.

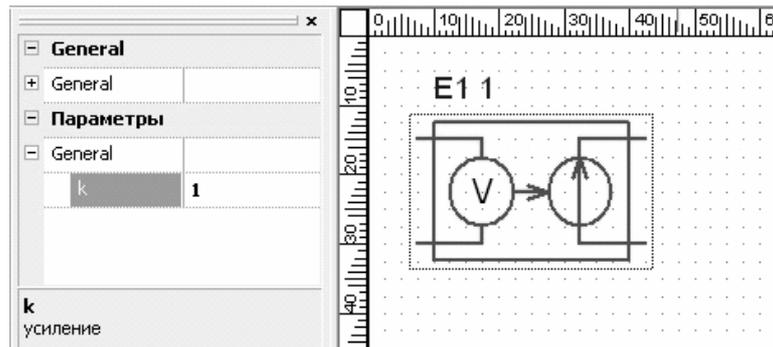


Поле **Comment** позволяет добавить текстовый комментарий. Раздел **Параметры** позволяет управлять параметрами компонентов. У резистора доступен один параметр — сопротивление. В нижней части инспектора находится краткая подсказка, которая активируется при редактировании поля.

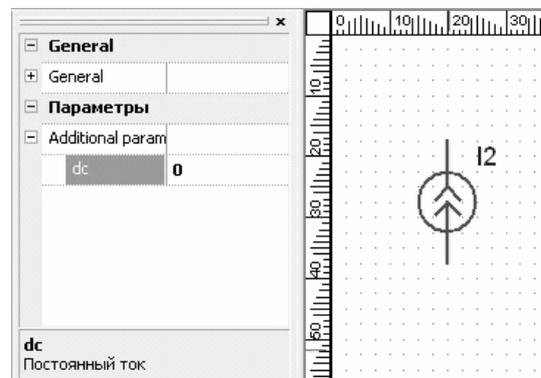
## 6.3. Элементы схем

### 6.3.1. Источники

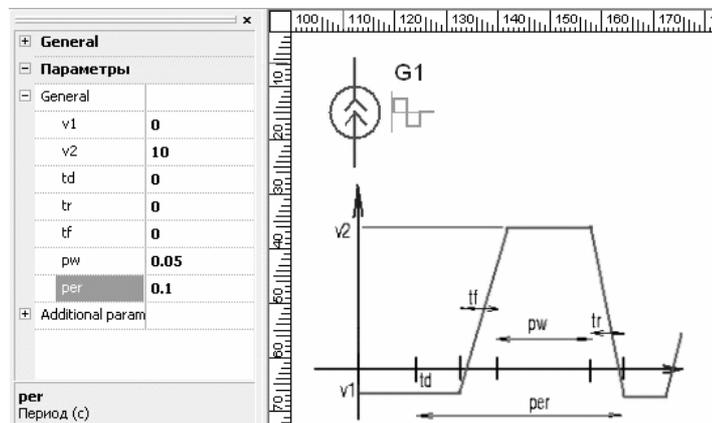
Линейно-зависимые источники, пример источника напряжения управляемого напряжением приведен на рисунке, у них можно управлять коэффициентом преобразования измеряемой входной величины в выходную, в нашем случае входного напряжения в выходное. По умолчанию коэффициент преобразования равен 1.



Источники постоянного тока (напряжения) позволяют задавать величину постоянного тока (напряжения)  $dc$ . По умолчанию  $dc=0$  В.



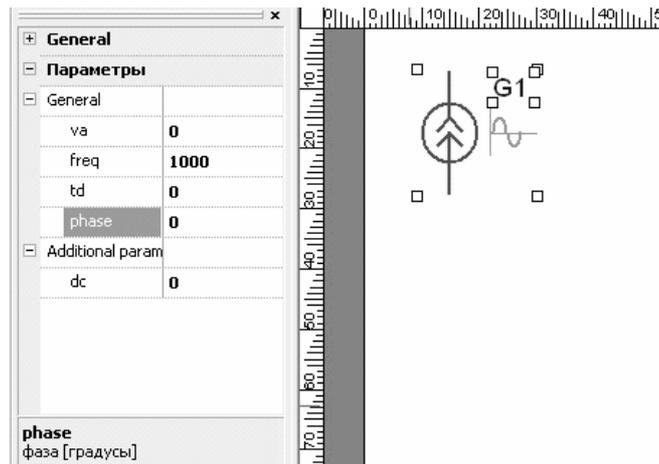
Источники импульсных токов (напряжений). Редактируемые параметры и их смысл можно понять из рисунка. Также при редактировании параметров в нижней части окна **Инспектора объектов** выводится подсказка. Если параметр **per**-период равен нулю, то импульс будет однократным — иначе, последовательность импульсов будет периодической.



Выходное напряжение источника синусоидального тока (напряжения) рассчитывается по формуле:

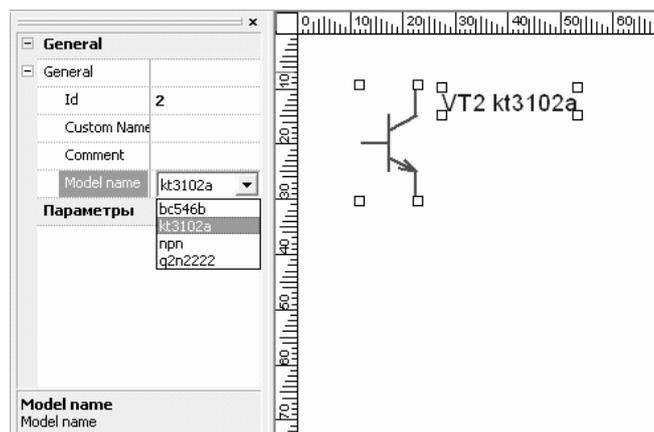
$$\text{value} = \text{va} * \sin(2\text{PI} * \text{freq} * (\text{t} - \text{td})),$$

Кроме того, можно управлять фазой источника, изменяя параметр **phase**. Изменение **phase** автоматически влечет пересчет параметра **td** — времени сдвига.



### 6.3.2. Транзисторы и диоды

Имеют математические модели такие же, как в SPICE3.



Они не имеют параметров, редактируемых **Инспектором объектов**. Параметрические описания (**.models** — в терминологии SPICE) полупроводниковых приборов находятся в текстовом файле **sample.lib**, который находится в каталоге LIB установочного пути. Вот как в библиотеке выглядит описание биполярного транзистора КТ3102А:

.model KT3102A npn (IS=5.25f XTI=3 EG=1.11 VAF=86 Bf=185.1 Ne=7 Ise=28n IKF=0.5 XTB=1.5 VAR=25 Br=2.7 NC=2 ISC=21p IKR=0.25 RB=52 RC=1.65 CJC=10p Vjc=0.65 Mjc=0.33 FC=0.5 CJE=11p Vje=0.69 Mje=0.33 Tr=57n TF=600p Itf=0.52 VTF=80 XTF=2)

С помощью текстового редактора возможно добавить в файл **sample.lib** модели других приборов или изменить их. Добавляемый тип должен поддерживаться **ASIMEC**, в настоящей версии это pnp/rnp — биполярные транзисторы; D — диоды; PJF, NJF — полевые транзисторы с P и N-каналом, а также подсхемы: **.SUBCKT**, содержащие эти приборы. В наших ближайших планах увеличить число типов моделируемых полупроводниковых приборов.

### 6.3.3. Общие сведения о моделировании

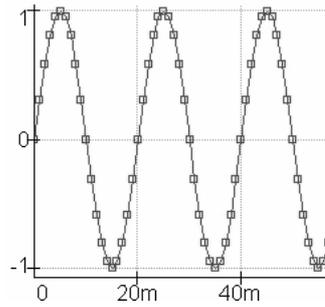
В **ASIMEC** реализовано два типа анализа **временной** и **малосигнальный частотный**.

Временной, аналогом которого в **SPICE** является **TRANSIENT** (от слова переходный) — анализ переходных процессов, или, говоря математическим языком — решение задачи Коши. Результатом решения являются массивы данных, один из которых хранит время, а другие — значения измеряемых переменных, соответствующие данному времени.

