

**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)**

И.М. Егоров

ИНФОРМАТИКА

**Руководство к организации
самостоятельной работы**

ТОМСК – 2007

Федеральное агентство по образованию

**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)**

Кафедра промышленной электроники (ПРЭ)

И.М. Егоров

ИНФОРМАТИКА

**Руководство к организации
самостоятельной работы**

2007

Егоров И.М.

Информатика: Руководство к организации самостоятельной работы. — Томск: Томский государственный университет систем управления и радиоэлектроники, 2007. — 51 с.

© Егоров И.М., 2007

© ТУСУР, 2007

СОДЕРЖАНИЕ

Рабочая программа курса	4
Порядок аттестации по лабораторному практикуму	17
<i>Лабораторная работа № 1. Файловая структура MS-DOS.</i>	
Команды MS-DOS	18
<i>Лабораторная работа № 2. Текстовый процессор MS WORD...</i>	21
<i>Лабораторная работа № 3. Табличный процессор MS Excel</i>	27
<i>Лабораторная работа № 4. СУБД MS Access</i>	29
<i>Лабораторная работа № 5. Mathcad: техника использования шаблонов при вводе формул</i>	31
<i>Лабораторная работа № 6. Матричные операции Mathcad</i>	32
<i>Лабораторная работа № 7. Интерполяция и регрессия выборочных данных.....</i>	34
<i>Лабораторная работа № 8. Ввод вещественного числа и управляющих символов на С++</i>	41
<i>Лабораторная работа № 9. Интерфейс ввода матриц на С++ ...</i>	43
<i>Лабораторная работа № 10. Объединения, битовые поля и поразрядные операции С++</i>	45
<i>Лабораторная работа № 11. Программирование односвязного списка.....</i>	48
Приложение А	50
Литература.....	51

РАБОЧАЯ ПРОГРАММА КУРСА

Федеральное агентство по образованию

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

УТВЕРЖДАЮ

Проректор по учебной работе

М.Т. Решетников

« _____ » _____ 2007 г.

РАБОЧАЯ ПРОГРАММА

По дисциплине: Информатика

Для специальности 210106 — Промышленная электроника
и направления 210100 — Электроника и микроэлектроника
(бакалавриат)

Факультет электронной техники

Профилирующая кафедра — Промышленной электроники

Курс I Семестр 1, 2

Учебный план набора 2006 г. и последующих лет

Распределение учебного времени, часы:

	1 семестр	2 семестр	Всего
Лекции	26	34	60
Лабораторные занятия	18	34	52
Практические занятия	18	18	36
Всего аудиторных занятий	62	86	148
Самостоятельная работа	46	94	140
Общая трудоемкость	108	180	288

Зачет 1 семестр

Экзамен 2 семестр

Рабочая программа составлена на основании ГОС ВО для специальности 210106 "Промышленная электроника", рассмотрена и утверждена на заседании кафедры « » _____ 2006 г., протокол № _____

Разработчик
доцент каф. ПрЭ

И.М. Егоров

Зав. выпускающей
кафедрой ПрЭ

А.В. Кобзев

Рабочая программа согласована с факультетом

Декан ФЭТ

В.М. Герасимов

1 ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1 Цель преподавания дисциплины

Целью преподавания дисциплины «Информатика» является изучение основ современной технологии обработки информации с использованием средств вычислительной техники, знакомство с популярными программными продуктами, применяемыми как в инженерных расчетах, так в офисных технологиях.

1.2 Задачи изучения дисциплины

Задача изучения дисциплины «Информатика» состоит в последовательном изложении студентам первого курса обширного ознакомительного материала по основам вычислительной техники и по ее применению в решении учебных и исследовательских задач. В ходе изучения дисциплины «Информатика» студенты знакомятся с принципами работы компьютера, получают представление о системном и прикладном программном обеспечении, овладевают навыками практической работы с компонентами MS Office, пакетом математических расчетов MathCad, овладевают основами программирования на языке C++.

1.3 Связь с другими дисциплинами

Дисциплина «Информатика» является базовой для последующих курсов, связанных с применением компьютерной техники в специальных задачах.

Содержание лекционного курса I семестра (26 часов)

1. *Предмет информатики и понятие информации* — 2 часа. Информация. Основные единицы количества информации. Синтаксический, семантический и прагматический аспекты информации. Создание, передача, прием и использование информации. Информационные ресурсы, информационные процессы. Влияние информационных ресурсов на развитие общества. Информатизация.

2. *Технические средства обработки информационных процессов* — 2 часа.

Компьютер как программный автомат. Архитектура компьютера. Процессор. Информационно — управляющая магистраль (шина), память, контроллеры устройств. Внешние устройства. Монитор, клавиатура, принтер, сканер, сетевое оборудование. Соотношение аппаратной и программной составляющих в вычислительных системах.

3. *Человек и компьютер* — 2 часа. Факторы производственной вредности при работе с компьютером. Взаимодействие человек — компьютер: психологический, гигиенический и физиологический аспекты. Синдром компьютерного стресса. Профилактика негативных последствий длительной работы с компьютером.

4. *Системное программное обеспечение* — 4 часа.

4.1. *Операционная система (ОС) компьютера* — 2 часа. Назначение ОС и ее роль как связующего звена между аппаратурой ЭВМ и пользовательскими программами. Основные подсистемы ОС: файловая, ввода-вывода, обработки ошибок. Выполнение программ, понятие прерывания.

4.2. *Однозадачные и многозадачные ОС* — 2 часа. Однозадачная система MS-DOS. Многозадачная система WINDOWS, разделение времени, событийное программирование.

5. *Хранение и обработка данных в компьютере* — 4 часа (14).

5.1. *Представление данных в компьютере* — 2 часа. Структуры данных. Хранение информации. Адрес памяти. Постоянная и временная память. Структура памяти. Носители информации. Устройства прямого и последовательного доступа. Файлы.

5.2. *Преобразование информации в компьютере* — 2 часа. Тип данных как основополагающая интерпретирующая категория в программировании. Операнды и операции. Структура команд процессора. Код операции. Адресация.

6. *Выполнение арифметических операций в компьютере* — 2 часа (16). Машинная арифметика. Интерпретация арифметических действий. Позиционные системы счисления. Арифметика по конечному модулю. Число и его дополнение. Двоичная система счисления. Системы счисления с основанием 2^k . Перевод чисел из одной системы в другую. Двоичная арифметика. Прямой, инверсный и дополнительный код.

7. *Прикладное программное обеспечение общего назначения. Элементы офисных технологий* — 10 часов.

7.1. *Подготовка, редактирование и оформление документации в Microsoft Word* — 2 часа. Конвертирование текстов различного формата при вводе и выводе. Использование буфера обмена в качестве универсального контейнера при межпрограммной передаче информации. Вид редактируемого документа и настройка панелей. Вставка в текст объектов — рисунки, диаграммы, рамки текста. Редактор формул. Автоматическая проверка правописания. Работа с таблицами. Автоматизация работы — создание шаблонов и стилей, оглавления и указатели, использование макросов. Сноски, перекрестные ссылки и гиперссылки. Использование помощи.

7.2. *Работа с электронными таблицами в Microsoft Excel* — 4 часа. Концепция электронной таблицы. Ячейки, листы и книги. Типы данных и формат ячеек. Редактирование и вычисление. Адресация данных. Ссылки. Редактирование таблицы. Автоматическая модификация относительных адресов в формулах. Автоматическое заполнение областей. Обзор библиотеки встроенных функций. Обработка массивов, средства визуального задания границ. Фильтрация данных. Построение диаграмм.

7.3. *Система управления базами данных Microsoft Access* — 2 часа. Первоначальные сведения о базах данных и их организации. Хранение информации в базах данных. Записи и поля. Компоненты Access: формы, запросы, отчеты, модули. Создание и редактирование таблиц. Фильтрация и сортировка данных в таблицах и запросах. Импорт внешних данных и связывание таблиц. Функции запросов: выборка, изменение, дополнение, удаление. Визуальные средства построения запросов. Отчеты в Access. Экспорт отчетов в Microsoft.

7.4. *Создание презентаций с помощью Microsoft PowerPoint* — 2 часа.

Содержание лекционного курса II семестра (34 часа)

8. *Компьютер в математических расчетах. Математический процессор MathCad* — 4 часа.

8.1. *Назначение, состав и техника работы в среде MathCad* — 2 часа. Интерфейс программы. Рабочее поле

(Workseet). Регионы MathCad, порядок выполнения вычислений. Техника навигации на рабочем листе. Палитра инструментов. Шаблоны операций. Построение графиков.

8.2. *Некоторые математические задачи их решение в среде MathCad* — 2 часа. Интерполяция и аппроксимация выборочных данных. Вычисление интегралов и производных. Работа с матрицами и векторами. Символьный процессор. Автоматические и ручные символьные вычисления.

9. *Основные категории программирования* — 2 часа. Понятие алгоритма. Базовые понятия переменной и константы. Принцип структурного программирования. Структура линейного алгоритма. Структура разветвляющегося алгоритма. Понятие цикла в структурном программировании. Рекурсивные алгоритмы. Итерационные алгоритмы. Понятия о процедурном и объектном программировании. Модульный принцип программирования. Декомпозиция задачи на самостоятельные функциональные блоки. Принцип инкапсуляции данных в модулях. Создание собственных типов программных объектов, наделенных внутренними и внешними свойствами.

10. *Основы программирования на языке C++* — 26 часов.

10.1. *Начальные сведения о языках C и C++* — 2 часа.

Краткая историческая справка развития языков программирования. Соотношение языков C и Pascal. Развитие C в C++. Алфавит символов и лексемы языка C++: константы, идентификаторы, ключевые слова, комментарии в программе.

