

А.О. Семкин, С.Н. Шарангович

ИНФОРМАТИКА

**Учебно-методическое пособие по практическим занятиям и
организации самостоятельной работы студентов
направления подготовки 11.03.02– «Инфокоммуникационные
технологии и системы связи»**

2015

Министерство образования и науки Российской Федерации
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ
И РАДИОЭЛЕКТРОНИКИ
(ТУСУР)

Кафедра сверхвысокочастотной и квантовой радиотехники
(СВЧиКР)

А.О. Семкин, С.Н. Шарангович

ИНФОРМАТИКА

Учебно-методическое пособие по практическим занятиям и
организации самостоятельной работы студентов направления
подготовки 11.03.02 «Инфокоммуникационные технологии и
системы связи»

УДК 004.4+519.6

Рецензент:
профессор каф..СВЧиКР,

А.Е. Мандель

А.О. Семкин, С.Н. Шарангович

Информатика: учебно-методическое пособие по практическим занятиям и организации самостоятельной работы для бакалавров направления подготовки 11.03.02 «Инфокоммуникационные технологии и системы связи» / А.О. Семкин, С.Н. Шарангович. – Томск: ТУСУР, 2015. – 40 с.

В данном пособии рассмотрены все виды самостоятельной работы при изучении дисциплины «Информатика».

Приводится программа курса, его цели и задачи. Каждый раздел программы заканчивается методическими указаниями со ссылкой на литературу. Представлены темы лабораторных занятий, а также темы практических занятий с примерами решения задач.

Методическое пособие предназначено для студентов очной, заочной, и вечерней форм обучения по направлению подготовки бакалавриата «Инфокоммуникационные технологии и системы связи». – 11.03.02.

УДК 004.4+519.6

© Томск. гос. ун-т систем упр. и
радиоэлектроники, 2015

© Семкин А.О., Шарангович С.Н., 2015

Оглавление

1. Цели и задачи дисциплины	6
2. Содержание лекционного курса	8
3. Лабораторные занятия	17
4. Практические занятия.....	18
5. Курсовая работа	24
6. Контрольные работы.....	25
7. Экзаменационные вопросы	29
8. Распределение самостоятельной работы.....	34
9. Методика текущего контроля освоения дисциплины.....	34
10. Учебно-методические материалы по дисциплине	38

Введение

Самостоятельная работа студентов является частью учебного процесса при подготовке квалифицированных специалистов, способных самостоятельно и творчески решать стоящие перед ними задачи. В ходе самостоятельной работы формируются важнейшие профессиональные навыки будущего специалиста, такие как: внутренняя готовность к самообразованию в профессиональной сфере, самостоятельность, инициативность и ответственность, умение работать с источниками информации.

Каждая дисциплина должна иметь методическое сопровождение по самостоятельному изучению разделов и тем, указанных в рабочей программе, по написанию рефератов, выполнению расчетно-графических и лабораторных работ. В связи с этим эффективная организация самостоятельной работы студентов требует проведения целого ряда мероприятий, создающих предпосылки и условия для реализации самостоятельной работы, а именно:

- обеспечение студентов информационными ресурсами (учебными пособиями, справочниками, банками индивидуальных заданий);
- обеспечение студентов методическими материалами (учебно-методическими практикумами, сборниками задач, указаниями по выполнению лабораторных работ);
- наличие материальных ресурсов (ПК, измерительного и технологического оборудования для выполнения заданий в рамках НИР и ГПО);
- организация консультаций преподавателей;
- возможность публичного обсуждения теоретических и практических результатов, полученных студентом самостоятельно при выполнении НИРС и ГПО (конференции, олимпиады, конкурсы).

Важным элементом в организации самостоятельной работы студентов является контроль. Контроль требует разработки преподавателем контролирующих материалов в текстовом или тестовом исполнении, а при использовании ПК – пакета прикладных программ для проверки знаний студентов. Эффективная система контроля (в т.ч. электронная система контроля), наряду с рейтинговой системой оценки знаний, позволит добиться систематической самостоятельной работы студентов над учебными материалами и повысить качество обучения.

Пособие разработано в соответствии с временными рекомендациями по организации самостоятельной работы студентов (письмо Минобразования РФ от 27.11.2002 "Об активизации самостоятельной работы студентов высших учебных заведений").

1. Цели и задачи дисциплины

1.1. Цель преподавания дисциплины.

Целью преподавания дисциплины «Информатика» является развитие навыков работы на персональных компьютерах в современных операционных системах для решения инженерных задач сбора, передачи, обработки и хранения информации.

Место дисциплины в учебном процессе. Дисциплина «Информатика» относится к федеральному компоненту цикла общих математических и естественно научных дисциплин и дает основу для последующего изучения таких специальных дисциплин, как «Сети связи и системы коммутации», «Системы и сети передачи дискретных сообщений», ряда разделов курсов «Оптические направляющие системы», «Оптические цифровые телекоммуникационные системы». Специальной подготовки для освоения данной дисциплины не требуется.

1.2. Задачи изучения дисциплины

Основными задачами изучения дисциплины являются: изучение принципов действия современных персональных компьютеров, основ алгоритмизации и технологии программирования научно-технических задач, языков программирования высокого уровня, технологии обработки и отладки программ, знакомство с современным программным обеспечением, методами решения типовых инженерных задач и их программной реализацией.