Таблица данных

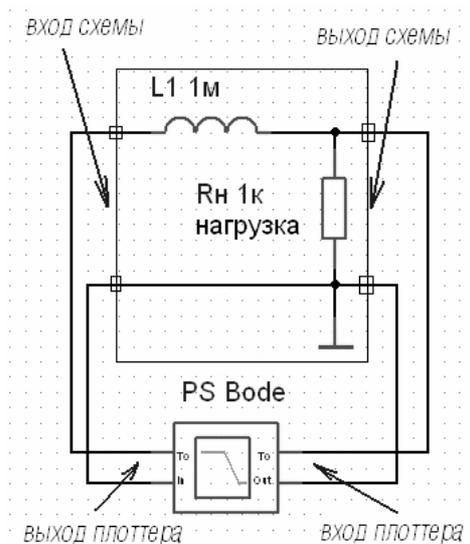
t, c	UA, B	UB, B
0	0	1
0.1	0.01	0.99
0.2	0.05	0.95
...		

После анализа по Таблице данных строятся графики, которые отображаются, например, на экране виртуального осциллографа, при этом точки, соответствующие строкам таблицы данных, соединяются **прямыми линиями**.



Точки, в которых реально производился расчет, на рисунке показаны квадратиками. На рисунке отображена синусоида с частотой 50 Гц и амплитудой 1. Важно понимать, что строго точные значения на этом графике находятся только в отмеченных точках.

Малосигнальный частотный анализ — для его проведения используется виртуальный прибор «Плоттер Боде». В ходе анализа с выхода Плоттера выдается синусоидальное напряжение, которое следует подать на вход схемы, а вход плоттера следует подключить к выходу схемы (см. рисунок).

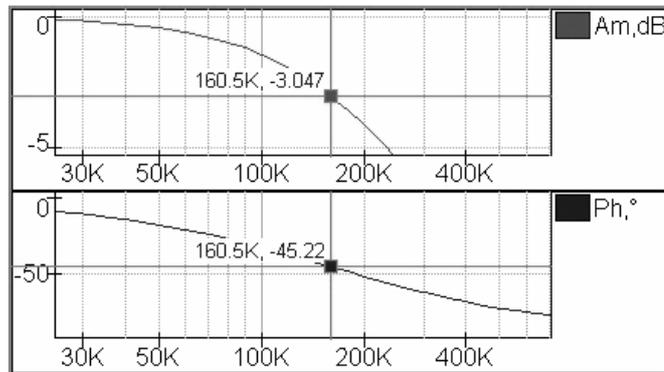


В результате малосигнального анализа строится таблица данных.

**Таблица данных**

f, кГц	Am(UA), Дб	Ph(UA), град
40	-0.26	-14
80	-1	-26
160	-3	-45
...		

Зависимость амплитуды и частоты отображаются на экране виртуального прибора. См. рисунок.

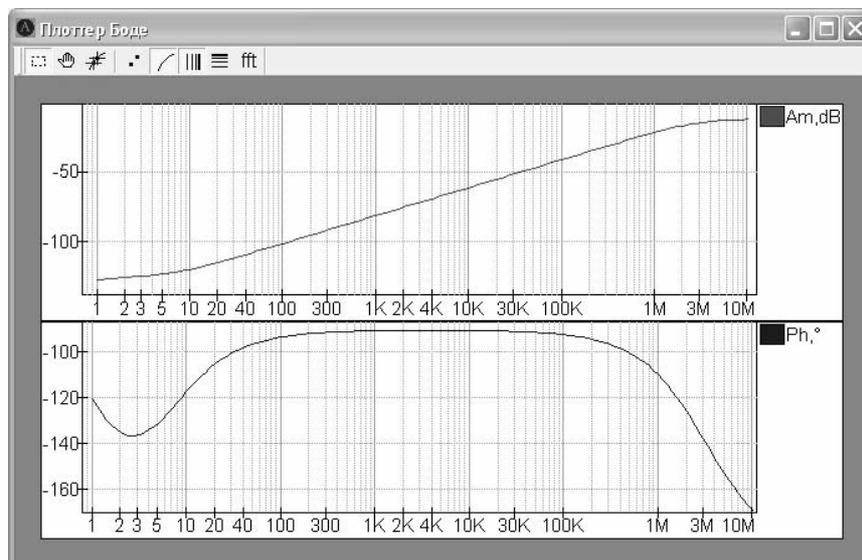


### 6.3.4. Управление режимами моделирования

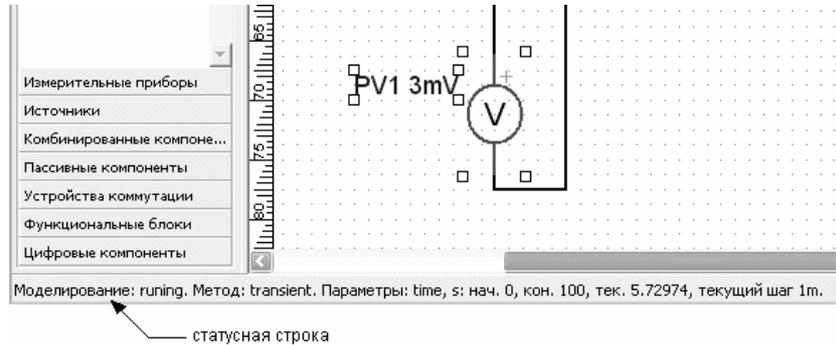
Чтобы выбрать требуемый тип анализа вдавните соответствующую кнопку на **панели режимов анализа**.



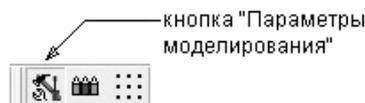
Чтобы запустить моделирование. Если на схеме присутствует символ «**Осциллограф**» или «**Плоттера Боде**», то в процессе моделирования откроется окно соответствующего прибора.



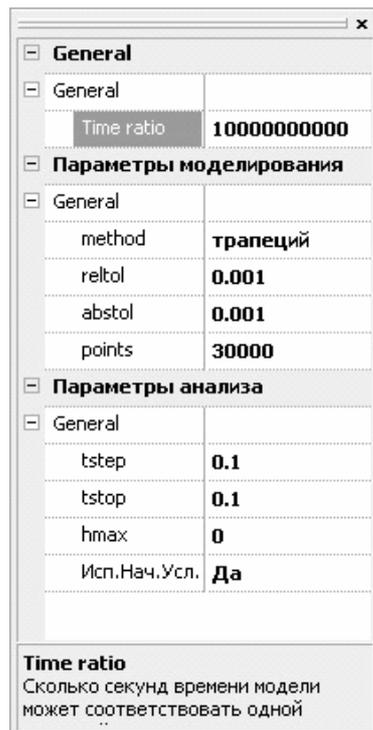
Если в схеме присутствует **вольтметр**, **амперметр** или **ваттметр**, то также произойдет запуск моделирования, при этом в **статусной строке** будут видны параметры моделирования (состояние, метод, начальное и конечное время, шаг и т.д.) и активизируется кнопка приостановить моделирование.



Чтобы задать **параметры моделирования** нажмите кнопку на панели «Свойства схемы».



Тогда **инспектор объектов** отобразит свойства моделирования.



Основные параметры анализа, которые необходимо знать при использовании **ASIMEC**.

**Time ratio.** При моделировании мы имеем дело с двумя типами времени: **реальным временем** (в котором мы живем) и **временем модели**. В некоторых случаях существует возможность моделировать очень быстро, при этом **время модели** будет существенно опережать **реальное время**. Иногда, однако, возникает необходимость замедлить скорость моделирования, синхронизировав ее, например, с **реальным временем** путем введения искусственных пауз в вычислительный процесс. Для этого и служит данный параметр. Если **Time ratio** = 1, то если время модели будет примерно соответствовать реальному. Если **Time ratio**  $\ll 1$ , то моделирование будет существенно замедляться. При **Time ratio**  $\gg 1$  — ускорятся. По умолчанию параметр равен  $10^{10}$  в этом случае моделирование происходит с максимально возможной скоростью.