10.2. *Типы данных C++* — 2 часа. Категории типов данных. Предопределенные и определяемые пользователем типы данных. Арифметические типы: целый, вещественный. Символьный тип. Интерпретация логических значений.

10.3. *Указатели и ссылки* — 2 часа. Работа с указателями и ссылками. Адресная арифметика C++. Операции с указателями. Контроль адресной арифметики. Указатель на **void**. Инициализация указателей и ссылок. Значение указателя **NULL**. Доступ к элементам массива через указатель. Доступ к элементам объектов типов по указателю. Многомерные массивы, массивы указателей и указатели на указатели. Динамическое создание объектов.

10.4. *Агрегатные типы* — 2 часа. Составные типы: массивы. Объектные типы: структуры и объединения. Битовые поля. Определение типов и инициализация данных. Перечислимый тип.

10.5. *Операции и выражения* — 2 часа. Выражения: операнды, операции, значение. Автоматическое преобразование типов в выражениях. Мультипликативные операции и аддитивные арифметические операции. Логические операции. Поразрядные операции. Операции разыменования и взятия адреса. Контекстная зависимость интерпретации знаков операций. Операция получения размера `sizeof`. Условная операция. Операции инкремента и декремента: постфиксная и префиксная формы. Простое и составное присваивание. Операция явного преобразования типа. Приоритеты операций и порядок вычислений. Операции доступа и расширения контекста.

10.6. *Операторы C++* — 2 часа. Оператор выражение. Пустой оператор. Составной оператор. Условные операторы **if** и **if ...else**. Оператор выбора **switch**. Операторы цикла: **for**, **while** и **do ... while**. Операторы управления: **break**, **continue**, **return** и **goto**. Операторы управления динамической памятью **new** и **delete**.

10.7. *Функции C++* — 6 часов.

Функции как основа программирования на C++. Объявление, определение и оператор вызова функций на C++. Инкапсуляция данных внутри функции. Функция **main**-головная программа. Передача параметров функции **main**. Интерфейс функции: прием и передача информации. Понятие побочного эффекта. Защита от нежелательных побочных эффектов. Передача параметров через указатель и по ссылке. Контроль типов параметров, передаваемых при вызове функций и их автоматическое преобразование. Функции с переменным числом параметров. Сигнатура функции в C++. Понятие перегрузки функций. Эквивалентность функций и операторов в C++, перегрузка операторов.

10.8. *Время жизни и область видимости программных объектов* — 2 часа.

Использование стека и сегмента данных для хранения переменных. Возникновение программных объектов и их уничтожение. Области видимости блок и файл. Скрытие имен объектов и расширение контекста. Понятие класса хранения данных. Клас-

сы хранения **register**, **auto**, **static**, **extern**. Класс памяти **static**. Инициализация глобальных и локальных перемен

10.9. *Структура программы C++* — 2 часа. Исходные файлы и заголовочные **h**-файлы. Объявление переменных и функций в заголовочных файлах. Препроцессор C++ и его функции. Управление препроцессором. Директивы препроцессора **#include** и **#define**.

10.10. *Обработка исключительных ситуаций в программах C++* — 2 часа. Нештатные ситуации времени выполнения, средства их контроля и обработки.

Темы лабораторных работ I семестра (18 часов)

1. *Работа под операционной системой MS DOS (эмуляция)* — 2 часа. Команды управления файловой системой. Просмотр содержимого тома. Создание, редактирование и удаление текстовых файлов. Перенаправление выходного потока в заданный текстовый файл.

2. *Работа с программным обеспечением Microsoft Office* — 16 часов.

2.1. *Подготовка документов в Microsoft Word*. Набор текста с его автоматическим форматированием. Использование стилей. Создание таблиц в Microsoft Word и работа с ними. Использование мастера таблиц. Создание и использование рисунков с использованием векторной графики Microsoft Word. Дополнительные возможности. Редактор формул. Диаграммы и графики. Word Art и др. Формирование сносок, ссылок и гиперссылок — 4 часа.

2.2. *Работа с Microsoft Excel* — 4 часа. Ввод и редактирование данных. Формулы и функции. Работа с электронной таблицей как с базой данных. Сортировка и фильтрация данных. Работа с датами. Работа с матрицами.

2.3. *Работа с Microsoft Access* — 4 часа. Создание простейшей базы данных. Обработка данных с помощью запросов: группировка, сортировка и фильтрация. Редактирование базы данных. Формирование отчетов.

2.4. *Составление простых документов в формате html* — 4 часа.

Темы лабораторных работ II семестра (34 часа)

3. *Математический процессор Mathcad* — 8 часов.

3.1. *Одномерная и двумерная интерполяция и аппроксимация (регрессия) выборочных данных* — 4 часа. Производные. Построение графиков функций одной и двух переменных.

3.2. *Работа с матрицами, векторами. Решение уравнений.* — 4 часа. Решение систем уравнений и неравенств.

4. *Программирование на C++* — 26 часов.

4.1. *Составление простейших программ* — 4 часа. Использование библиотечных функций ввода/вывода и потоков ввода/вывода. Управление выводом на консоль. Файловый ввод и вывод символьной информации.

4.2. *Операторы ветвления и итерации* — 4 часа.

4.3. *Работа с массивами, указателями и динамической памятью.* — 4 часа.

4.4. *Структуры, объединения, битовые поля. Поразрядные операции* — 4 часа.

4.5. *Функции C++.* Составление программ с использованием функций. Передача параметров по значению и по ссылке. Массивы в качестве параметров функции. Блокировка нежелательных побочных эффектов — 4 часа.

4.6. *Перегрузка функций и функции — операторы C++* — 4 часа.

4.7. *Многофайловые программы C++* — 2 часа. Проекты. Собственные `h` — файлы.

Темы практических занятий I семестра (18 часов)

1. Представление чисел в различных системах счисления — 2 часа.

2. Арифметика по модулю, прямой и дополнительный код — 2 часа.

3. Настройка параметров рабочей среды и панели инструментов и Microsoft Word — 2 часа.

4. Настройка параметров рабочей среды и панели инструментов Microsoft Excel — 2 часа.

5. Алгоритмизация векторно-матричных операций в Excel — 2 часа

6. Элементарные сведения об организации баз данных —

2 часа.

7. Массивы данных. Поиск, сортировка, групповые операции — 2 часа.

8. Элементы работы с гипертекстом — 2 часа.

9. Создание отчета-презентации по лабораторным работам — 2 часа.

Темы практических занятий II семестра (18 часов)

1. Задачи интерполяции и аппроксимации при обработке выборочных данных на MathCad — 2 часа.

2. Операции матричной алгебры в MathCad — 2 часа.

3. Численные методы решения алгебраических уравнений — 2 часа.

4. Численные методы решения дифференциальных уравнений — 2 часа.

5. Библиотека функций Си. Заголовочные файлы — 2 часа.

6. Работа со структурами и объединениями — 2 часа.

7. Программирование списков — 2 часа

8. Создание «безразмерного» массива строк произвольной длины и работа с ним — 2 часа.

9. Массивы списков. Использование кэш-функций — 2 часа.

Темы контрольных работ I семестра

1. Архитектура ЭВМ. Операционные системы MS DOS и WINDOWS. Системы счисления и арифметика по конечному модулю.

2. Операции матричной алгебры в среде MS Excel.

Темы контрольных работ II семестра

1. Типы данных и операции C++. Области видимости, классы памяти и время жизни объектов в программах на C++

2. Основные конструкции структурного программирования и их реализация на языке C++.

3. Функции на языке C++. Описание и вызов. Обмен информацией с вызывающей программой.

Тема индивидуального задания I семестра

Создание отчета по лабораторной работе №1 в виде презентации PowerPoint.

Темы индивидуальных заданий II семестра

1. Решение оптимизационных задач в среде MathCad.
2. Реализация на C++ метода исключения переменных Гаусса в системе линейных алгебраических уравнений.

Содержание самостоятельной работы (140 часов)

№	Наименование работы	Кол-во часов	Форма контроля
1	Проработка лекционного материала	36	Зачет, экзамен
2	Выполнение индивидуальных заданий	30	Проверка индивидуальных заданий
3	Подготовка к лабораторным работам	40	Защита лабораторной работы. Допуск к зачету и экзамену
4	Подготовка к практическим занятиям и к выполнению контрольных работ	34	Устный опрос на практическом занятии. Проверка контрольных работ

Рейтинговая раскладка лабораторных, контрольных и индивидуальных работ I семестра

Наименование работы	Рейтинг
Лабораторные работы	
1. Работа под операционной системой MS DOS	10
2. Подготовка документов в Microsoft Word	12
3. Работа с Microsoft Excel	12
4. Работа с Microsoft Access	10
5. Работа с Microsoft PowerPoint	10
6. Составление простых документов в формате html	10

Наименование работы	Рейтинг
Контрольные работы	
<i>1. Архитектура ЭВМ. Операционные системы MS DOS и WINDOWS. Системы счисления и арифметика по конечному модулю</i>	16
<i>2. Операции матричной алгебры в среде MS Excel</i>	20
Индивидуальное задание	
<i>Создание отчета по лабораторной работе №1 в виде презентации PowerPoint</i>	20
<i>Итого</i>	120

Рейтинговая раскладка лабораторных и контрольных работ II семестра

Наименование работы	Рейтинг
Лабораторные работы	
<i>1. MathCad. Одномерная и двумерная интерполяция и аппроксимация (регрессия) выборочных данных</i>	10
<i>2. MathCad. Работа с матрицами, векторами. Решение уравнений</i>	10
<i>3. Работа с массивами, указателями и динамической памятью</i>	10
<i>4. Структуры, объединения, битовые поля</i>	12
<i>5. Функции C++.</i>	12
<i>6. Перегрузка функций и функции-операторы</i>	10
<i>7. Многофайловые программы C++</i>	6
Контрольные работы	
<i>1. Типы данных и операции C++. Области видимости, классы памяти и время жизни объектов в программах на C++</i>	10
<i>2. Основные конструкции структурного программирования и их реализация на языке C++</i>	10

Наименование работы	Рейтинг
<i>3. Функции на языке C++. Описание и вызов. Обмен информацией с вызывающей программой</i>	10
<i>Индивидуальные задания</i>	
<i>1. Решение оптимизационных задач в среде MathCad</i>	10
<i>2. Реализация на C++ метода исключения переменных Гаусса в системе линейных алгебраических уравнений</i>	10
<i>Итого:</i>	120

Литература

Основная:

1. Симонович С.В. Информатика. Базовый курс: Учебник для вузов. — 2-е изд. — СПб.: Питер, 2006. — 640 с.
2. Острейковский В.А. Информатика: Учебник для вузов. — М.: Высшая школа, 2001. — 512 с
3. Макарова Н.В. [и др.] Информатика: Практикум по технологии работы на компьютере: Учебное пособие для вузов. — 3-е изд., перераб.. — М.: Финансы и статистика, 2005. — 255 с.
4. Павловская Т.А. C/C++. Программирование на языке высокого уровня: Учебник для вузов. — СПб.: Питер. 2006. — 461 с.