В результате изучения курса студенты должны:

Знать

- современное состояние уровня и направлений развития вычислительной техники и программных средств;
- возможности, принципы построения и правила использования наиболее распространенных пакетов прикладных программ общего назначения (текстовые и графические редакторы, электронные таблицы, системы управления базами данных) и компьютерных средств связи
- основные принципы организации записи хранения и чтения информации в ЭВМ.
- аппаратную реализацию ЭВМ.
- основы организации операционных систем Linux и Windows.
- основы организации и функционирования глобальных и локальных сетей ЭВМ.
- основные приемы алгоритмизации и программирования на языке высокого уровня C++;
- основные численные методы для решения вычислительных задач, наиболее часто встречающихся в инженерной практике.

Уметь

- работать с программными средствами общего назначения, соответствующими современным требованиям;
- пользоваться электронными таблицами или системами управления базами

данных;

- решить поставленную задачу, используя алгоритмический язык высокого уровня C++ и необходимое программное обеспечение (среды Qt и Qt Elipse Integration);
- использовать ЭВМ для решения функциональных и вычислительных задач, наиболее часто встречающихся в инженерной практике;
- пользоваться математическими пакетами Mathcad и MathLab;

Владеть

- технологией работы на ПЭВМ в операционных системах Linux и Windows;
- компьютерными методами сбора, хранения и обработки (редактирования) информации;
- приемами структурированного, объектно-ориентированного и обобщенного программирования на языке C++;
- методами математического моделирования процессов и явлений;
- приемами антивирусной защиты.

1.3. Перечень обеспечивающих дисциплин

Специальной подготовки для освоения данной дисциплины не требуется.

1.4 Объем дисциплины и виды учебной работы

Вид учебной деятельности (очная форма обучения)	1 семестр	2 семестр	Всего
Лекции	36	28	64
Лабораторные занятия	64	50	114
Практические занятия		14	14
Контроль самостоятельной работы (курсовой проект / курсовая работа)		10	10
Подготовка к экзамену	36	36	72
Самостоятельная работа	80	78	158
Всего	216	216	432
В зачетных единицах	6	6	12.0

Вид учебной деятельности (заочная форма обучения)	2 семестр	3 семестр	Всего
Лекции	8	6	14
Лабораторные занятия	12	12	24
Практические занятия	4	2	6
Контроль самостоятельной работы (курсовой проект / курсовая работа)		2	2
Подготовка к экзамену	9	9	18
Самостоятельная работа	183	185	368
Всего	216	216	432
В зачетных единицах	6	6	12.0

2. Содержание лекционного курса

2.1. Понятие информации. Общая характеристика процессов сбора, передачи, обработки и накопления информации – 2 часа

Введение. Объем, содержание и основные задачи дисциплины «Информатика». Сигналы, данные, методы. Понятие об информации. Свойства информации.

Данные. Носители данных. Операции с данными. Кодирование данных двоичным кодом. Кодирование чисел, текстовых, графических, звуковых и видео данных. Файлы и файловые структуры.

Методические указания. Усвоение этого раздела является обязательным для успешного изучения курса, однако современный уровень развития техники обуславливает знакомство студентов с понятием об информации с самого раннего возраста в рамках школьного образования. Таким образом, предполагается, что студент обладает начальными знаниями по тематике данного раздела. Для повторения и более глубокого изучения основ информатики рекомендуются учебные пособия [1,2].

2.2. Аппаратные реализации информационных процессов – 2 часа

Устройство персонального компьютера. Базовая аппаратная конфигурация. Внутренние устройства системного блока. Системы, расположенные на материнской плате. Периферийные устройства персонального компьютера.

Методические указания. Содержание данного раздела предполагает наличие у студентов базовых знаний об устройстве персонального компьютера. В качестве повторения рекомендуется обратиться к пособиям [1,2]. В разделе рассматриваются основные элементы персонального компьютера, их характеристики и взаимодействие друг с другом.

2.3. Алгоритмизация и программирование. Языки программирования высокого уровня – 2 часа

Общие принципы вычислений на ЭВМ. Языки программирования. Методика подготовки и решения задач на ЭВМ. Погрешности при вычислениях на ЭВМ. Влияние организации вычислительного процесса на точность. Алгоритмизация вычислений. Линейные, разветвляющиеся, циклические алгоритмы. Циклические и рекуррентные вычисления – общие схемы вычислений, условия завершения циклов по заданной точности вычислений. Алгоритмы со структурой вложенных циклов.

Методические указания. Данный раздел является основополагающим для всего дальнейшего процесса изучения языков и принципов программирования. Глубокое понимание основ позволит без затруднений разобраться в дальнейших тонкостях предмета. В связи с этим рекомендуется уделить особое внимание самостоятельному изучению тематики данного раздела. В качестве пособий можно порекомендовать [1-3, 5, 6].

2.4. Программирование задач выбора и сортировки – 1 час

Критерии выбора, примеры программирования задач выбора минимального, максимального элемента, поиска экстремума функции. Сортировки в одномерных массивах данных. Методы ранжирования числовых последовательностей.

Методические указания. Программирование простейших задач позволит получить представление о процессе программирования как таковом и об общих подходах к решению математических задач. Литература [3, 5-7].