Параметр **method** позволяет выбрать метод численного интегрирования. Метод интегрирования используется только при **Временном анализе**. Методы интегрирования имеют свои достоинства и недостатки. Так методы Эйлера и Гира дают лучшую сходимость результатов, а метод трапеций лучшую точность, когда нет проблем со сходимостью.

**reitol** — максимально допустимая относительная погрешность решения дифференциальных уравнений. Применяется для выходных переменных и «переменных состояния» в случае, когда величина больше отношения **abstol/reitol**.

**abstol** — максимально допустимая абсолютная погрешность решения дифференциальных уравнений. Применяется для выходных переменных и «переменных состояния» в случае малых величин — меньше отношения **abstol/reitol**.

В процессе моделирования **ASIMEC** оценивает точность, используя правило Рунге, а параметры **reitol** и **abstol** используются для коррекции (уменьшения или увеличения) шага интегрирования. Следует помнить, что правило Рунге дает **приближенную** оценку (говоря строго точную оценку при  $h \rightarrow 0$ ), в силу чего достижение указанной точности **не гарантируется**, заметим, что **SPICE** программы используют менее достоверные способы оцен-

ки погрешностей. Когда точность особенно важна, необходимо выбирать параметры **abstol** и **reltol** с разумным запасом.

Параметр **points** определяет размер буфер для хранения результатов. Данная версия **ASIMEC** хранит результаты моделирования только в буфере, находящемся в оперативной памяти компьютера, который может быть просмотрен с помощью доступных приборов, например, осциллографа.

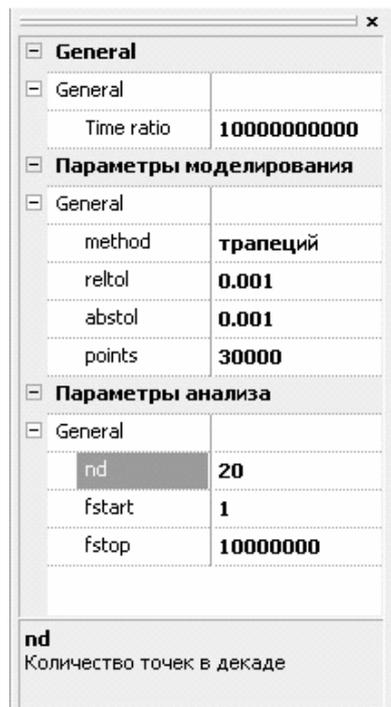
Раздел **параметры анализа** будет выглядеть в зависимости от того какой тип анализа выбран кнопками (). Вид раздела на рисунке, приведенном выше, соответствует **временному анализу**. Можно управлять следующими параметрами:

**tstep** — начальный шаг моделирования, является рекомендуемым значением, программа может его уменьшить;

**tstop** — время окончания моделирования;

**hmax** — ограничивает шаг моделирования сверху, что может быть полезным для повышения точности расчетов, поскольку, чем меньше шаг, тем выше точность. Однако малое значение этого параметра будет замедлять процесс моделирования. При  $h_{max}=0$  усечение шага отсутствует.

**Исп.Нач.Усл.** — флаг, который указывает на то что следует или использовать нулевые начальные условия — **Исп.Нач.Усл.=Да**, или производить предварительный расчет рабочей точки по постоянному току — **Исп.Нач.Усл.=Нет**.



Окно инспектора объектов в случае **частотного анализа** отличается лишь разделом «**Параметры анализа**». Можно изменить параметры **nd** — число точек на декаду; начальную и конечную частоту **fstart** и **fstop** соответственно.

### 6.3.5. Особенности проведения измерений

Доступны следующие измерительные приборы:

*В аналоговой части*

**Осциллограф** и **плоттер Боде**, представляющие информацию в графическом виде. А также **амперметр**, **вольтметр** и **ваттметр** — показывающие число, равное результату измерений.

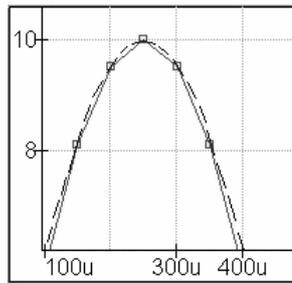
При анализе цифровых схем доступны логический пробник и семисегментный индикатор с входом разрешения E. Если на вывод E подать «1», то значения будут отображаться на индикаторе. При логическом нуле на выводе E индикатор будет показывать последнее значение, запомненное при E=«1».

**Осциллограф** (запоминающий) и **плоттер Боде** показывают графики, которые строятся прямыми линиями по смежным рассчитанным точкам. Если плотность точек недостаточна ее можно увеличить, изменяя параметры **hmax** и **nd** в соответствующих типах анализа.

**Амперметр**, **вольтметр** и **ваттметр** имеют 3 режима работы: измерение мгновенного среднего или среднеквадратичного значения измеряемой величины.

Мгновенное значение это значение, соответствующее текущему времени моделирования, если процесс моделирования еще не завершен, или значение соответствующее времени окончания моделирования, если процесс завершен.

Среднее и среднеквадратичное значение имеет смысл, когда указан период измерения, обычно такие измерения имеют смысл, если период измеряемых величин известен. Например, когда присутствует один или несколько источников синусоидального напряжения одинаковой частоты. В электротехнике, чаще всего это 50 Гц, что соответствует периоду измерений  $1/50 \text{ Гц} = 20 \text{ мС}$ .



Измерение среднего и среднеквадратичного значения происходит путем обработки массива выходных данных с линейной интерполяцией между точками! При этом точность измерения будет зависеть не только от точности расчетов в точках, но и от их числа! См. рисунок, где точное гладкое решение показано темной пунктирной линией! Это приводит к появлению погрешностей. Так при 10-ти точках на период (ограничение на шаг по умолчанию при наличии периодических источников) погрешность нахождения среднеквадратичного решения синусоиды составляет 2 %.

## 6.4. Аналоговое, цифровое и смешанное моделирование

### 6.4.1. Аналоговое моделирование

При моделировании схем, содержащих аналоговые элементы, моделирующее ядро программы сначала формирует систему нелинейных алгебро-дифференциальных уравнений, а затем интегрирует ее численными методами.

Любой цифровой элемент может быть представлен множеством аналоговых (диодов, резисторов, транзисторов и т.п.) и смоделирован таким же образом. Однако, такой способ, является накладным бессмысленным при моделировании большого числа цифровых элементов, если известно, что цифровые элементы достаточно надежны и работают строго по законам логики при соблюдении технических условий — питающего напряжения, нагрузочной способности и т.п.

По этому с точки зрения сокращения временных затрат на порядки (!) и повышения численной устойчивости моделей полезно моделировать цифровые элементы на логическом уровне.

### 6.4.2. Цифровое моделирование

В ходе цифрового моделирования выходы цифровых элементов посылают цифровым узлам события такие как «начало фронта», «окончание фронта», «переход в неопределенное третье состояние» и т.п. Узел, получив событие, в свою очередь, рассылает его входам цифровых элементов, которые подсоединены к данному узлу.

### 6.4.3. Смешанное моделирование

Таким образом, методы аналогового и цифрового моделирования радикально различаются. Перед моделированием ядро создает два графа — цифровой и аналоговый. Для связи аналогового и цифрового графов служат конверторы A/D и D/A, соответственно аналогово-цифровой и цифро-аналоговый.

## 6.5. Примеры некорректных схем

Примеры правильных и не правильных схем (которые не будут моделироваться) приведены ниже.

На рисунке 6.2 приведён пример топологически некорректной схемы.