Дополнительная

1. Воройский Ф.С. Информатика. Новый систематизированный толковый словарь-справочник — М.: ФИЗМАТЛИТ, 2003. — 760 с.
2. Пасько В. Энциклопедия ПК. Аппаратура. Программы. Интернет. — СПб.: Питер, 2004. — 800 с.
3. Волкотруб Л.П., Егоров И.М. Компьютер и здоровье. — Томск: Томск. гос. ун-т систем упр и радиоэлектроники, 2006. — 158 с.
4. Дьяконов В. Mathcad 2001: Учебный курс — СПб.: Питер, 2001.

ПОРЯДОК АТТЕСТАЦИИ ПО ЛАБОРАТОРНОМУ ПРАКТИКУМУ

Выполнение лабораторных работ допустимо по бригадному принципу, когда одна программа создается группой не более, чем из 2—3 человек, но зачет по лабораторным работам производится по результатам ее защиты, проводимой в форме *индивидуального* собеседования. На защиту принимается полностью отлаженная и, безусловно, работоспособная программа.

В процессе защиты лабораторной работы *каждый* студент должен быть готов дать комментарии по любому фрагменту защищаемой работы. С целью упрощения чтения программы ее текст целесообразно снабдить необходимыми комментариями.

Оценка работ производится по рейтинговой системе: каждой работе сопоставлен максимальный рейтинговый балл, начисляемый в случае ее успешной защиты.

Если студент претендует на более высокую оценку, чем исчисляемая по рейтингу, то он может пройти индивидуальное собеседования по материалам всех выполненных им работ и получить дополнительные 25% баллов, при условии, что сумма баллов при этом не превысит 120.

Оценка качества программы зависит не столько от факта ее работоспособности, сколько от того, насколько качественно написан ее исходный код.

Внимание! Для решения возможных спорных вопросов, а также для прохождения заключительного собеседования, сохраняйте файлы с исходными текстами программ до конца семестра.

ЛАБОРАТОРНАЯ РАБОТА № 1

Файловая структура MS-DOS. Команды MS-DOS

После загрузки Windows активизируйте режим эмуляции MS-DOS:

1. В меню «Пуск» выберете пункт «Выполнить»
2. В открывшемся диалоговом окне набрать команду:

cmd — при работе под Windows XP;

command.com — при работе под Windows 98;

и выполнить ее. <Enter>

На экране должно появиться окно эмулятора, имитирующее экран монитора в системе DOS (стандартные установки: цвет фона — черный, цвет шрифта — серый). В нижней части экрана выводится строка-«приглашение», содержащая путь к директории, установленной в данный момент в качестве текущей. Команды DOS вводятся в конец строки-приглашения и активируются нажатием клавиши. <Enter>

Завершение работы эмулятора DOS с выходом в Windows происходит после ввода команды **exit**.

Все последующие пункты задания выполняются с помощью команд MS-DOS.

Создать систему каталогов, используя команду создания каталога **md <имя>**.

1. Создать в корневом каталоге диска каталог **TEST**.
2. Создать внутри него два подкаталога с именами **KAT1** и **KAT2**.
3. Внутри каталога **KAT1** создать подкаталог **KAT11**.
4. Создать в подкаталоге **KAT1** текстовый файл **primer1.txt** и занесите в него текст из нескольких строк:

```
"student gruppy 367-8"
```

```
"familija imja otchestvo"
```

5. Создать в подкаталоге **KAT2** текстовый файл **primer2.txt** и занесите туда строку "moi den rojdeija 10/09/1989"

Указания. Для создания файлов используйте команду копирования консоли в файл **copy con**. Текст набирается на экране монитора, окончание ввода текста — команда **<Ctrl>+<Z>** или **<F6>**, затем **<Enter>**. Появление файлов в каталогах проверьте с помощью команды чтения каталога **dir <имя каталога>**.

Команда **dir** без указания имени каталога дает оглавление текущего активного каталога.

Просмотрите командой **dir /p** содержимое корневого каталога активного диска (активизация корневого каталога диска — команда **cd**).

Переход в дочерний подкаталог активного каталога производится командой **cd <имя каталога>**, возврат в родительский каталог — командой **cd**.

6. В подкаталоге **KAT1** создать текстовый файл с именем **d1.txt**, перенаправив в него вывод команды **dir** каталога **KAT1**.

7. В подкаталоге **KAT2** создать текстовый файл с именем **d2.txt** путем вывода в него выходной информации **dir** каталога **KAT2**.

8. Скопируйте в подкаталог **KAT2** с помощью команды, **copy** файл **primer1.txt** и **primer2.txt**, объединив их в один файл **primery.txt**

9. Выведите на экран содержимое файла **primery.txt** командой **type**.

10. Переместите файл **primer2.txt** в подкаталог **KAT1**. Для этого скопируйте его в подкаталог **KAT1** и затем удалите его в подкаталоге **KAT2** командой **del primer2.txt**

11. В каталог **KAT11** поместите копии файлов **primer1.txt** и **primer2.txt**.

12. Создать в каталоге **TEST** файлы-справочники для всех использованных в этой работе команд DOS путем перенаправления в них выходных потоков команд, вызванных с параметром **/?** — справка по команде. Имена файлов-справочников рекомендуется формировать по имени команды с расширением **.txt**. Например, с текстом справки команды **dir** — файл **dir.txt**, для

команды **copy** — файл **copy.txt** и т.п.

13. Создать в каталоге **TEST** файл *tree.txt* путем направления в него выходного потока команды **tree test > tree.txt**.

Отчетным документом по работе служит содержимое каталога **TEST** и его структура, отраженная деревом в файле **test\tree.txt**

ЛАБОРАТОРНАЯ РАБОТА №2 **Текстовый процессор MS WORD**

Задание на лабораторную работу

С помощью контекстного меню создать на рабочем столе папку **ТЕХТ**. Запустите приложение *MS Word*.

1. Подготовьте редактор к вводу нового текста. Для этого:

- Установите размеры листа и полей.

Размер бумаги — 210x297 (A4);

поля:

Верхнее — 2 см

Нижнее — 2 см

Левое — 3 см

Правое — 2 см.

От края до колонтитулов

верхнего — 1,5

нижнего — 1 см

- Установите формат абзаца

Отступ слева — 0 см

Отступ справа — 1 см

Первая строка — Отступ 1 см

Междустрочный интервал — одинарный

Выравнивание — по левому краю.

- Используя панель инструментов, установите:

Шрифт — Arial

Размер шрифта — 10 pt

- Установите отображение непечатаемых символов (Пробелы; Символы конца абзаца), используя меню «Сервис — Параметры... — Вид».

- Откажитесь от автоматического переноса слов.

- Создать элементы *Автотекста* для слов **Windows, Paintbrush, Word, Excel**.

2. Введите предложенный ниже текст, используя *Автотекст* и сохраняя текстовые выделения. Клавишу <Enter> ис-

пользуйте только в конце абзаца.

Каждая прикладная программа (*приложение*) Windows чаще всего рассчитана на конкретный объект обработки (*документ*). Например, *графический редактор* Paintbrush работает с изображениями (картинками), *текстовый процессор* Word — с текстовыми документами, *табличный процессор* Excel — с электронными таблицами, информационное приложение — с таблицами баз данных и т. д.

Как правило, каждый такой объект мы храним на диске в виде отдельного *файла*, формат которого определен соглашениями данной программы. *Формат* файла обычно ассоциируется с *расширением имени*: например, картинки хранятся в графическом файле *.bmp* или *.rct*, тексты — в файлах с расширением имени *.doc*, электронные таблицы — в файлах *.xls*.

3. Проверьте правильность написания слов (*орфографию*).
4. Вставьте над текстом заголовок РАБОТА С ФАЙЛАМИ и в следующую строку — свою фамилию и инициалы.
5. Отцентрируйте заголовки.
6. Выделите заголовок и примените к нему особый шрифт (например, *Жирный*), а затем стиль (например, *Заголовок 3*), используя соответствующие элементы панели инструментов *Форматирование*.
7. Сохраните текст на диске в папке ТЕХТ в файле с именем Text1.doc.
8. Закройте окно документа. Произведите «загрузку» Вашего документа с диска.
9. Выполните форматирование (выравнивание по ширине) всего текста.
10. Разбейте текст на абзацы по предложениям.
11. Переформатируйте весь текст, установив *Отступ справа* 3 см.
12. Переформатируйте первый абзац, установив для него *Отступ справа* 5 см.
13. Переформатируйте второй абзац, установив для него *Отступ слева* 4 см, *Отступ справа* 4 см, и *Первая строка* — *Отступ* 1 см.
14. В режиме *Предварительный просмотр* просмотрите

расположение текста на бумаге в форме *Книжная* и *Альбомная*.