2.5. Машинное преобразование матриц – 2 часа

Операции с многомерными массивами данных – перестановка строк, столбцов, транспонирование матриц. Преобразование квадратной матрицы к треугольному виду. Вычисление определителей. Вычисление обратной матрицы. Вычисление собственных значений матриц.

Методические рекомендации. Для понимания процесса автоматизации обработки матриц необходимы начальные знания соответствующего математического аппарата. Рекомендуется повторить соответствующий материал дисциплины «Математика». Литература [3, 5].

2.6. Решение систем линейных алгебраических уравнений – 1 час

Обзор способов решения систем линейных уравнений с помощью ЭВМ. Метод Крамера, метод Гаусса, итерационные методы. Сравнение методов. Алгоритм метода Гаусса с выбором главного элемента.

Методические указания. Для понимания процесса автоматизации обработки матриц необходимы начальные знания соответствующего математического аппарата. Рекомендуется повторить соответствующий материал дисциплины «Математика». Литература [3, 5].

2.7. Численное решение нелинейных уравнений – 1 час

Общая характеристика задачи – классификация уравнений, методы и этапы их решения. Задачи отделения корней и уточнение их значений. Метод дихотомии для решения задач отделения и уточнения корней. Исследование поведения функции в интервале отделения корня. Методы хорд, касательных, метод итераций, комбинированный метод. Сравнение возможностей различных методов уточнения корня.

Методические указания. Для понимания процесса автоматизации решения нелинейных уравнений необходимы начальные знания соответствующего математического аппарата, этапов и методов решения. Рекомендуется повторить соответствующий материал дисциплины «Математика». Литература [3, 5].

2.8. Численные методы интегрирования - 1 час

Методы прямоугольников, трапеций и Симпсона. Метод Гаусса. Оценки точности вычисления интегралов и их связь с задачей интерполяции функции. Адаптивный алгоритм.

Методические указания. Для понимания процесса автоматизации интегрального исчисления необходимы начальные знания соответствующего математического аппарата. Рекомендуется повторить соответствующий материал дисциплины «Математика». Литература [3, 5].

2.9. Решение обыкновенных дифференциальных уравнений – 1 час

Типы задач для обыкновенных дифференциальных уравнений. Методы Эйлера и Рунге-Кутты в задачах Коши. Сравнение методов.

Методические указания. Для понимания процесса автоматизации дифференциального исчисления необходимы начальные знания соответствующего математического аппарата. Рекомендуется повторить соответствующий материал дисциплины «Математика». Литература [3, 5].

2.10. Методы обработки экспериментальных данных – 1 час

Интерполяция и аппроксимация функций. Интерполяция каноническим полиномом. Сплайновая интерполяция. Метод наименьших квадратов при построении аппроксимирующей функции.

Методические указания. Для понимания процесса автоматизации обработки экспериментальных данных необходимы начальные знания соответствующего математического аппарата, теории функций и их областей определения и значения. Рекомендуется повторить соответствующий материал дисциплины «Математика». Литература [3, 5].

2.11. Начальные сведения о языке C++ – 1 час

История возникновения C++. Преимущества C++. Комментарии в языке C++. Знакомство с препроцессором, заголовочными файлами, пространством имен. Функция main(). Объекты cin и cout. Символ новой строки. Операторы объявления и переменные. Оператор присваивания. Функции. Функции с возвращаемым значением. Функции без возвращаемого значения. Прототип функции. Формат определения функции. Заголовок функции.

Методические указания. Данный раздел является вводным в изучение языка программирования высокого уровня C++. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [5-7].

2.12. Базовые типы данных – 1 час

Переменные. Имена переменных. Целочисленные типы данных (short, int и long). Целочисленные константы. Определение типа константы в C++. Тип данных char. Константы типа char. Тип данных wchar_t. Функция-элемент: cout.put(). Типы данных без знака. Тип данных bool. Квалификатор const. Числа с плавающей точкой. Типы данных с плавающей точкой. Константы с плавающей точкой. Арифметические операции языка C++. Приоритет операций и ассоциативность. Разновидности операции деления. Операция деления по модулю. Преобразования типов данных (в выражениях и при присваивании). Приведение типов.

Методические указания. Данный раздел является вводным в изучение

языка программирования высокого уровня C++. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [5-7].

2.13. Производные типы данных – 1 час

Массивы. Строки. Строчно-ориентированный ввод: `getline()` и `get()`. Структуры. Массивы структур. Структуры и разрядные поля. Объединения. Перечисления.

Методические указания. Данный раздел является вводным в изучение языка программирования высокого уровня C++. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [5-7].

2.14. Указатели и динамическая память – 1 час

Указатели. Объявление и инициализация указателей. Оператор `new`. Оператор `delete`. Динамические переменные. Динамические массивы. Указатели, массивы и арифметика указателей. Указатели и строки. Динамические структуры. Автоматическая, статическая и свободная память.

Методические указания. Данный раздел является вводным в изучение языка программирования высокого уровня C++. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [5-7].

2.15. Циклы и выражения сравнения – 1 час

Цикл `for`. Операторы инкремента (`++`) и декремента (`--`). Комбинированные операторы присваивания. Составные операторы или блоки. Оператор "запятая" (или дополнительные синтаксические приемы). Выражения сравнения. Сравнение строк. Цикл `while`. Псевдонимы типов. Цикл `do while`. Вложенные циклы и двумерные массивы.