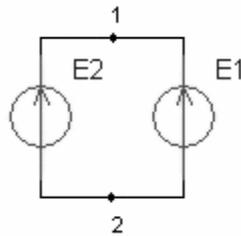


Рис. 6.2 — Топологически некорректная схема

Напряжение между узлами 1 и 2 должно быть равно значению источников напряжения, но поскольку их два, то возникает неразрешимое противоречие. Если попытаться создать физический аналог такой схемы, то возникнет короткое замыкание, разрушительные эффекты которого будут тем сильнее, чем чище эксперимент. В схемах ASIMES недопустимы контура, образуемые:

- 1) только источниками напряжения (рис. 6.3, а);
  - 2) только источниками напряжения и измерителями тока (рис. 6.3, б);
  - 3) только источниками напряжения и емкостями (рис. 6.3, в);
- Также недопустимы узлы, к которым подсоединены:
- 1) только измерители напряжения (рис. 6.3, г);
  - 2) только источники тока (рис. 6.3, д);
  - 3) только источники тока и измерители напряжения (рис. 6.3, е);
  - 4) только источники тока и индуктивности (рис. 6.3, ж);
  - 5) только источники тока, индуктивности и измерители напряжения (рис. 6.3, з).

Примеры других возможных некорректных фрагментов схем изображены на рисунке 6.3.

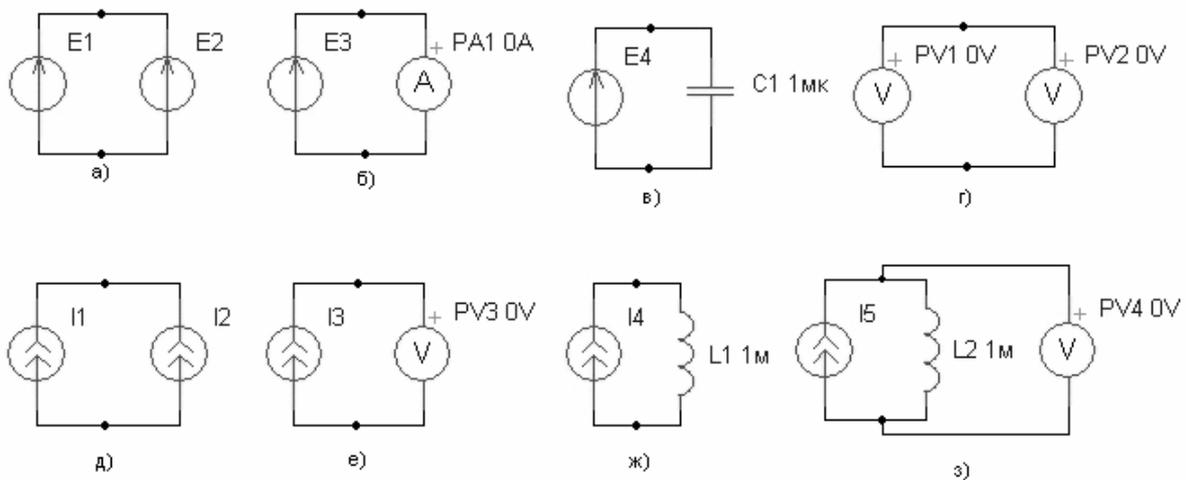
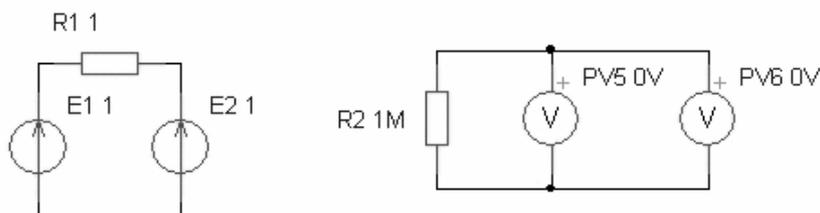


Рис. 6.3 — Примеры некорректных фрагментов схем

Избавится от особенностей можно, например, путем введения сопротивлений (см. нижний рисунок). В случае с особенностями типа контуров — последовательно в контур (значение сопротивления должно быть достаточно малым), а в случае особенностей типа узлов — присоединив сопротивление к таким узлам (такое сопротивление должно быть достаточно большим).



## 7. СИСТЕМА ДИСТАНЦИОННОГО ОБУЧЕНИЯ MOODLE

**Moodle** — это *программный продукт*, позволяющий создавать курсы и web-сайты, базирующиеся в сети Интернет. Это постоянно развивающийся проект, основанный на теории социального конструктивизма.

Moodle распространяется бесплатно в качестве программного обеспечения с открытым кодом (*Open Source*) под лицензией *GNU Public License (rus)*. Это значит, что Moodle охраняется авторскими правами, но и Вам доступны некоторые права. Вы можете копировать, использовать и изменять программный код по своему усмотрению в том случае если вы согласны: предоставлять код другим, не изменять и не удалять изначальные лицензии и авторские права и использовать такую же лицензию на всю производную работу.

Moodle (Мудл) может быть *установлен* на любом компьютере, поддерживающем PHP, а также базы данных типа SQL (например, *MySQL*).

Он может быть запущен на операционных системах Windows или Mac и многих разновидностях linux (например, Red Hat или Debian GNU). Есть много людей, сотрудничающих с Moodle, которые обладают достаточными знаниями, чтобы помочь Вам, и даже поспособствовать в создании Вашего собственного сайта Moodle.

Слово Moodle — это аббревиатура от понятия Модулярная Объектно-Ориентированная Динамическая Обучающая Среда, которая наиболее полезна для программистов и теоретиков. Это также может быть и глаголом, который означает процесс медленного продирирования сквозь дебри, изучения чего-либо по мере его появления, исправление своих ошибок, которое впоследствии ведет к развитию интуиции, сообразительности и творческих способностей. По существу, обе эти трактовки подходят, учитывая то, как разрабатывался этот проект и то, что учителя и студенты смотрят на вопрос он-лайн образования по-разному. Все, кто используют проект Moodle (Мудл) — Moodlers (Мудлеры).

## 7.1. Основные принципы и функции системы дистанционного обучения Moodle

При проектировании Moodle особое внимание уделяется следующим принципам:

- Продвижение педагогики социального конструкционизма (сотрудничество, активное обучение, критическая рефлексия и пр.).
- Поддержка различных подходов к обучению: дистанционное, смешанное, очное.
- Простой, интуитивно понятный, эффективный, кросс-платформенный интерфейс в окне браузера.
- Простая установка на большинство платформ, поддерживающих PHP.
- Совместимость с большинством широко используемых баз данных.
- Список курсов содержит описания и доступен любому пользователю.
- Курсы структурируются по категориям. По ним можно производить поиск. Один сайт может содержать тысячи курсов.
- Существенное внимание уделяется вопросам безопасности. Все формы и вводимые данные проверяются, cookies шифруются и т.п.
- Для большинства текстовых областей (ресурсы, сообщения форумов и т.п.) используется встроенный WYSIWYG HTML редактор.

## 7.2. Управление системой дистанционного обучения

- Сайт управляется *администратором*.
- Основные настройки сайта определяются в процессе установки, но могут быть легко изменены и в будущем.
- Для изменения настроек сайта используется удобный блок *Администрирование*.
- Подключаемые модули «тем» дают администратору широкие возможности по изменению внешнего вида сайта.
- С помощью *дополнительных модулей* можно легко расширить возможности сайта.

- Подключаемые языковые пакеты позволяют сделать перевод на любой язык. *Редактирование перевода* осуществляется с помощью встроенного веб-редактора. На данный момент перевод осуществлен более чем на *70 языков мира*.
- Весь код написан на языке PHP под открытой лицензией GPL, что позволяет легко адаптировать систему под свои нужды.