15. Выполните перестановку первого абзаца в конец текста, используя механизм перетаскивания с помощью мыши.

16. Перенесите второй абзац текста в другой документ, предварительно его создав. Сохраните новый документ в папке **ТЕХТ** в файл с именем **Text2.doc**.

17. Выберите в тексте какое-либо слово и выполните его поиск средствами программы MS Word.

18. Замените в тексте все буквы «а» на цифру «5». Выполните обратную замену.

19. Выделите какое-либо слово в тексте курсивом и какое-либо слово подчеркиванием.

20. Скопируйте свою фамилию из 2-й строки в правую часть строки после текста, используя панель инструментов.

21. Сделайте рамку вокруг заголовка текста и установите цветной фон.

Верстка, колонтитулы

1. Запустите программу MS Word. Откройте документ с именем **Text1.doc**

2. Скопируйте введенный текст многократно (30...35 раз) для получения длинного текста.

3. Установите режим *Разметка страницы*.

4. Вставьте после первого абзаца **Разрыв** страницы (команда **<Ctrl> + <Enter>**). Просмотрите разбиение текста на страницы.

5. Выполните двухколончатую верстку для фрагмента текста, занимающего две страницы:

6. просмотрите размещение текста на странице;

7. верните текст к обычному одноколончатому размещению.

8. Пронумеруйте пять абзацев текста, т. е. создать пронумерованный список. Из других пяти абзацев создать маркированный список.

9. Вставьте номера страниц, начиная с первой, располагая номер внизу страницы по центру. Просмотрите текст.

10. Оформите верхний колонтитул в виде:

Томский государственный университет систем управления и радиоэлектроники

Дата стр. из

Указания:

Текст колонтитула выполните шрифтом **Times New Roman 12**.

Для ввода текущей даты и номера страницы воспользуйтесь пиктограммами.

11. Обведите колонтитул рамкой.

12. Просмотрите расположение текста на страницах в режиме *Предварительный просмотр*, установив режим *Несколько страниц*.

13. Сохраните документ в файле с именем **Text5.doc** в папке **ТЕХТ**.

Вставка формул

1. Создать новый документ и введите в него формулы путем вставки объекта *Microsoft Equation*:

введите формулу

$$\int_0^{\pi} \sin(x) dx = 2$$

2. Запишите, не пользуясь *Microsoft Equation* квадратное уравнение:

$$ax^2 + bx + c = 0$$

3. С помощью средств *Microsoft Equation* запишите формулу для нахождения корней квадратного уравнения:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

4. Сохраните документ в файле с именем **Text6.doc** в папке **ТЕХТ**.

5. Создать новый документ и введите в него блок-схему решения квадратного уравнения, пользуясь средствами рисования программы MS Word. Внутри блоков вместо формул используйте текстовые пояснения.

6. Сохраните документ в файле с именем **Text7.doc** в папке **ТЕХТ**.

Работа с таблицами

Создайте новый документ MS Word.

1. В меню «Таблица» включите режим *Сетка*. Выберите режим *Вставить таблицу*. В диалоговом окне установите:

Число столбцов — 4;

Число строк — 4.

2. Установите границы столбцов на позициях линейки 0, 3, 8, 11, 16, перемещая мышью непосредственно границы столбцов или табулятопы на координатной линейке.

3. Выделите всю таблицу и установите формат абзаца:

Поле Левое — 0,

Правое — 0 см,

Первая строка — Нет.

4. В режиме *Формат — Границы и заливка* снимите оформление таблицы (если оно существует).

Снимите выделение таблицы.

5. Заполните таблицу по предлагаемому ниже образцу (без обрамления). Одинаковое содержание ячеек таблицы вводите с помощью процедуры копирования.

Программный продукт	Функциональное назначение	Операционная система	Размер дисковой памяти, Мб
EXCEL 2000	Табличный процессор	Windows	60,0
LOTUS 2.0	Табличный процессор	MS-DOS	12,0
ACCESS 2000	СУБД	Windows	54,0
FOXPRO 2.0	СУБД	MS-DOS	8,0
MS WORD 2000	Редактор текстов	Windows	72,0
ЛЕКСИКОН для Windows	Редактор текстов	Windows	7,0

6. Отцентрируйте заголовки столбцов и числовые значения в правом столбце.

7. Добавьте в таблицу еще одну строку и вычислите сумму чисел в последней колонке таблицы.

8. Измените значение в одной из ячеек последнего столбца и пересчитайте значение суммы (команда *Обновить поле* или <F9>).

9. Уберите (выключите) сетку таблицы.

10. Создать обрамления:

ячеек таблицы линиями толщиной 1,5 пт

всей таблицы линией толщиной 2,5 пт.

11. Отделите заголовки столбцов от остальной таблицы линией толщиной 2,5 пт.

12. Выполните сортировку строк таблицы по возрастанию значений в столбце 4.

13. Примените к созданной таблице *Автоформат*, предварительно выделив всю таблицу и выбрав формат *Стандарт 3*.

14. Сохраните работу в папке ТЕХТ в файле Таблица.doc.

По окончании работы файлы папки ТЕХТ предъявляются преподавателю.

ЛАБОРАТОРНАЯ РАБОТА № 3

Табличный процессор MS Excel

Параметрическая идентификация динамической системы

Имеется динамическая система, на вход которой подается тестовый сигнал $X(t)$ в виде единичной ступеньки, сигнал на выходе системы $Y(t)$ фиксируется в равноотстоящие моменты времени t_k и записывается в таблицу в виде последовательности y_k .

Аналитического выражение для выходного сигнала — его параметрическая модель — имеет вид:

$$Y_m(t, a, b) = 1 - \frac{\sqrt{a^2 + b^2}}{b} e^{-at} \sin\left(bt + \arcsin\left(\frac{b}{\sqrt{a^2 + b^2}}\right)\right),$$

где параметры **a** и **b** неизвестны.

Задача состоит в том, чтобы на основе наблюдаемой выборки y_k получить значения параметров **a** и **b**. Такие задачи называют задачами *параметрической идентификации*, их решение осуществляется *методом наименьших квадратов*, суть которого сводится к нахождению значений неизвестных параметров из условия минимума суммы квадратов невязок между значениями выхода модели и наблюдаемыми данными.

$$I(a, b) = \sum (y_k - Y_m(t_k, a, b))^2 \xrightarrow{a, b} \min$$

Задание 1. Построить функцию $Y_m(t, a, b)$, зависящую от трех параметров. Значения параметров **a** и **b** должны располагаться в ячейках, ссылки на которые формируется по *абсолютной адресации*. Ссылку на ячейку для t выполнить по *относительной адресации*.

Рекомендация: ввиду сложности формулы, ее вычисление целесообразно построить в несколько этапов с использованием ячеек для промежуточных результатов.

Задание 2. Построить ряд из 10—15 значений для t_k в каком-либо столбце (например, столбце A, начиная с ячейки A2), используя заполнение ячеек арифметической прогрессией с начальным значением $t_0 = 0$ и шагом 0,001.

Задание 3. Задав значения параметрам $a = a_i > 0$ и $b = b_i > 0$,

рядом со столбцом t_k (например, в столбце В, начиная с ячейки В1) получить столбец значений для $Y_m(t_k, a_i, b_i)$. Этот столбец будет имитировать наблюдения y_k . Сохранить в каких-либо ячейках значения параметров a_i и b_i , которые будем считать «истинными» параметрами динамической системы.

Задание 4. Задав другие, отличные от «истинных», значения параметрам $a > 0$ и $b > 0$, рядом со столбцом y_k (например, в столбце С, начиная с ячейки С1) получить столбец значений для $Y_m(t_k, a, b)$. Этот столбец будет имитировать сигнал модели. В какой-нибудь ячейке поместим формулу расчета суммы квадратов невязок $I(a, b)$, воспользовавшись встроенной функцией Excel.

Задание 5. Воспользовавшись построителем диаграмм, постройте в одних осях графики наблюдаемого сигнала и модельного процессов.

Задание 6. Установите параметр b модели равным «истинному» значению $b = b_i$. Изменяя параметр модели a в пределах $a_i \pm 0.1 a_i$, проследите за изменениями $I(a, b)$.

Задание 7. Установите параметр a модели равным «истинному» значению $a = a_i$. Изменяя параметр модели b в пределах $b_i \pm 0.1 b_i$, проследите за изменениями $I(a, b)$.

Задание 8. Сделайте вывод о чувствительности $I(a, b)$ к рас-
согласования параметров модели и реального объекта.

ЛАБОРАТОРНАЯ РАБОТА №4 СУБД MS Access

Перед началом работы убедимся, что в рабочем каталоге имеется файл *Ведомость.txt*. Если этот файл отсутствует, то его необходимо занести туда, обратившись к преподавателю.

Откройте файл *Ведомость.txt* с помощью MS Word, включив режим отображения символов форматирования. Посмотрите на табличную структуру, записанную в этом файле. Обратите внимание, какими символами разделены поля в строках.

ВНИМАНИЕ. После просмотра в MS Word выйдете из редактора без сохранения файла.

Запустите MS Access и создайте новую базу, снабдив ее оригинальным именем.