Методические указания. Данный раздел является вводным в изучение языка программирования высокого уровня C++. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [5-7].

2.16. Операторы ветвления и логические операции – 1 час

Оператор `if`. Оператор `if else`. Конструкция `if else if else`. Логические выражения (логическое ИЛИ: `||`, логическое И: `&&`, логическое НЕ: `!`) Приоритет логических операций. Библиотека символьных функций `cctype`. Оператор `?:`. Оператор `switch`. Операторы `break` и `continue`.

Методические указания. Данный раздел является вводным в изучение языка программирования высокого уровня C++. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [5-7].

2.17. Функции языка C++ – 4 часа

Определение функции. Прототипирование и вызов функций. Аргументы функции и передача по значению. Функции с несколькими аргументами. Функции и массивы. Массивы и указатели. Массивы в качестве аргументов. Отображение массива и его защита с помощью спецификатора `const`. Указатели и спецификатор `const`. Функции и строки в стиле C. Функции, возвращающие

строки. Функции и структуры. Передача адресов структур. Рекурсия. Указатели на функции. Получение адреса функции. Объявление указателя на функцию. Использование указателя для вызова функции.

Встроенные функции. Ссылочные переменные. Ссылки в роли параметров функции. Временные переменные, ссылочные аргументы и модификатор `const`. Использование ссылок при работе со структурами. Аргументы, заданные по умолчанию. Полиморфизм функций (перегрузка функции). Шаблоны функций. Перегруженные шаблоны функций. Явная специализация. Раздельная компиляция.

Методические указания. Данный раздел является основным для понимания технологий структурированного программирования и создает основу для дальнейшего изучения объектной модели и объектно-ориентированного программирования. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [5-7].

2.18. Классы памяти, диапазоны доступа и связывание – 2 часа

Автоматическая память. Автоматические переменные и работа со стеком. Переменные типа `register`. Статический класс памяти. Внешние переменные. Модификатор `static` (локальные переменные). Связывание и внешние переменные. Спецификаторы классов памяти: `const`, `volatile` и `mutable`. Классы памяти и функции. Языковое связывание. Классы памяти и динамическое распределение. Пространства имен.

Методические указания. Данный раздел является основным для понимания технологий объектно-ориентированного программирования. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [5-7].

2.19. Объекты и классы – 2 часа

Концепция класса. Определение и реализация класса. Общедоступный и приватный доступ к классу. Элементы данных- класса. Методы класса (функции-элементы класса). Создание и использование объектов класса. Конструкторы и деструкторы класса. Функции-элементы `const`. Указатель `this`. Создание массивов объектов. Диапазон доступа к классу. Абстрактные типы данных (ADT).

Методические указания. Данный раздел является основным для понимания технологий объектно-ориентированного программирования. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [5-7].

2.20. Работа с классами – 2 часов

Перегрузка операторов. Дружественные функции. Перегрузка операции «, используемой при выводе данных. Элементы состояния. Использование функции `rand()` для генерации случайных чисел. Автоматическое преобразование и приведение типов для классов. Функции преобразования классов.

Методические указания. Данный раздел является основным для понимания технологий объектно-ориентированного программирования. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [5-7].

2.21. Классы и динамическое распределение памяти – 2 часа

Применение метода динамического распределения памяти для элементов класса. Неявные и явные конструкторы копирования. Неявные и явные перегруженные операторы присвоения. Последовательность действий при использовании конструктора new. Использование элементов класса static. Использование указателей на объекты. Реализация ADT типа "очередь".

Методические указания. Данный раздел является основным для понимания технологий объектно-ориентированного программирования. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [5-7].

2.22. Наследование классов – 2 часа

Наследование в качестве отношения is-a. Общедоступное наследование классов. Защищенный доступ. Списки инициализатора конструктора. Приведение вверх и приведение вниз. Виртуальные функции-элементы. Раннее (статическое) и позднее (динамическое) связывание. Полностью виртуальные функции. Работа с методом общедоступного наследования.

Методические указания. Данный раздел является основным для понимания технологий объектно-ориентированного программирования. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [5-7].

2.23. Повторное использование программного кода в C++ – 2 часа

Отношения has-a. Классы с объектами-элементами (включение). Частное и защищенное наследование. Создание шаблонов классов. Использование шаблонов классов. Специализации шаблонов. Множественное наследование. Виртуальные базовые классы.

Методические указания. Данный раздел является основным для понимания технологий объектно-ориентированного программирования. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [5-7].

2.24. Дружественные конструкции и исключения – 2 часа

Дружественные классы. Методы дружественных классов. Вложенные классы. Генерирование исключений, блоки try и catch. Классы исключений. Библиотека RTTI. Операторы dynamic_cast и typeid. Операторы static_cast, const_cast и reinterpret_cast.

Методические указания. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [5-7].

2.25. Класс string и стандартная библиотека шаблонов – 2 часа

Стандартный класс языка C++ string. Шаблон auto_ptr. Стандартная библиотека шаблонов (STL). Классы-контейнеры. Итераторы. Объекты-функции (функторы). Алгоритмы STL.

Методические указания. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [5-7].