### 7.3. Управление пользователями

- Основная цель: свести действия администратора по управлению пользователями к минимуму. Сохранив при этом высокий уровень безопасности.
- Поддерживается большое количество механизмов аутентификации за счет использования плагиновой архитектуры. Тем самым обеспечивается простота интеграции с внешними системами.
- Стандартная e-mail аутентификация: пользователи сами создают себе учетные записи. E-mail адрес проверяется путем отсылки на него письма с подтверждением регистрации.
- LDAP аутентификация производится через LDAP сервер. Обладает гибкими настройками используемых полей.
- IMAP, POP3, NNTP: аутентификация производится через почтовый или новостной сервер. Поддерживается SSL, сертификаты, TLS.
- Внешняя БД: любая база данных, содержащая, как минимум, два поля может выступать в качестве внешнего источника аутентификации.
- Пользователи имеют широкие возможности по заполнению своего профиля. При необходимости e-mail адреса могут быть защищены от показа.
- Администратор имеет возможность гибко управлять полями профиля: добавлять, защищать от изменений и т.п.
- Каждый пользователь может указать свою временную зону. Эти настройки будут учитываться при отображении любых дат в рамках системы (например, даты отправки сообщений, сроки сдачи заданий и т.п.).

- Каждому пользователю может быть предоставлена возможность выбора языка, на котором ему будет отображаться интерфейс системы.

#### **7.4. Запись на курсы**

- Учителя могут устанавливать для своих курсов так называемые «*кодовые слова*» для предотвращения доступа к курсу посторонних участников. Учителя могут сообщить кодовое слово своим учащимся лично, по e-mail или каким-нибудь другим способом.

- Учителя могут записывать учащихся на курс вручную, если это необходимо.

- Учителя могут вручную отчислять учащихся с курса, если это необходимо. Кроме того, администратором может быть задан период (Управление > Сервер > Очистка), по окончании которого, если учащийся не заходил в курс, он будет автоматически отчислен.

- Плагиновая архитектура методов записи позволяет легко создавать и подключать произвольные процедуры записи на курс.

- Каждый пользователь имеет всего одну учетную запись в системе, но при этом может иметь разные уровни доступа к курсам.

- *Метакурсы* предоставляют удобное средство автоматической синхронизации подписок пользователей. Список участников метакурса наследуется из других курсов, связанных с данным метакурсом.

#### **7.5. Роли**

- Система ролей является мощным средством наделения пользователей различными правами на разных уровнях иерархии контекстов (Система->Категория курса->Курс->Элемент курса).

- Роли определяются администратором сайта. Они представляют собой набор разрешений для всех возможностей системы (например, Разрешить отвечать на сообщения в форуме, Запретить создавать темы форума и т.п.).

- Существует набор предустановленных ролей (права которых все равно могут быть отредактированы): администратор,

создатель курсов, учитель, учитель без права редактирования, учащийся, гость.

- Создатель курсов может создавать курсы, редактировать содержимое и обучать.
- Учитель может редактировать содержимое курсов и обучать в них.
- Учитель без права редактирования может только обучать в курсе.
- Учащийся может обучаться в курсе.
- Гость имеет минимальные права просмотра некоторых элементов курса.

## 7.6. Управление курсом

### *Обзор*

- Учитель имеет полный контроль над курсом: изменение настроек, правка содержания, обучение.
- В стандартную поставку включены следующие форматы курсов: календарь, структура, сообщество (форум), SCORM. Благодаря плагиновой архитектуре любой формат курса может быть создан и подключен к системе.
- Для каждого курса может быть выбрана своя тема отображения.
- Каждый курс может включать большой набор различных элементов: ресурсы, форумы, тесты, задания, глоссарии, опросы, анкеты, чаты, лекции, семинары, wiki, базы данных, SCORM объекты. Благодаря плагиновой архитектуре любой элемент курса может быть разработан и подключен к системе.
- В рамках курса могут быть загружены файлы, доступ к которым смогут получить только участники курса.
- Большой набор блоков (календарь, последние новости, наступающий события и т.п.) расширяет функциональность и удобство использования системы. Благодаря плагиновой архитектуре любой блок может быть разработан и подключен к системе.
- Учащиеся и учителя могут быть записаны в несколько групп и groupings (потоков). Группы позволяют разделить активность пользователей в рамках элементов курса. Groupings (пото-

ки) позволяют отображать некоторые элементы курса только для определенных объединений участников.

- Блок и страница последних действий позволяют в удобном виде вывести для вас все изменения в курсе с вашего последнего входа в систему.
- Содержимое большинства текстовых областей (ресурсы, сообщения форумов и т.п.) легко редактируется с помощью встроенного WYSIWYG HTML редактора.
- Большинство элементов курса могут быть оцениваемыми.
- Все оценки собираются в единый журнал, который содержит удобные механизмы для подведения итогов, создания и использования различных отчетов, импорта и экспорта оценок.
- Создание собственных шкал дает широкие возможности для критериального оценивания результатов обучения.
- Мощная система учета и отслеживания активности участников курса позволяет в любой момент увидеть полную картину как об участии в курсе целом, так и детальную информацию по каждому элементу курса.
- Интеграция с почтой позволяет отправлять по e-mail копии сообщений в форумах, отзывы и комментарии учителей и другую учебную информацию.
- Благодаря функции резервного копирования любой курс может быть сохранен как одиночный zip Файл и затем восстановлен на любом сервере с Moodle.

## 7.7. Ресурс

- Поддерживается отображение любого электронного содержания: Word, Powerpoint, flash, видео, музыка и т.п. Все это может храниться как локально, так и на любом внешнем сайте.
- Файлы могут загружаться на сервер и размещаться в курсе. Затем файлы можно архивировать, разархивировать, переименовывать, перемещать и т.п.
- Для структурирования файлов можно создавать папки, которые также можно размещать потом на странице курса.
- С помощью WYSIWYG HTML редактора можно создавать веб-страницы и размещать их в курсе.

- На странице курса можно размещать ссылки на внешние веб приложения и передавать данные в них.
- MP3 аудио-файлы отображаются и проигрываются с помощью удобного flash-плеера.

### **7.7.1. Лекция**

- Лекция представляет собой последовательность страниц, которые могут отображаться линейно, как презентации, нелинейно с ветвлениями или условными переходами между страницами, либо комбинировано с использованием обоих вариантов.
- В лекцию могут быть добавлены вопросы, правильность ответа на которые может быть связана с условными переходами между страницами.
- Страницы с вопросами могут быть следующих типов: множественный выбор, верно-неверно, числовой ответ, короткий ответ, эссе.
- Возможны различные варианты начисления баллов и выставления оценки
- Страницы могут быть импортированы из презентации PowerPoint. Страницы с вопросами могут быть импортированы из различных форматов.
- Можно настраивать количество попыток учащихся, временные ограничения, минимальные баллы.
- Содержимое страниц хранится в формате HTML и может быть отредактировано встроенным WYSIWYG редактором.
- Учащемуся могут отображаться индикатор выполнения, полученные на данный момент баллы.
- Возможны различные ограничения на доступ к лекции: пароль, временные ограничения и т.п.

### **7.7.2. Глоссарий**

- Это один из тех модулей, которые очень хорошо иллюстрируют, как Moodle может дополнить и расширить возможности традиционного очного обучения.
- Когда учащийся размещает результаты своей деятельности на общее обозрение (например, как в глоссарии), он прикла-

дывает больше усилий, получает лучший результат, а соответственно и больший опыт.

- Участники могут создавать и поддерживать список определений, похожий на словарь, энциклопедию и т.п.
- Записи учащихся перед опубликованием могут быть предварительно просмотрены учителем.
- По записям можно производить поиск и перемещение, используя алфавит, категории, даты и имена авторов.
- Если какой-нибудь из терминов глоссария встретится в тексте в любом месте курса, он может быть автоматически преобразован в ссылку на этот термин.
- Записи глоссария можно структурировать по категориям.
- Участники могут комментировать записи глоссария.
- Записи можно оценивать по любой определенной учителем шкале.
- Записи глоссария можно легко экспортировать и импортировать в формате xml.
- По записям глоссария можно производить поиск.
- Существует несколько форматов отображения записей.