Импорт данных

Активируем режим **Файл > Внешние данные > Импорт**

В открывшемся диалоговом окне находим файл *Ведомость.txt* и открываем его. После этого начинает работать Мастер импорта данных. Следите за работой Мастера, нажимая все время кнопку *Далее* Импорт завершится появлением в базе данных таблицы **Ведомость**.

Работа с таблицей

1. Выделите записи, в которых в поле *Математика* оценка более 3.
2. Выделите записи, в которых во всех полях оценки более 3.

Работа с запросами

В режиме Конструктор создайте следующие запросы на выборку:

1. «Должники», выводящий названия групп и шифры студентов, у которых имеется хотя бы одна двойка.
2. «Должники2», выводящий названия групп и шифры студентов, у которых имеется более одной двойки.
3. Создайте запрос на выборку с вычисляемым полем *Средний*, в котором вычисляется средний балл каждого студента по всем предметам.

4. На основе данных запроса «Инд_средний_балл» создайте запрос с групповой операцией «Средний_балл_группа» и выполнив группировку по полю *Группа* и групповую операцию Avg с полем *Средний* определите средний балл в каждой группе. Поле со значениями среднего балла по группе назовите *СГ*

5. На основе данных запроса «Средний_балл_группа» создайте запрос с групповой операцией «Общий_средний» и выполнив групповую операцию Avg с полем *СГ* определите средний балл по всем группам

6. На основе данных, выданных запросами «Инд_балл» и «Средний_балл» создать новый запрос «Выше_среднего», выделяющий названия групп и шифры тех студентов, у которых индивидуальный средний балл превышает средний балл группы.

7. Построить *перекрестный* запрос «Оценки», формирующий выходную информацию *по одному из предметов* в форме следующей таблицы:

Группа	Отл	Хор	Удовл	Неуд	Н А

где в столбце «Группа» содержится название группы, в остальных — количество соответствующих оценок у студентов данной группы. В столбце Н_А указано количество не аттестованных студентов.

Работа с отчетом

1. Сформировать отчет на основе данных запроса «Оценки», создав в нем заголовок «Итоги сессии по предмету». В разделе *Примечания* разместить итоговую строку с надписью «Всего» под столбцом *Группа*, а под остальными столбцами — суммарное по всем группам количество оценок соответствующего вида.

2. Осуществить экспорт полученного отчета в MS Word.

ЛАБОРАТОРНАЯ РАБОТА № 5
Mathcad: техника использования шаблонов
при вводе формул

Задание на лабораторную работу

Запустить Mathcad и настроить интерфейс так, чтобы были видны панели *Calculator* и *Graph* :

1. Вычислить значения выражений:

$$1) \frac{2^{10} - 5^{3.2} \cdot e^{-5.1} \cdot \ln(3^{2.5} - 1)}{2 \cdot \sin(5\pi - 1.2)}$$

$$2) \frac{X^{\sin(Z+1)} + X^{\cos(Z)}}{X^{4.1} + 2}$$

для $X = 5(7.8 + \sqrt{75})$ $Z = 2.1 \cdot \pi + \frac{1}{\pi}$

2. Решить уравнение $X^2 \cdot \ln(X) - e^{-X} = 0$

а) табулированием функции на интервале $[0.1; 2]$ с шагом 0.05;

б) графически;

в) численно с помощью функции **root**.

3. Задан полином:

$$p(t) = t^5 - 15 \cdot t^4 + 85 \cdot t^3 - 225 \cdot t^2 + 274 \cdot t - 120$$

а) построить график $p(t)$ на интервале $[0, 8; 5, 2]$

б) построить в тех же осях графики

$$0.2 \cdot \frac{d}{dt} p(t) \text{ и } 0.05 \cdot \frac{d^2}{dt^2} p(t)$$

в) найти корни полинома, используя функцию $p(t)\text{coeffs}, t$ из меню *Symbolic* и функцию *polyroots*;

г) с помощью оператора символического вывода получите выражения для $\frac{d}{dt} p(t)$ и $\frac{d^2}{dt^2} p(t)$;

д) по аналогии с пунктом «в)» найти корни полиномов $\frac{d}{dt} p(t)$ и $\frac{d^2}{dt^2} p(t)$;

ЛАБОРАТОРНАЯ РАБОТА № 6

Матричные операции Mathcad

Задание на лабораторную работу

1) Ввести строчную переменную **S**, инициализировав ее какой-нибудь фразой, например, "у лукоморья дуб зеленый".

2) Определите фактическую длину введенной строки **n** с помощью функции **strlen**. Функция возвращает число символов в строке аргументе.

3) С помощью функции **str2vec** получите вектор **V**, состоящий из ASCII-кодов этой строки. Размер вектора будет совпадать с длиной строки **n**.

4) Используя функции **identity**, **submatrix** и **stack** построить матрицу циклической перестановки, размером **n×n**.

Определение. Матрицей циклической перестановки называется квадратная матрица **T**, размера **n×n**, следующего вида:

$$T = \begin{pmatrix} 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & 0 & \dots & 0 & 0 \end{pmatrix}$$

Действие такой матрицы на вектор **V** размера **n** сводится к циклической перестановке его компонент.

5) Выполнить преобразование **TV** и осуществить преобразование результата в строку с помощью функции **vec2str**. Убедиться, что произошла циклическая перестановка букв в исходной фразе.

6) Показать, что $T^n = E$, где **E** — единичная матрица, Убедиться, что $T^T = T^{-1}$, где T^T — транспонированная матрица, а T^{-1} — обратная матрица.

Указание. Формирование матрицы \mathbf{T} должно выполняться только на основе информации извлекаемой из строки \mathbf{S} . Изменение содержимого

7) Используя покомпонентное формирование матрицы, построить матрицу инверсной перестановки \mathbf{IN}

Определение. Матрицей инверсной перестановки называется квадратная матрица \mathbf{IN} , размера $\mathbf{n} \times \mathbf{n}$, следующего вида:

$$\mathbf{IN} = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & 0 & \dots & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & \dots & 0 & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 & 0 \end{pmatrix}$$

Действие такой матрицы на вектор \mathbf{V} размера \mathbf{n} сводится к перестановке его компонент в обратном порядке.

8) Выполнить преобразование \mathbf{INV} и осуществить преобразование результата в строку с помощью функции `vec2str`. Убедиться, что произошло «переворачивание» исходной фразы.

9) Показать, что $\mathbf{IN}^2 = \mathbf{E}$, где \mathbf{E} — единичная матрица, Убедиться, что $\mathbf{IN}^T = \mathbf{IN}$, где \mathbf{IN}^T — транспонированная матрица, и а $\mathbf{IN}^{-1} = \mathbf{IN}$.

Указание. Формирование матрицы \mathbf{IN} должно также выполняться только на основе информации извлекаемой из строки \mathbf{S} .

ЛАБОРАТОРНАЯ РАБОТА № 7

Интерполяция и регрессия выборочных данных

Имеется выборка экспериментальных данных, которые представляются в виде массива, состоящего из пар чисел (x_i, y_i) . Возникает задача аппроксимации дискретной зависимости $y(x_i)$ непрерывной функцией $f(x)$. Функция $f(x)$, в зависимости от специфики задачи, может отвечать различным требованиям:

- $f(x)$ должна проходить через точки (x_i, y_i) , т.е. $f(x_i) = y_i, i=1..n$. В этом случае говорят об *интерполяции* данных функцией $f(x)$ во внутренних точках между x_i или *экстраполяцией* за пределами интервала, содержащего все x_i ;
- $f(x)$ должна некоторым образом (например, в виде определенной аналитической зависимости) приближать $y(x_i)$, выполнение условия $f(x_i) = y_i$ не обязательно. Такова постановка задачи *регрессии*, которую во многих случаях также можно назвать сглаживанием данных.

В составе Mathcad имеется встроенные функции, позволяющие осуществлять различную регрессию, интерполяцию и экстраполяцию.

Интерполяция. Для построения интерполяции-экстраполяции в Mathcad имеется несколько встроенных функций, позволяющих "соединить" выборочные точки (x_i, y_i) кривой разной степени гладкости. Точки (x_i, y_i) называют *узлами интерполяции*. По определению интерполяция означает построение функции $A(x)$, аппроксимирующей зависимость $y(x)$ в промежуточных точках (между x_i). Поэтому интерполяцию еще по-другому называют аппроксимацией. В точках x_i значения интерполяционной функции должны совпадать с исходными данными, т.е. $A(x_i) = y(x_i)$.

Линейная интерполяция. Самый простой вид интерполяции — линейная, которая представляет искомую зависимость $A(x)$ в виде ломаной линии. Интерполирующая функция $A(x)$ состоит из отрезков прямых, соединяющих точки.

Для построения линейной интерполяции служит встроенная функция **linterp**:

- **linterp**(x,y,t) — функция, аппроксимирующая данные векторов x и y кусочно-линейной зависимостью:

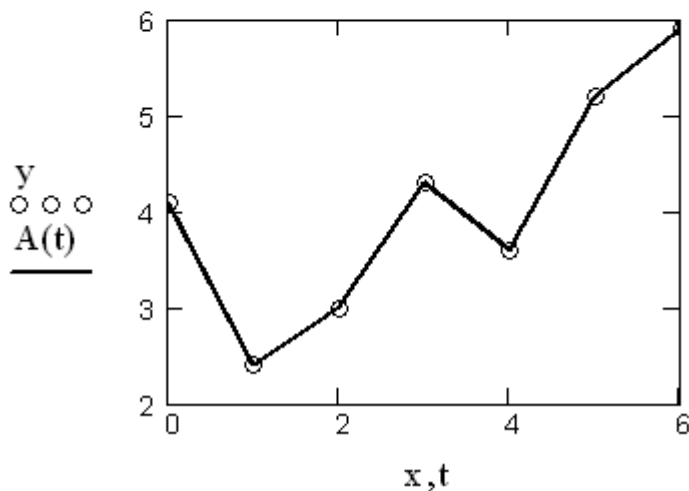
- x — вектор действительных данных аргумента;
- y — вектор действительных данных значений того же размера;
- t — значение аргумента, при котором вычисляется интерполирующая функция.