2.26. Ввод/вывод данных и работа с файлами – 2 часа

Ввод и вывод данных с точки зрения C++. емейство классов ostream. Перенаправление. Методы класса ostream. Форматирование результатов вывода. Методы класса istream. Состояния потока. Файловый ввод/вывод. Использование класса ifstream для ввода файлов. Использование класса ofstream для вывода файлов. Использование класса fstream для одновременного ввода/вывода файлов. Обработка параметров командной строки. Двоичные файлы. Произвольный доступ к файлу. Внутреннее форматирование.

Методические указания. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [5-7].

2.27. Графический пользовательский интерфейс. Среда разработки Qt – 1 час

Понятие графического пользовательского интерфейса (GUI). Назначение, структура, компоненты, типы. Общие сведения о возможностях создания GUI в Qt Creator.

Методические указания. Освоение материала данного раздела и последующих даст общее представление о разработке приложений с графическим пользовательским интерфейсом. Знания, полученные в рамках изучения предыдущих разделов являются основополагающими, последующие разделы позволят изучить технологию создания программных средств более глубоко. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [8,9].

2.28. Библиотека Qt. Виджеты. Компоновка Виджетов – 1 час

Понятие виджета (Window gadget). Типы виджетов, назначение. Центральный виджет. Компоновка и взаимодействие виджетов. Менеджеры компоновки виджетов.

Методические указания. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [8,9].

2.29. Библиотека Qt. Взаимодействие виджетов. Механизм сигналов и слотов – 2 часа

Взаимодействие виджетов. Обмен информацией, обработка событий. Механизм сигналов и слотов. Синтаксис, правила применения. Макрос Q_OBJECT. Создание пользовательских виджетов. Подходы, реализация. Функции-обработчики событий.

Методические указания. Для самостоятельного изучения и подготовки к

аудиторным занятиям рекомендуются пособия [8,9].

2.30. Библиотека Qt. Создание диалоговых и главных окон программ – 1 час

Диалоговые и главные окна программы. Структура, компоненты графического интерфейса. Класс QAction. Меню программы. Панель инструментов. Контекстное меню. Строка состояния

Методические указания. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [8,9].

2.31. Библиотека Qt. Возможности разработки сетевых приложений – 1 час

Документация Qt. Шаблоны проектов для Qt Creator. Реализация протокола TCP в библиотеке Qt. Пример сетевого приложения обмена текстовыми сообщениями по локальной сети.

Методические указания. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [8,9].

2.32. Библиотека Qt. Возможности разработки приложений для мобильных устройств – 2 часа

Кроссплатформенные среды Qt Eclipse Integration и Qt Necessitas. Операционная система Android. Особенности программирования. Примеры проектов приложений для ОС Android на C++.

Методические указания. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [8-10].

2.33. Математические пакеты Mathcad и MatLab – 2 часа

Назначение пакетов MathCAD и MatLab. Основные приемы программирования и визуализации расчетов в пакетах MathCAD и MatLab. Математические пакеты Mathcad и MatLab в задачах вычислительной математики.

Методические указания. Использование данных пакетов позволяет автоматизировать процесс математических вычислений, создавать математические модели физических процессов и формировать их графическое представление. Освоение материала данного раздела позволит сократить время на решение учебных задач последующих дисциплин и сконцентрироваться на их физическом смысле. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [1,2].

2.34. Прикладное программное обеспечение общего назначения – 4 часа

Основные пакеты прикладных программ. Офисные приложения. Текстовые редакторы и редакторы электронных таблиц. Компьютерная графика. Растровая и векторная графика. Графические редакторы.

Методические указания. Использование современного прикладного

программного обеспечения позволяет автоматизировать процесс оформления графических и текстовых документов, формируемых не только в процессе обучения студента, но и в весь период его последующей трудовой деятельности. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [1,2].

2.35. Введение в архитектуру вычислительных систем и операционные системы – 2 часа

Операционные системы персональных компьютеров (семейства ОС Windows и ОС Unix и Linux). Функции операционных систем. Обеспечение интерфейса пользователя. Организация и обслуживание файловой системы. Обеспечение взаимодействия с аппаратным обеспечением. Обслуживание компьютера.

Методические указания. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [1,2].

2.36. Базы данных и системы управления базами данных – 2 часа

Общее понятие о базах данных. Основные понятия систем управления базами данных и банками знаний

Методические указания. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [1,2].

2.37. Локальные и глобальные сети ЭВМ – 2 часа

Компоненты вычислительных сетей. Принципы построения сетей. Компьютерные сети и технологии интернет. Локальные сети. Основные сведения. Термины локальной сети. Работа в сети: предоставление ресурсов в сеть, подключение к ресурсам сети, управление сетью. Глобальные сети. Интернет. Имена в интернет. Выход в интернет. Типы сервиса в интернет: FTP, WWW. Поиск информации в интернет. Сервисы общения в Интернет. Электронная почта. Работа с почтовыми программами. Новости. Электронные конференции и доски объявлений. Представление информации в WWW. Гипертекст. Язык описания гипертекста HTML, редакторы html.

Методические указания. Изучение основ построения локальных и глобальных сетей в рамках данного курса позволит студентам быстрее освоить материал последующих дисциплин и даст представление об области дальнейшей профессиональной деятельности. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [1,2].