### **7.7.3. База данных**

- Расширяет возможности глоссария — позволяет определять произвольную структуру записей.
- Поддерживаются следующие типы полей: дата, картинка, меню, меню (выбор нескольких), переключатели, ссылка, текстовая область, текстовое поле, файл, флажки, число, широта/долгота. Благодаря плагиному механизму можно разрабатывать новые типы полей.
- Записи могут просматриваться в двух режимах: «Просмотр списка» и «Просмотр по одной записи».
- Учитель может определять разметку режимов просмотра.
- Набор полей и настройки режимов просмотра можно сохранить как предустановку. Предустановки можно распространять как в рамках одного сайта, так и между сайтами с Moodle.
- По записям возможен поиск.
- Записи можно экспортировать и импортировать.
- Участники могут комментировать записи в базе данных.

- Записи можно оценивать по любой определенной учителем шкале.
- Учитель может определять различные ограничения по работе с БД: временные промежутки просмотра, добавления записей, ограничения на минимальное и максимальное количество записей.

#### 7.7.4. Тест

- Учитель составляет базу тестовых заданий, которые потом могут использоваться в различных тестах.
- Тестовые задания можно структурировать по категориям для более удобной работы.
- Категории можно делать доступными на разных уровнях. Иерархия уровней выглядит следующим образом: Элемент курса ->Курс->Категория курса->Система.
- Большинство тестовых заданий оцениваются автоматически. При изменении задания, тест может быть переоценен.
- Учитель может определять различные ограничения по работе с тестом: начало и конец тестирования, задержки по времени между попытками, количество попыток, пароль на доступ, доступ только с определенных сетевых адресов.
- Учитель может настраивать количество попыток для сдачи теста, указывать комментарии как к отдельным ответам, так и ко всему тесту в зависимости от различных условий.
- При выполнении последующих попыток могут учитываться предыдущие варианты ответов. В рамках одной попытки учащиеся могут выполнять тест в несколько подходов.
- Как тестовые задания, так и варианты ответов могут автоматически перемешиваться при каждой новой попытке.
- Как вопросы, так и варианты ответов могут содержать HTML, картинки и пр.
- Вопросы можно экспортировать и импортировать в различных форматах.
- Учитель может получить подробную информацию об ответах учеников и затрачиваемом ими времени.
- Существует возможность статистического анализа выполнения тестовых заданий.

Поддерживаются вопросы следующих типов:

- В закрытой форме (множественный выбор): предоставляет возможность выбора одного и нескольких вариантов. К каждому варианту учителем может быть указан комментарий, который отобразится ученику после ответа.

- Короткий ответ.

- Числовой: принимается ответ в виде числа с определенной погрешностью, также можно предоставить возможность ответа в различных единицах измерения.

- Вычисляемый: текст вопроса формулируется как шаблон, для него формируется определенный набор значений, которые будут автоматически подставляться. На основе заданной формулы и этих значений формируется правильный ответ. Таким образом, каждому ученику будет отображен вопрос со своими числовыми значениями и, соответственно, своим правильным вариантом ответа.

- На соответствие.

- Верно/Неверно.

- Эссе: предполагает текстовый ответ ученика в свободной форме. Оценивается учителем.

- Вложенные ответы: текст вопроса содержит поля, в которых учащийся должен ввести или выбрать ответ.

- Случайные вопросы: на место случайных вопросов в тесте ученику при каждой новой попытке будут случайным образом подставляться вопросы из выбранной категории.

- Благодаря плагиному механизму можно создавать и подключать свои типы вопросов.

#### **7.7.5. Задание**

- Задание предполагает ответ учащегося в виде текста, файла, нескольких файлов или вне сайта.

- Для задания указывается максимальная оценка или шкала оценивания.

- Выполнение задания может быть ограничено по времени.

- Учитель может разрешить ученикам отвечать после окончания срока выполнения. В этом случае учителю будет отображаться информация о просроченных ответах.

- Учитель может выставлять оценки и указывать комментарии как индивидуально, так и на одной странице для нескольких учеников.
- Комментарий учителя будет отображаться учащемуся под оценкой и может быть отправлен по e-mail.
- Учитель может разрешить повторную отправку ответа после того, как оценка была выставлена. Таким образом можно организовать итеративный режим работы с заданием.

#### **7.7.6. Форум**

- Доступны различные типы форумов: новостной, стандартный форум для общих обсуждений, простое обсуждение, каждый открывает одну тему, Вопрос-Ответ.
- Все сообщений содержат картинку участника.
- Дискуссии можно просматривать в разных режимах отображения: древовидно, плоско.
- Участникам может быть разрешено индивидуально управлять подпиской на каждый форум или учитель может принудительно подписать всех.
- Учитель может запретить отправку сообщений в форум (например, используя форум только как рассылку новостей).
- Дискуссии можно перемещать между различными форумами.
- Для форума может использоваться оценивание по произвольной шкале, при этом оно может быть ограниченно определенным временным промежутком.

#### **7.7.7. Чат**

- Позволяет организовать синхронное текстовое взаимодействие между участниками.
- Поддерживает смайлики, HTML, картинки и т.п.
- При включенных математических фильтрах позволяет использовать формулы.
- В окне чата отображаются картинки из профилей пользователей.
- Все чат-сессии могут быть сохранены для последующего просмотра.

### **7.7.8. Опрос**

- Может быть использован для голосования или сбора мнений по какому-либо вопросу.
- Учитель видит результаты в виде интуитивно понятной таблицы.
- Учитель может гибко настроить просмотр результатов учащимися.

### **7.7.9. Анкета**

- Встроенные анкеты (COLLES, ATTLS) — это хорошо зарекомендовавший себя инструмент для анализа он-лайн классов.
- Всегда доступны подробные отчеты с большим количеством диаграмм. Данные могут быть загружены в виде документа Excel или CSV файла.
- Интерфейс анкет не допускает частичного заполнения.
- Учитель может оставить комментарий для учащегося о его результатах по сравнению со средними показателями других участников.

### **7.7.10. Wiki**

- Wiki — это веб-страница, которую может добавить и редактировать кто угодно.
- Старые версии страниц сохраняются и могут быть легко восстановлены.
- Wiki позволяет организовать совместную работу над документами и тем самым обучение в сотрудничестве.

### **7.7.11. Семинар**

- Позволяет проводить многопозиционное многокритериальное оценивание работ.
- Учитель может управлять процессом оценивания и оценивать сам процесс.
- Поддерживает широкий спектр шкал оценивания

- Учитель может предоставить образец работы для того, чтобы учащиеся могли попрактиковаться в оценивании.
- Очень гибкий с большим количеством опций.

## 7.8. Система дистанционного обучения ТУСУРа

Система дистанционного обучения ТУСУРа создана на базе использования системы Moodle, сайт которой находится по адресу <http://fdo.tusur.ru/online/>, на рис показана начальная страница этого сайта.

The screenshot displays the Moodle LMS interface for TUSUR. The page layout includes a navigation menu on the left, a central content area, and a sidebar on the right. The main content area is titled 'Категории курсов' and lists various courses under the heading 'ФДО ТУСУР'. The sidebar on the right contains a login form, a calendar for April 2012, and a list of users on the site.

**Основное меню**

- Новости сайта
- Опрос "Форматы представления электронных учебных пособий"

**Категории курсов**

**ФДО ТУСУР**

- Институциональная экономика
- Экономическое моделирование
- Правоохранительные органы
- Финансовые рынки
- Нанозлектроника
- ИКТ в ДО
- Муниципальное право РФ
- Введение в вычислительную технику
- Финансовые вычисления
- Теоретические основы реструктуризации
- Архитектура предприятия
- Демонстрация возможностей Moodle (для студентов)
- Компьютерные системы и сети
- Конституционное право зарубежных стран
- Лингвистическое программное обеспечение САПР
- Прокурорский надзор
- Право социального обеспечения
- Семейное право
- Уголовно-процессуальное право
- Юридическая психология
- Документационное обеспечение управленческих решений
- Управление качеством программных систем
- Информатика
- Для разработчиков УМК
- Информационные технологии
- Консультация по математике (дифференцирование)

**Вход**

Вы используете гостевой доступ (Вход)

Логин

Пароль

[Забыли пароль?](#)

**Календарь**

апреля 2012

Вс	Пн	Вт	Ср	Чт	Пт	Сб
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

**Пользователи на сайте**

(последние 5 минут)

- Гость
- Сурин Геннадий Сергеевич
- Флят Александр Юргисович
- Шлак Олег Анатольевич

**Новостной форум**

28 апр 09:33  
Зорин Юрий  
Редактор формул еще...