ВНИМАНИЕ! Элементы вектора x должны быть упорядочены в порядке возрастания, т. е. $X_1 < X_2 < X_3 < \dots < X_N$.

Листинг 1. Пример линейной интерполяции

```
x := (0.0 1.0 2.0 3.0 4.0 5.0 6.0)T
y := (4.1 2.4 3.0 4.3 3.6 5.2 5.9)T
A(t) := linterp(x, y, t)

A(0.5) = 3.25  A(3.75) = 3.775
A(-1) = 5.8  A(7.5) = 6.95
```



Как видно из листинга, чтобы осуществить линейную интерполяцию, надо выполнить следующие действия:

- ввести векторы данных x и y (первые две строки листинга);
- определить функцию **linterp** (x, y, t).

Далее можно вычислять значение интерполирующей функции в любой точке как внутри интервала данных, так и за его пределами, можно построить график интерполирующей функции. На графике Листинга 1 интерполируемые значения изображены кружками, а интерполирующая функция — сплошной линией.

ПРИМЕЧАНИЕ. Обратите внимание, что функция $A(t)$ на графике имеет аргумент t , а не x . Это означает, что функция $A(t)$ вычисляется не только при значениях аргумента x (т. е. в семи точках), а при гораздо большем числе аргументов в интервале $(0,6)$, что автоматически обеспечивает *Mathcad*.

Линейная интерполяция дает непрерывную функцию $A(t)$, но ее первая производная имеет разрывы в узлах интерполяции. Для некоторых приложений это не приемлемо, требуются более гладкие функции. В составе *Mathcad* имеются средства интерполяции отрезками кубических полиномов — *кубическими сплайнами*.

Интерполяция кубическими сплайнами. Для интерполяции кубическими сплайнами используются функции **cspline** или **pspline** или **lspline**, рассчитывающие коэффициенты кубических полиномов, и функция **interp**, непосредственно рассчитывающая значения интерполирующей функции.

Применение интерполяции кубическими сплайнами иллюстрирует *Листинг 2*. Интерполируемые данные (x_i, y_i) здесь те же самые, что и в примере *Листинга 1*.

На графиках *Листинга 2* в одних осях показаны интерполирующие функции **Ac(t)**, **Ap(t)** и **Al(t)**, полученные соответственно с помощью функций **cspline**, **pspline** и **lspline**.

Функции **cspline**, **pspline** и **lspline** различаются характером полиномов за пределами интервала данных, т.е. экстраполирующими полиномами. Функция **cspline** использует кубический экстраполирующий полином, **pspline** — параболический, а **lspline** — линейную функцию. Поскольку интерполирующий и экстраполирующие полиномы гладко «сшиты» между собой, то интерполирующие кривые, построенные на основе этих функции различаются только вблизи граничных узлов (см. график *Листинга 2*).

На *Листинге 2* приведены значения в интерполированных точках, расположенных внутри интервала выборочных данных и экстраполированные точки. Значения в интерполированных точках, полученных с помощью **cspline**, **pspline** и **lspline**, близки между собой, в экстраполированных точках различие более существенно.

Листинг 2. Интерполяция кубическими сплайнами

$$x := (0.0 \ 1.0 \ 2.0 \ 3.0 \ 4.0 \ 5.0 \ 6.0)^T$$

$$y := (4.1 \ 2.4 \ 3.0 \ 4.3 \ 3.6 \ 5.2 \ 5.9)^T$$

$$vc := cspline(x, y) \quad vp := pspline(x, y) \quad vl := lspline(x, y)$$

$$Ac(t) := \text{interp}(vc, x, y, t)$$

$$Ap(t) := \text{interp}(vp, x, y, t)$$

$$Al(t) := \text{interp}(vl, x, y, t)$$

Интерполированные значения:

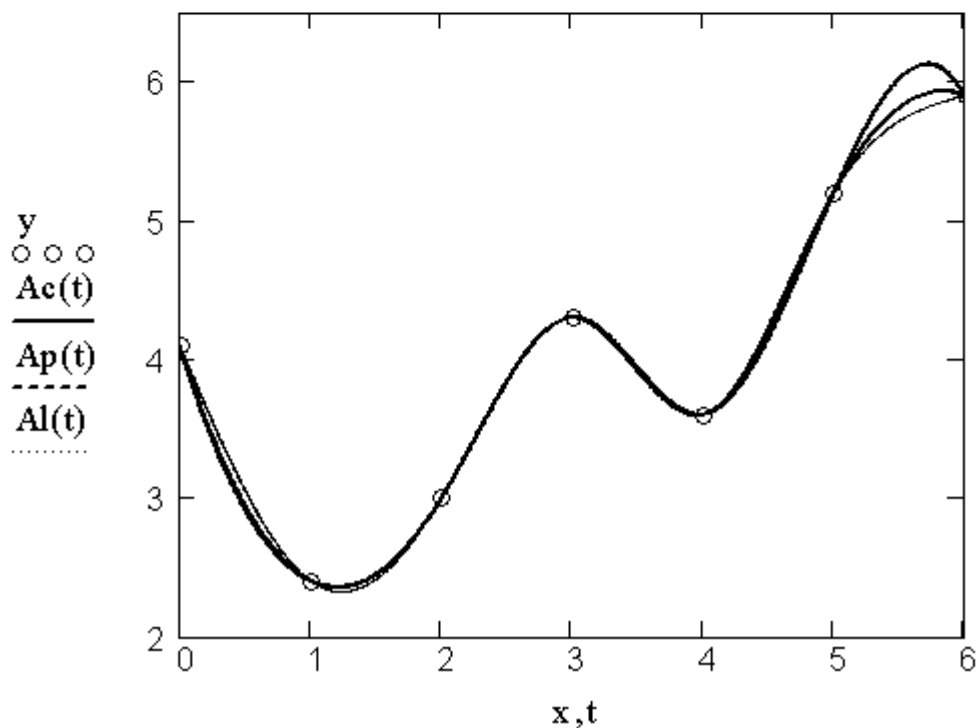
$$Ac(0.5) = 2.921 \quad Ap(0.5) = 2.945 \quad Al(0.5) = 3.057$$

$$Ac(3.75) = 3.691 \quad Ap(3.75) = 3.677 \quad Al(3.75) = 3.669$$

Экстраполированные значения:

$$Ac(-1) = 8.771 \quad Ap(-1) = 8.237 \quad Al(-1) = 5.8$$

$$Ac(7.5) = -7.269 \quad Ap(7.5) = 2.994 \quad Al(7.5) = 7.785$$



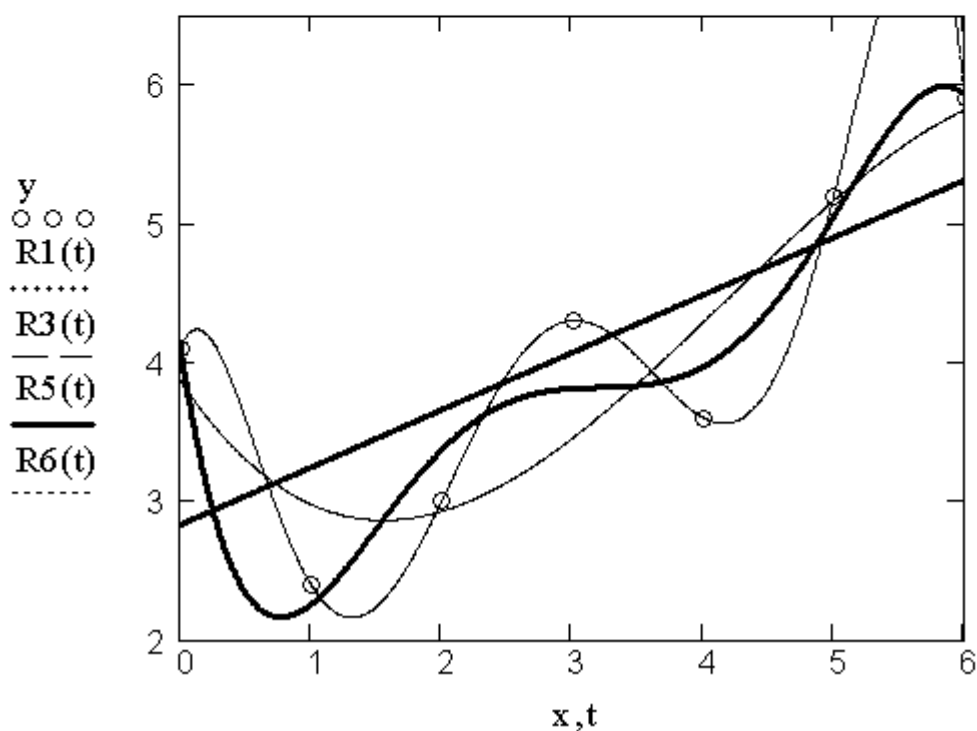
Полиномиальная регрессия. Алгоритмически полиномиальная регрессия на Mathcad строится подобно интерполяции сплайнами. Отличие состоит лишь в том, что вместо функций **cspline**, **pspline** и **lspline** их роль отведена функции **regress**. Функция **regress** принимает три параметра, первые два аналогичны по смыслу и назначению параметрам **cspline**, **pspline** и **lspline**. Третий параметр — целочисленный указывает степень полинома, который будет аппроксимировать выборочные данные.