2.38. Компьютерные вирусы – 2 часа

Классификация и характеристика компьютерных вирусов . Пути распространения вирусов. Профилактика заражения. Работа с антивирусными программами, лечение зараженных файлов.

Методические указания. Для самостоятельного изучения и подготовки к аудиторным занятиям рекомендуются пособия [1,2].

3. Лабораторные занятия

Основными целями выполнения лабораторных работ являются:

- изучение технологии проектирования и разработки программных средств для решения прикладных задач;
- приобретение студентами навыков структурированного, объектно-ориентированного программирования в современных средах, а также навыков разработки графического пользовательского интерфейса;
- углубленное освоение студентами теоретических положений изучаемой дисциплины.

При выполнении лабораторных работ студент должен продемонстрировать знание соответствующего теоретического материала и знакомство с учебно-методической литературой по заданной теме.

Методические указания к лабораторным работам размещены на научно-образовательном портале ТУСУРа, а также представлены в локальной вычислительной сети кафедры СВЧиКР.

Список лабораторных работ:

№ п/п	№ раздела дисциплины	Наименование лабораторных работ	Трудо-емкость (час.)	ОК, ПК
1.1	3.1.	Начальные сведения о языке C++	2	ОК-1, ОК-9, ПК-2, ПК-14
1.2	3.2.	Базовые типы данных.	2	
1.3	3.3.	Производные типы данных	2	
1.4	3.4.	Указатели и свободная память	2	
1.5	3.5.	Циклы и выражения сравнения	4	
1.6	3.6.	Операторы ветвления и логические операции	4	
1.7	2.1., 3.7.	Функции языка C++. Программирование задач выбора и сортировки	4	
1.8				
1.9	2.3., 3.7.	Работа с функциями. Численное решение систем линейных алгебраических уравнений.	4	
1.10				
1.11	2.4., 3.7.	Работа с функциями. Численное решение нелинейных уравнений	4	
1.12	2.5., 3.7.	Работа с функциями. Численные методы интегрирования .	4	
1.13	2.6., 3.7.	Работа с функциями. Решение обыкновенных дифференциальных уравнений	4	
1.14	3.7., 2.7.	Работа с функциями. Методы обработки экспериментальных данных.	4	
1.15	3.8.	Классы памяти, диапазоны доступа и связывание	4	
1.16				
2.1	3.9.	Объекты и классы	4	
2.2	3.10.	Работа с классами	4	
2.3	3.11.	Классы и динамическое распределение памяти	4	
2.4	3.12.	Наследование классов	4	
2.5	3.13.	Повторное использование программного кода в	4	

		C++		
2.6	3.14.	Дружественные конструкции, исключения	4	
2.7	3.15.	Класс string и стандартная библиотека шаблонов.	4	
2.8	3.16.	Ввод/вывод данных и работа с файлами	4	
2.9	3.17. – 3.19.	Создание GUI в Qt Creator. Механизм сигналов и слотов	4	
2.10	3.20.	Библиотека Qt. Создание диалоговых окон программы	4	
2.11	3.20.	Библиотека Qt. Создание главного окна программы	4	
2.12	3.21.	Библиотека Qt. Разработка сетевых приложений	4	
2.13	3.22.	Библиотека Qt. Разработка приложений для ОС Android	4	
2.14	3.23.	Математический пакет MathCAD	4	
2.15	3.23.	Математический пакет Matlab	4	
2.16	4.2.	Создание базы данных в СУБД Access	4	
2.17	4.3.	Изучение Internet. Язык описания гипертекста HTML.	6	

Итого 114

На каждой лабораторной работе выделено 20 минут для теоретического теста (контрольной работы) по теме лабораторной работы.

4. Практические занятия

Основными задачами практических занятий по дисциплине «Информатика» являются:

- развитие навыков проектирования и моделирования разрабатываемых программных средств;
- развитие навыков работы с основными прикладными программами, освоение которых требуется для выполнения студенческих работ (курсовых, квалификационных и т.д.) на протяжении всего срока обучения.

Ниже, в п. 4.2 приведены примеры заданий для самостоятельного выполнения студентами на практических занятиях. В качестве пособий для решения поставленных задач можно порекомендовать [1, 2, 13, 14].

4.1 Темы практических занятий

№ п/п	№ раздела дисциплины	Тематика практических занятий	Трудо-емкость (час.)	ОК, ПК
1	1.3	Построение блок-схем алгоритмов решения прикладных задач	2	ОК-9, ПК-1
2	1.3	Универсальный язык моделирования UML. Построение диаграмм классов, диаграмм последовательностей, диаграмм взаимодействий	4	ОК-9, ПК-1

3	3.24	Векторная графика. Основы работы в редакторе MS Visio	2	ПК-2
4	3.24	Текстовые редакторы. Основы работы в редакторе MS Word	2	ПК-2
5	3.24	Редакторы электронных таблиц и баз данных. Основы работы в редакторе MS Excel	2	ПК-2
6	3.24	Создание электронных презентаций. Основы работы в MS PowerPoint	2	ОК-9, ПК-2

4.2. Примеры заданий для самостоятельного выполнения на практическом занятии

4.2.1. ТЕМА: Построение блок-схем алгоритмов решения прикладных задач

Задача №1

Постройте блок-схему алгоритма программы, которая считывает с клавиатуры число $n > 0$ и проверяет условие $n < 5$. Если условие не выполняется, на экран выводится n . Если условие выполняется, число n складывается с самим собой $5 - n$ раз. На экран выводится результат сложения. Например, если введено число $n = 2$, то для невыполнения условия $n < 5$, его нужно сложить с самим собой 3 раза ($5-2=3$). На экран при этом выводится «8».