7 окт 09:45  
Зорин Юрий  
Материалы квосов и MathML

Для входа в систему необходимо ввести пароль и логин. Для ознакомления организацией обучения используется открытый курс «Демонстрация возможностей системы Moodle для студентов».

## 7.9. Система контроля знаний ФДО ТУСУРа

Система контроля знаний ФДО ТУСУРа организована на основе систем Moodle и Лоцман.edu. В системе Лоцман.edu хранится база данных по успеваемости студентов. В системе Moodle хранятся тесты и протоколы сдачи экзаменов. В момент наступления сессии для конкретного студента открывается доступ к систем сдачи экзаменов. При этом ссылка «*Сдать экзамен*» в разделе «Сайт студента -> Учебный план» становятся доступна. Пример показан на рис. 7.1.

Дисциплина	Контроль	Часов Оценка	К. работ Дата	Л. работ Преподаватель	Аттестация
- Введение в специальность			1	0	Зачет;
Контрольная работа 1 (К)	У	24.12.09		компьютер	
Зачет	У	24.12.09			
- Высшая математика - 1			2	0	Зачет; Экзамен
Контрольная работа 1 (К)	У	23.12.09		компьютер	
Контрольная работа 2					<a href="#">Сдать</a> <a href="#">Вопрос</a>
Зачет	У	23.12.09			
Экзамен (К)					<a href="#">Сдать экзамен</a>

Рис. 7.1

Далее студент приступает к выполнению экзамена. На выполнение одного экзамена отведено максимум два астрономических часа, после истечения, которых автоматически выставляется оценка за экзамен. После завершения работы с экзаменом протокол сдачи фиксируется в базе данных Moodle, а в систем Лоцман.edu формируется зачетная или экзаменационная ведомость. Вся предварительная информация о работе студента с экзаменами доступна на странице «Сайт студента -> Результаты сессии». По окончании сессии доступ к экзаменам закрывается автоматически.

### 7.9.1. Выполнение экзаменационной работы в режиме online

После выбора ссылки «сдать экзамен» браузер автоматически переадресует студента на страницу с выбранной экзаменационной работой в среду СДО (рис. 7.2).

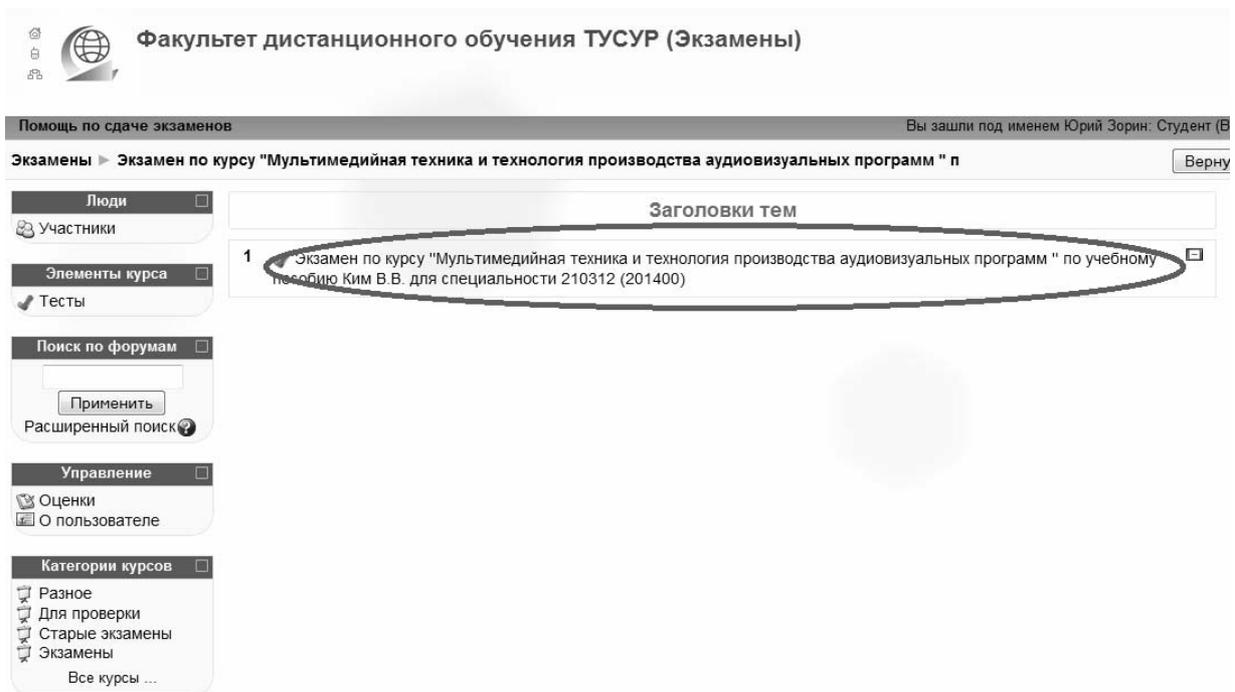


Рис. 7.2 — Страница СДО с экзаменационной работой

Перейдя по ссылке выбранной работы, следует нажать «**Начать тестирование**». На рисунке 7.3 изображена страница прохождения экзамена. Каждая работа имеет заставку и индивидуальные параметры (время, отведенное под тестирование, критерии оценок). Таймер в левом верхнем углу экрана показывает, сколько времени осталось до конца сдачи. Чтобы отправить Ваши ответы и завершить текущую попытку, нажмите кнопку «**Отправить все и завершить тест**» (рис. 7.4).

Экзамены ▶ Экзамен по курсу "Мультимедийная техника и технология производства аудиовизуальных программ " п ▶ Тесты ▶ Экзамен по курсу "Мультимедийная техника и технология производства аудиовизуальных программ " по учебному пособию Ким В.В. для специальности 210312 (201400) ▶ Попытка 1

Оставшееся время  
3:59:41

Экзамен по курсу "Мультимедийная техника и технология производства аудиовизуальных программ " по учебному пособию Ким В.В. для специальности 210312 (201400) - Попытка 1

1  
Баллов: 1

Для человеческого слуха и звукового восприятия укажите все верные утверждения.

Выберите по крайней мере один ответ:

- Человеческий слух позволяет определить позицию звукового источника с учетом 22 различных параметров.
- Левое и правое ухо человека независимо воспринимают звуковой сигнал, поэтому считается, что слух человека является стереофоническим.
- Технология "binaural recordings" учитывает пространственное размещение источников звука и обеспечивает высококачественный трехмерный звук.
- Для каждого человека характерен индивидуальный алгоритм обработки мозгом поступивших звуковых сигналов, которые интерпретируются в знакомые человеку понятия.
- К сожалению, на способности восприятия человеком звуков очень слабо влияет предшествующий опыт, все определяется только природной предрасположенностью.
- Громкость звука в акустике измеряют по линейной шкале.

2  
Баллов: 1

В сетевых технологиях Интернета термин 'сервер' расшифровывается как:

Выберите по крайней мере один ответ:

- каждый из миллионов компьютеров, подключенных друг к другу при помощи самых разных по скорости и типу каналов;
- компьютер с набором специальных программ и документов, называемых Web-страницами, имеющих свой адрес в сети;
- серия документов, содержащих сведения о рекомендуемых стандартах и технологиях для применения в Интернете;
- поставщик услуг Интернета, который обычно имеет собственные сети компьютеров, подключенных к Интернету высокоскоростными каналами связи;
- компьютер или программы, работающие на компьютере, предназначенные для обслуживания пользователей

Рис. 7.3 — Страница прохождения контрольной работы

12  
Баллов: 1

В технологии цифрового синтеза звука понятие 'технология сжатия данных, кодирования и смешивания звуковых каналов' обозначается термином:

Выберите по крайней мере один ответ:

- 'психоакустическое' сжатие;
- FM-операторы;
- AC-3 (Audio Coding, ver.3);
- WT (Wave Table);
- Sample;
- Bank;
- Dolby Digital 5.1;
- FM (Frequency Modulation).