Листинг 3 Полиномиальная регрессия

```

x := (0.0 1.0 2.0 3.0 4.0 5.0 6.0)T
y := (4.1 2.4 3.0 4.3 3.6 5.2 5.9)T
v1 := regress(x, y, 1) v5 := regress(x, y, 5)
v3 := regress(x, y, 3) v6 := regress(x, y, 6)
R1(t) := interp(v1, x, y, t) регрессия полиномом 1-й степени
R3(t) := interp(v3, x, y, t) регрессия полиномом 3-й степени
R5(t) := interp(v5, x, y, t) регрессия полиномом 5-й степени
R6(t) := interp(v6, x, y, t) регрессия полиномом 5-й степени

```



Полином 1-го порядка — прямая линия, его график проходит с минимальным среднеквадратическим удалением от выборочных точек.

Полином 6-го порядка имеет 7 параметров — коэффициентов, поэтому его график точно прошел через все выборочные точки, регрессия превратилась в интерполяцию.

Качество регрессии можно оценить величиной среднеквадратической ошибки аппроксимации в выборочных точках. Величины таких ошибок приведены на *Листинге 4*.

Листинг 4. Среднеквадратические ошибки регрессии.

$$\sqrt{\frac{1}{\text{rows}(x)} \cdot \sum_{k=0}^{\text{rows}(x)-1} (y_k - R1(x_k))^2} = 0.759$$

$$\sqrt{\frac{1}{\text{rows}(x)} \cdot \sum_{k=0}^{\text{rows}(x)-1} (y_k - R3(x_k))^2} = 0.477$$

$$\sqrt{\frac{1}{\text{rows}(x)} \cdot \sum_{k=0}^{\text{rows}(x)-1} (y_k - R5(x_k))^2} = 0.281$$

$$\sqrt{\frac{1}{\text{rows}(x)} \cdot \sum_{k=0}^{\text{rows}(x)-1} (y_k - R6(x_k))^2} = 2.561 \times 10^{-12}$$

Здесь функция **rows** используется для определения фактической размерности массива **x**.

Как следует из приведенных выше результатов, «точность» регрессии растет с увеличением порядка аппроксимирующего полинома. В том случае, когда число коэффициентов полинома (оно на единицу больше его степени) сравняется с размером выборки график полинома начинает проходить через все выборочные точки, ошибка становится нулевой (до ошибок округления). Полином регрессии превращается в интерполяционный полином, дальнейшее наращивание степени бесполезно.

С повышением порядка полинома регрессии ухудшаются «сглаживающие» свойства аппроксимации, кривая графика начинает осциллировать, стремясь «достать» каждую выборочную точку. Это обстоятельство негативно сказывается на результате, если в выборке имеется шумовая компонента.

Задание на лабораторную работу

1. Путем поэлементного формирования создать набор «экспериментальных» данных из двух векторов X и Y по n строк в каждом. Рекомендуется выбрать нечетное n из интервала (10;20). Заполнить вектор X целыми числами от 1 до n — это будет выборка значений независимой переменной. Сформировать выборку значений зависимой переменной Y , заполнив все элементы нулями и только один центральный отсчет положить равным 1. Если выбрать n нечетным, как было рекомендовано выше, то ненулевому отсчету будет соответствовать строка со значением независимой переменной $(n+1)/2$.

2. Воспроизвести выборочные точки на графике, используя тип графика **points** с символом **o's**.

3. Выполнить линейную интерполяцию функцией **linterp** и воспроизвести интерполирующую кривую на том же графике, что и выборочные точки.

4. Выполнить интерполяцию кубическими сплайнами с использованием функцией **lspline** **pspline** и **cspline** воспроизвести интерполирующие кривые на одном графике. На том же графике отобразить и выборочные точки.

5. Оценить экстраполирующие свойства аппроксимирующих функций, построенных в п. 4. Для этого следует продолжить их графики за пределы выборочных данных.

6. На одном графике отобразить интерполирующую функцию, первую, вторую и третью ее производную. Обратит внимание на вид графиков производных.

7. Анализируя графики производных аппроксимирующих функций за пределами интервала выборочных значений X , установить порядок экстраполирующего полинома используемого функциями **lspline** **pspline** и **cspline**.

ЛАБОРАТОРНАЯ РАБОТА № 8

Ввод вещественного числа и управляющих символов на C++

Цель работы. Получение навыков управления программой с помощью ввода символов с клавиатуры.

Тематика изучаемого материала: операторы ветвления вычислительного процесса, циклы и логические выражения.

Задание на лабораторную работу

Составить программу, осуществляющую обработку потока символов ASCII с клавиатуры с разделением их на три категории:

- 1) информационные,
- 2) управляющие,
- 3) неинформационные.

Нужно организовать разную реакцию программы на ввод символов различных категорий.

Информационные символы, должны в порядке поступления накапливаться в символьном массиве-буфере, откуда в определенный момент направляются на дальнейшую обработку в виде сплошной символьной строки.

Управляющие символы (4—5 заданных символов ASCII) используются для ветвления вычислительного процесса. В частности, ввод одного из них может инициировать обработку накопленной в буфере информации, ввод другого — вызывать завершение программы и т.д.

Неинформационные символы должны игнорироваться программой и не вызывать в ней никакой реакции.

Обработка символьной информации буфера сводится к преобразованию содержащейся в нем символьной строки в вещественное число с помощью библиотечной функции с именем **atof**. Описание прототипа этой функции содержится в заголовочном файле **math.h**, который следует подключить с помощью директивы препроцессору `#include <math.h>`. Прототип имеет вид:

double atof (const char*)

Эта функция принимает аргументом строку символов, соответствующую внешнему представлению числа и возвращает значение вещественного числа во внутреннем формате компьютера.

Например, запись:

```
double d = atof("-12.678");
```

приведет к инициализации переменной `d` значением `-12.678`

Таким образом, к информационным символам, пропускаемым в буфер, следует отнести только те символы, которые образуют внешнее представление вещественного числа. Это коды символов минус (-), точка (.) и коды десятичных цифр (от 0 до 9). Прочие символы не должны заноситься в буфер.

При вводе необходим логический контроль последовательности заносимых в буфер информационных символов. Так, символ минус может появиться в буфере один раз, занимая в нем только лидирующее положение, символ точки тоже может заноситься в буфер только однажды.

После обработки символьной последовательности буфера, он должен быть очищен и готов к приему следующей последовательности. Перед отправкой содержимого буфера на обработку его необходимо дополнить т.н. "нуль-терминатором" — нулевым байтом.

Вывод результата обработки должен производиться в формате, соответствующем типу результата, т.е. как вещественного числа.

Указания:

- Выполнить программу в виде "вечного цикла" с возможностью его завершения вводом управляющего символа (например, Esc).
- Ввод символов с клавиатуры производить без автоматической выдачи эхо — печати на монитор, для чего воспользоваться функцией ввода символа `getch()`, подключив заголовочный файл **conio.h**. Эхо — печать следует предусмотреть только для информационных символов.
- Для фильтрации ASCII-символов воспользоваться операциями отношения. В тексте программы вместо численного значения ASCII-кодов символов использовать их представление в виде символьных констант.

ЛАБОРАТОРНАЯ РАБОТА № 9

Интерфейс ввода матриц на C++

Цель работы. Получение навыков программирования интерфейса ввода с разделением управляющих и информационных символов.

Тематика изучаемого материала: операторы управления и логические выражения.

Задание на лабораторную работу

Составить программу, осуществляющую ввод двумерной числовой матрицы с клавиатуры с расположением элементов на экране в виде прямоугольной таблицы, состоящей из n строк и m столбцов (см. рис. 9.1.) Для ввода числовых значений элементов матрицы следует воспользоваться функцией из лабораторной работы №8

При этом размеры матрицы — числа n и m — задаются, исходя из размеров фактически введенной таблицы. Число столбцов в матрице m определяется фактическим числом элементов введенных в *первую строку матрицы*, а число строк n — по количеству фактически введенных строк.

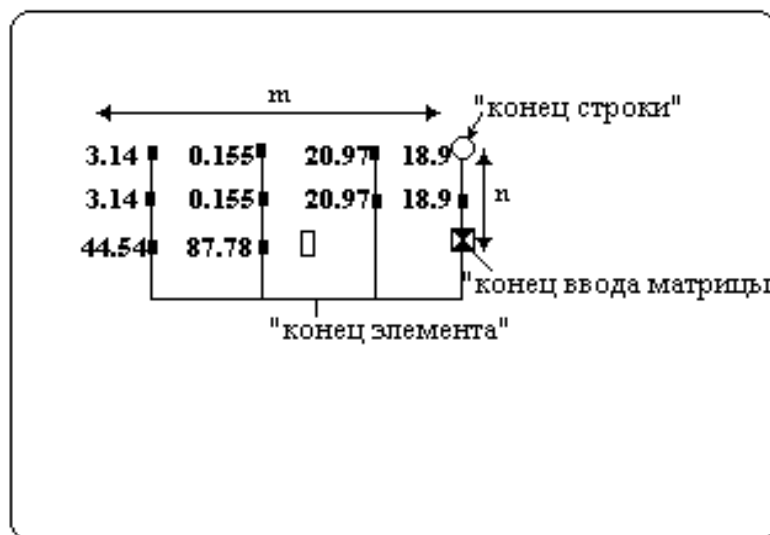


Рис. 9.1 — Вид поля ввода матрицы и расположение точек ввода управляющих символов

Программа должна работать в режиме ожидания событий — ввода управляющих символов с клавиатуры и обрабатывать следующие события:

- 1) ввод управляющего символа "**конец элемента**", в данном случае-последовательности информационных символов, образующих внешнее представление вещественного числа.
- 2) ввод управляющего символа "**конец строки**",
- 3) ввод управляющего символа "**конец ввода матрицы**",
- 4) ввод управляющего символа "**выход из программы**".