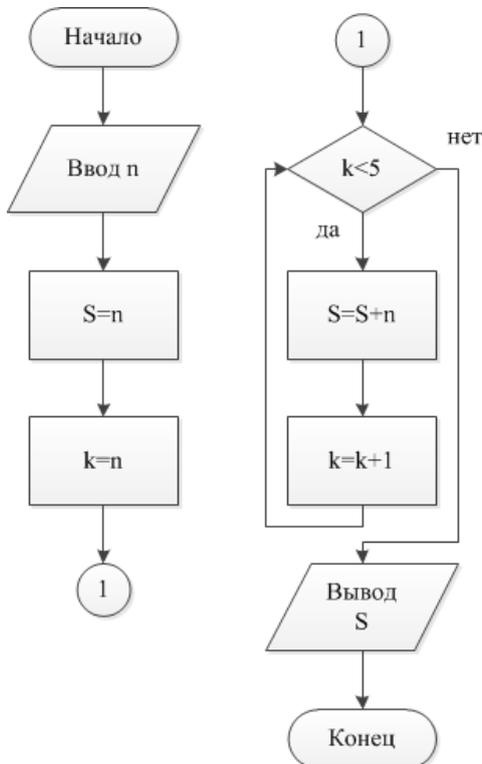


Рис. 1

Решение: Блок-схема описанного алгоритма приведена на рис. 1. После ввода числа $n > 0$, оно копируется в две переменные S, k . Одна (S) будет хранить конечный выводимый результат (сумму), другая (k) – будет определять количество процедур сложения числа с самим собой. Далее проводится проверка выполнения условия $k < 5$ (не $n < 5$) и, если условие не выполняется, на экран выводится S без дополнительных вычислений. Если условие выполняется, то S увеличивается на n , k увеличивается на единицу и снова выполняется проверка условия $k < 5$. Процедуры повторяются до тех пор, пока выполняется условие $k < 5$. Затем S выводится на экран.

Программную реализацию данного алгоритма на выбранном языке программирования студент выполняет самостоятельно.

4.2.2. ТЕМА: Универсальный язык моделирования UML. Построение диаграмм классов, диаграмм последовательностей, диаграмм взаимодействий

Задача №1

Постройте диаграмму наследования для следующих классов: «утюг», «электроплита», «газовая плита», «микроволновая печь», «печь», «стиральная машина», «кухонное оборудование» «бытовая техника», «комбайн».

Решение: Анализируя представленный перечень классов, можно выделить класс «бытовая техника» как класс с наиболее высоким уровнем абстракции, значит, данный класс является базовым для всех представленных в перечне. Производными от класса «бытовая техника» будут «утюг» и «стиральная машина», а «печь» и «комбайн» будут производными от класса «кухонное оборудование». Оставшиеся три класса «электроплита», «газовая плита», «микроволновая печь» являются производными классами от «печь» и наиболее конкретно определяют тип бытовой техники, а значит будут размещаться в нижней части «дерева» наследников. Таким образом, используя элементы языка UML, можно построить следующую диаграмму (рис. 2).

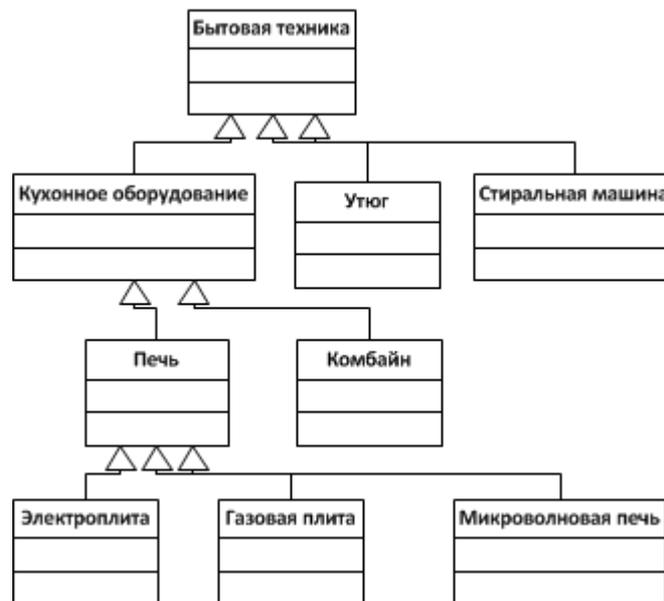


Рис. 2

Программную реализацию данной диаграммы на выбранном языке программирования студент выполняет самостоятельно.

Задача №2

Постройте диаграмму классов для задачи автоматизации процесса продажи какого либо товара. Используйте следующие классы: «продавец», «покупатель», «товар», «сделка», «платежная квитанция».