Сохранить, но не отправлять

Отправить всё и завершить тест

Рис. 7.4 — Страница прохождения контрольной работы. Завершение попытки

После того, как Вы завершили выполнение экзамена, Вашему вниманию будет представлена страница (рис. 7.5) с результатами и оценка. Результат заносится в «Учебный план» в поле с соответствующей экзаменационной работой на следующий календарный день.

Сумма ваших предыдущих попыток				
Попытка	Завершено	Баллов / 24	Оценка / 100	Комментарий
1	четверг 19 мая 2011, 09:51	0	0	2
2	четверг 19 мая 2011, 15:18	12	50	3
3	четверг 19 мая 2011, 15:27	0	0	2

Высшая оценка: 50 / 100.

Комментарий к тесту

3

Сделать тест заново

Рис. 7.5 — Страница с результатами выполнения экзаменационной работы

## 7.10. Типы вопросов

Каждый экзамен в системе может содержать несколько вопросов различных типов. Поддерживается 4 типа вопросов:

1. Меню.
2. Ввод ответа.
3. Вопрос с пропусками.
4. Множественный выбор.

Рассмотрим каждый из этих типов более подробно.

### 7.10.1. Меню

Вопрос состоит из формулировки и множества вариантов ответа. Каждый вариант ответа может быть правильным или неправильным. На рисунке 7.6 представлен пример вопроса с единственным верным вариантом ответа. В вопросе на рисунке 6 верных ответов может быть несколько. В крайних случаях все варианты могут быть правильными или неправильными.

**1** Найти преобразование Фурье данной функции.  
Баллов: -1

$$f(x) = (x - 1)e^{-ix}$$

Выберите один ответ.

a.  $-\frac{\sqrt{2}(1 + \omega^2 + 2\omega i)}{\sqrt{\pi(\omega^2 + 1)}}$

b.  $-\frac{\sqrt{2}(1 + \omega^2 + 2\omega)}{\sqrt{\pi(\omega^2 + 1)^2}}$

c.  $-\frac{\sqrt{2}(1 + \omega^2 + 2\omega i)}{\sqrt{\pi(\omega^2 + 1)}}$

d.  $-\frac{\sqrt{2}(1 + \omega^2 + 2\omega)}{\sqrt{\pi(\omega^2 + 1)^2}}$

Рис. 7.6 — Вопрос с единственным вариантом ответа

**1** Укажите возможные способы выделения данных в MS-Excel с помощью мыши

Баллов: -/1

Выберите по крайней мере один ответ:

- a. При нажатых клавишах Alt+Shift
- b. При нажатой клавише Alt
- c. При нажатой клавише Shift
- d. При нажатых клавишах Ctrl+Shift
- e. При нажатой клавише Ctrl
- f. При нажатых клавишах Alt+Ctrl

Рис. 7.7 — Вопрос с множественным количеством ответов

### 7.10.2. Ввод ответа

Вопрос содержит формулировку и правильный ответ (либо несколько ответов). Ответом может быть строка символов, число, либо список слов или чисел. При сравнении введенного ответа с правильным, не учитывается регистр букв и наличие лишних пробелов. Верный ответ может иметь несколько вариантов. Пример: Кто написал поэму «Евгений Онегин»? Правильными будут ответы: Пушкин, А.С. Пушкин, Пушкин А.С. Числовой ответ может быть записан в виде десятичной дроби (0.056) или с использованием мантиссы (56E-3). По умолчанию дробная часть отделяется точкой, однако не забывайте руководствоваться текстом и комментариями вопроса во избежание получения неверного результата. На рисунке 7.8 представлен пример вопроса с вводом ответа.

**1** Сравните точность обычных механических бытовых часов с суточным ходом 10 секунд (суточный ход - величина, на которую часы уходят вперед - 'спешат' или отстают за сутки) и бытовых кварцевых часов, нестабильность частоты кварца у которых  $5 \cdot 10^{-6}$  за сутки. Во сколько раз кварцевые часы точнее?

Баллов: -/1

Расчетное значение округлить до целого числа.

Ответ:

Рис. 7.8 — Вопрос с вводом ответа

### 7.10.3. Вопрос с пропусками

Тип вопроса с пропусками предполагает наличие некоторого утверждения или вопроса, в котором некоторые слова пропущены и вместо них стоят прямоугольные области для ввода соответствующих слов. Для ответа на такой вопрос необходимо заполнить пропущенные места (рис. 7.9).

**1**  
Баллов: -/1  
Теплоёмкость электронного газа  зависит от температуры. Вставьте пропущенное слово.

Рис. 7.9 — Вопрос с пропуском

#### 7.10.4. Множественный выбор

В вопросе данного типа из заданного множества слов и словосочетаний необходимо выделить те, которые являются правильными в данном контексте вопроса (рис. 7.10).

**1**  
Баллов: -/2  
«Преступление и наказание» — роман , впервые опубликованный в 1866 году в журнале «Русский вестник». Сюжет разворачивается вокруг главного героя,  в голове которого созревает теория преступления.

- Родиона Раскольникова
- Павлика Морозова
- Евгения Онегина
- Порфирия Петровича

Рис. 7.10 — Вопрос типа «Множественный выбор»

## ЛИТЕРАТУРА

1. Компьютерные технологии в науке и образовании: методические указания к практическим занятиям / Сост. В.Н. Арефьев. — Ульяновск, Ул-ГТУ, 2001. — 42 с.

1. Изюмов А.А. Компьютерные технологии в науке и образовании: учеб. пособие / А.А. Изюмов, В.П. Коцубинский. — Томск: Эль Контент, 2012. — 150 с.

2. Кручинин В.В. Разработка компьютерных учебных программ. Изд-во ТГУ, 1998. — 211 с.

3. Кручинин В.В. Генераторы в компьютерных учебных программах. — Томск: Изд-во ун-та, 2003. — 200 с.

4. Медведев Д.С. Конспект лекций по курсу «Компьютерные технологии в науке и производстве в области электронной техники». — Томск: Эль Контент, 2012. — 125 с.

5. Львовский С. М. Набор и верстка в системе LaTeX. — 3-е изд., испр. и доп. — М.: МЦНМО, 2003. — 448 с.

6. Балдин Е.М. Компьютерная типография LaTeX. — СПб.: БВХ-Петербург, 2008. — 304 с.

7. Чичкарев Е.А. Компьютерная математика с Maxima: Руководство для школьников и студентов. — М.: AltLinux, 2009. — 233с. [<http://www.altlinux.org/Books:Maxima>]

8. Кручинин В.В. Комбинаторика композиций и ее приложение. — Томск: В-Спектр, 2010. — 156 с.

7. Изотов Д.А., Кручинин В.В., Кулик А.С., Тановицкий Ю.Н. Система моделирования электронных схем // Современное образование: Массовость и качество. Тез. док. Региональной науч.-метод. конф. 1—2 февр. 2001 г. Томский государственный ун-т систем управления и радиоэлектроники. — Томск: ТУСУР — С. 147—148.

8. Белозубов А.В., Николаев Д.Г. Система дистанционного обучения Moodle: учебно-методическое пособие. — СПб., 2007. — 108 с.

9. Анисимов А.М. Работа в системе дистанционного обучения Moodle: учеб. пособие. — 2-е изд. испр. и дополн. — Харьков: ХНАГХ, 2009. — 292 с.

[<http://moodle.org/mod/forum/discuss.php?d=132400>]

**Ресурсы**

1. Сайт MikTeX <http://www.miktex.org/>
2. Сайт Texmaker <http://www.xmlmath.net>
3. Comprehensive TEX Archive Network <http://www.ctan.org/>
4. Сайт ресурсов Latex <http://editorz.narod.ru/latex/>
5. Сайт Moodle <http://moodle.org>