Обработку этих событий следует организовать по следующей логической схеме.

После ввода первой строки необходимо зафиксировать размер m матрицы.

При вводе второй и последующих строк, когда размер m уже зафиксирован, нужно блокировать реакцию на событие "**конец строки**" вплоть до заполнения строки требуемым количеством элементов.

На попытку ввода в последующие строки большего числа элементов, чем в первой строке требуется предусмотреть переход к вводу элемента следующей строки. Этот переход должен визуально отображаться перемещением курсора соответствующую позицию в поле ввода.

Значения вводимых элементов должны накапливаться в массиве вещественных чисел, и по завершении ввода всей матрицы (после ввода символа "**конец ввода матрицы**"), содержимое массива необходимо вывести на экран по формату вещественных чисел в виде матрицы. Этот контрольный вывод нужно воспроизвести на экране где-либо за пределами поля ввода, чтобы можно было сравнить вводимую матрицу с матрицей, образовавшейся в памяти. При правильной работе программы эти матрицы должны совпасть.

Указание. Позиционирование курсора на экране при вводе и выводе обеспечить с помощью функции `void gotoxy (int x,int y)`, прототип которой дан в `conio.h`, при этом необходимо учесть, что эта функция работает с координатами символьного экрана.

ЛАБОРАТОРНАЯ РАБОТА № 10

Объединения, битовые поля и поразрядные операции C++

Цель работы — получение навыков программирования поразрядных операций на C++, работы с объединениями и битовыми полями.

Тематика изучаемого материала: объединения и битовые поля, выражения с поразрядными операциями.

Задание на лабораторную работу.

I. Определить в программе тип структуры с битовыми полями **struct** ВУ, общим размером 32 бита (4 байта), размеры битовых полей выбрать таким образом, чтобы одно из них по положению совпало с расположением двоичного порядка во внутреннем представлении числа типа **float**. Формат внутреннего представления числа типа **float** приведен в Приложении А.

Назначая размеры элементов структуры с битовыми полями, необходимо учесть два фактора:

а) размер элемента не может превышать принятого в данной системе размера **int** или, что эквивалентно, **unsigned**.

б) элементы объекта типа структура — битовое поле располагаются в адресном пространстве по порядку их записи в описании структуры, *начиная с младших битов*.

II. Сформировать в программе три объекта типа **union** каждый с двумя полями одинаковой длины разного типа:

```
union {
long L;
unsigned long UL;
} LUL;
```

```
union {
float F;
unsigned long UL;
} FUL;
```

```
union {
```

```

float F;
BY Bit
} FBY;

```

III. Составить 2 подпрограммы-функции, принимающие по два аргумента:

для функции 1 первый типа **long**, второй — типа **char***.

для функции 2 первый типа **float**, второй — типа **char***.

В результате работы функций в массив, передаваемый вторым аргументом, помещается последовательность из 32 символов '0' и '1', отражающую состояние битов в первом аргументе. Для удобства использования можно сделать эти функции, возвращающими указатель на символьный массив второго аргумента, после его заполнения внутри функций.

Указание. Внутри функции 1 следует использовать объект LUL, внутри функции 2 — объект FUL, в обеих функциях поразрядные операции сдвига кода и поразрядную операцию «И»

Продемонстрируем на примере, как эта функция должна работать. Пусть эта функция имеет интерфейсы вида:

```
char* LtoBit(long T, char* buf)
```

```
.char* FtoBit(float T, char* buf)
```

тогда при выполнении фрагмента программы

```
char bf[33];
```

```
cout<< LtoBit(1, bf)<<endl;
```

```
cout<< FtoBit(1, bf)<<endl;
```

в выходной поток должно быть выведено:

```
0000000000000000000000000000000001
```

```
0011111110000000000000000000000000
```

Строки отражают состояния битов в 32-х разрядном слове, где содержится двоичный код 1 в формате типов **long** и **float** соответственно

IV. Составить программы двух функций осуществляющих вывод значений *порядка* и *мантиссы* числа типа **float**.

IV. Составить программу осуществляющую.

Ввод вещественных чисел с клавиатуры в переменную типа **float**.

Присвоение переменной типа **long** значение, хранимое в переменной типа **float**

Вывод на экран в битовом представлении введенных значений сохраненных как **long** и как **float**, воспользовавшись, соответственно, функциями `LtoBit` и `FtoBit`. В результате на экране должны появиться последовательности нулей и единиц соответствующие состоянию бит в кодах введенного числа.

Для вещественных типов вывести значения мантиссы и *порядка* числа во внутреннем представлении.

Указания

- Для реализации программы следует использовать объект типа **union** (объединение), и поразрядные операции языка C++.
- Учесть, что поразрядные операции в C++ применимы только к операндам целого типа (**int**, **unsigned** и т. п.).
- Получение значений порядка и мантиссы числа с плавающей точкой следует выполнить с использованием *битовых полей*.

ЛАБОРАТОРНАЯ РАБОТА № 11

Программирование односвязного списка

Цель работы. Получение навыков работы с указателями и динамическим распределением памяти.

Задание на лабораторную работу

Составить программу, реализующую универсальный *контейнер* в форме *односвязного списка*.

Определение 1: *Контейнером* называют программный объект, способный вмещать в себя другие объекты.

Определение 2: *Односвязным списком* называется схема организации множества элементов — данных, при которой с каждым элементом связан вспомогательный атрибут — ссылка на следующий элемент (см. рис. 7.1.)

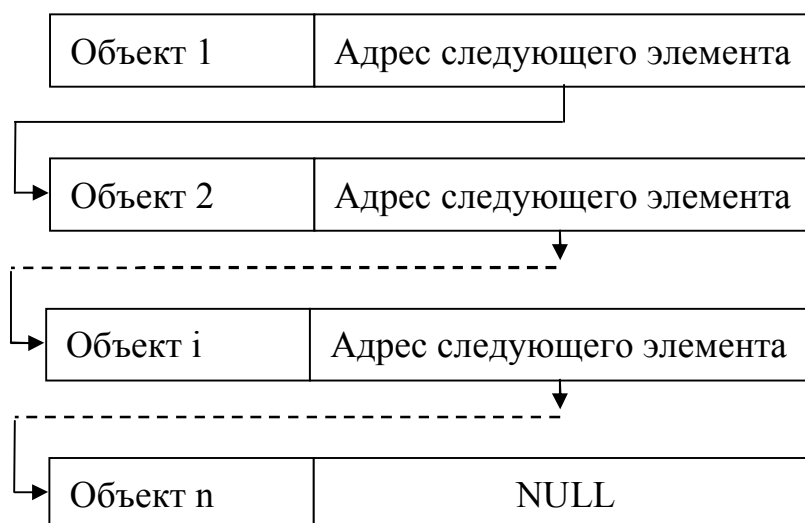


Рис. 7.1. Структура односвязного списка

Необходимо реализовать следующие функции:

- дополнение контейнера объектами (теоретически в неограниченном количестве);
- удаление объектов из контейнера;
- подсчет количества элементов в контейнере;
- доступ к элементам по их номеру (индексу)

При программировании следует использовать указатели C++ и операторы динамического распределения памяти **new** и **delete**.

Реализовать контейнер для хранения объектов типа **int**;

В качестве усложненного варианта предлагается создать контейнер для хранения строк произвольной длины.

ЛИТЕРАТУРА

1. Симонович С.В. Информатика. Базовый курс. 2-е изд.: Учебник для вузов. — СПб.: Питер, 2006. — 640 с.
2. Острейковский В.А. Информатика: Учебник для вузов. — М.: Высшая школа, 2001. — 512 с.
3. Н. В. Макарова [и др.] Информатика: Практикум по технологии работы на компьютере: Учебное пособие для вузов. — 3-е изд., перераб. — М.: Финансы и статистика, 2005. — 255 с.
4. Павловская Т.А. С/С++. Программирование на языке высокого уровня: Учебник для вузов. — СПб.: Питер, 2006. — 461 с.
5. Конев Ф.Б. Информатика для инженеров: Учеб. пособие. — М.: Высш. шк., 2004. — 272 с.
6. Карпов Б. Microsoft Office 2000: Справочник. — СПб: Изд-во «Питер», 2000. — 448 с.
7. Акулов О.А., Медведев Н.В. Информатика: базовый курс: Учеб. для студентов вузов. — 4-е изд. — Москва: Омега-Л, 2007. — 560 с.
8. Васильев А.Н. Mathcad 13 на примерах. — СПб.: БХВ-Петербург, 2006. — 528 с.
9. Очков В.Ф. Mathcad 12 для студентов и инженеров. — СПб.: БХВ-Петербург, 2005. — 464 с.
10. Кирьянов Д.В. Самоучитель Mathcad 13. — СПб.: БХВ-Петербург, 2006. — 528 с.
11. Степанов А.Н. Информатика: Учебник для вузов. — 5-е изд. — СПб: Изд-во «Питер», 2007. — 765 с.
12. Воройский Ф.С. Информатика. Новый систематизированный толковый словарь-справочник. — М.: ФИЗМАТЛИТ, 2003. — 760 с.
13. Пасько В. Энциклопедия ПК. Аппаратура. Программы. Интернет. — СПб.: Питер, 2004. — 800 с.
14. Волкотруб Л.П., Егоров И.М. Компьютер и здоровье. — Томск: Томск. гос. ун-т систем упр и радиоэлектроники, 2006. — 158 с.
15. Дьяконов В. Mathcad 2001: Учебный курс. — СПб.: Питер, 2001.