Решение: Анализируя представленный перечень классов и процедуру покупки какого-либо товара, определим отношения между указанными классами. Во-первых, из данного перечня нельзя выделить отношения типа

«обобщение» (наследование). Во-вторых, ни один из представленных классов не является составной частью другого. Таким образом, наиболее подходящими типами взаимоотношений между ними являются «ассоциации» (взаимодействия). Определим эти взаимодействия. Центральным элементом рассматриваемой задачи является «сделка», поэтому можно предположить, что все остальные классы будут взаимодействовать именно со «сделкой». «Покупатель» может совершить несколько покупок, поэтому между ним и «сделкой» возможна связь типа «один-ко-многим». Аналогично и «продавец» может совершать несколько «сделок». Со «сделкой» может быть связан список «продуктов», но «сделка» и «платежная квитанция» состоят в отношении «один-к-одному».

Таким образом, используя элементы языка UML, можно построить следующую диаграмму (рис. 3).

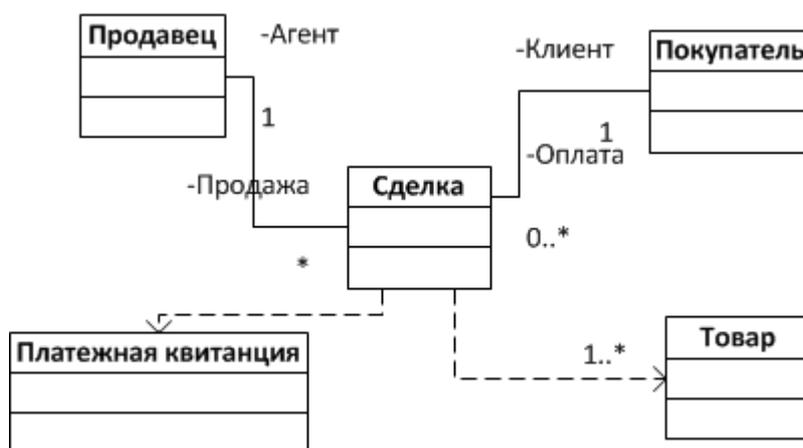


Рис. 3

Программную реализацию данной диаграммы на выбранном языке программирования студент выполняет самостоятельно.

4.2.3. ТЕМА: Векторная графика. Основы работы в редакторе MS Visio

Примеры заданий для самостоятельной работы на практических занятиях

Задача №1

Используя встроенные средства MS Visio для построения блок-схем, составьте блок-схему из задачи №1 темы 4.2.1.

Задача №2

Используя встроенные средства MS Visio для моделирования UML, составьте диаграмму классов из задачи №1 темы 4.2.2.

Задача №3

Используя встроенные средства MS Visio для моделирования UML, составьте диаграмму классов из задачи №2 темы 4.2.3.

4.2.4. ТЕМА: Текстовые редакторы. Основы работы в редакторе MS Word

Примеры заданий для самостоятельной работы на практических занятиях

Задание №1

- Создайте на диске X:\ новый файл-документ Microsoft Office Word и назовите его Практика_информатика Фамилия (с Вашей фамилией в конце).
- Создайте таблицу из трех строк. В первой строке сделайте два столбца, во второй – четыре, в третьей – восемь. Сделайте так, чтобы границы третьей строки были штриховой линией.
- С помощью инструмента WordArt напишите по центру страницы ниже таблицы надпись ТУСУР, надпись синего цвета.
- Создайте многоуровневый список: раздел первого уровня – ТУСУР, разделы второго уровня – специальности, количество – не менее четырех, разделы третьего уровня – группы, не менее шести в сумме.
- С помощью инструмента Microsoft Equation добавьте в документ следующие формулы:

- $$\mathbf{E}_j = A_j \cdot \mathbf{e}_j \cdot \exp [i(\omega \cdot t - \mathbf{k}_j \cdot \mathbf{r})];$$

- $$\mathbf{e}_j = \frac{\mathbf{e}_j^1 + i \cdot \rho_j \cdot \mathbf{e}_j^2}{\sqrt{1 + \rho_j^2}};$$

- $$\xi_e(\mathbf{r}) = \left[\frac{K_{33} \cdot 8\pi}{\varepsilon_e - \varepsilon_o} \cdot \frac{1}{[E^o(\mathbf{r})]^2 + [E^e(\mathbf{r})]^2} \right]^{1/2};$$

- $$R_j = \begin{bmatrix} \cos(\gamma_j) & -\sin(\gamma_j) \\ \sin(\gamma_j) & \cos(\gamma_j) \end{bmatrix}.$$

- Нарисуйте с помощью инструмента «Фигуры» двумерную систему координат, все четыре квадранта, подпишите оси x и y, в первом квадранте системы координат нарисуйте прямоугольник, в третьем – круг.
- Добавьте в нижний колонтитул документа нумерацию страниц и свою фамилию, имя, отчество.
- Сделайте рамку, обрамляющую страницы документа.
- Скопируйте файл с выполненным заданием в папку с названием Практика информатика на диске S:\СемкинАО.

4.2.5. ТЕМА: Редакторы электронных таблиц и баз данных. Основы работы в редакторе MS Excel

Примеры заданий для самостоятельной работы на практических занятиях

Задание №1

Пусть фирма ведет учет выручки по четырем округам города в летние месяцы (июнь, июль, август). Исходные данные – 12 чисел. Рассчитать в Excel сумму по каждому округу, сумму всего по округам и процентное соотношение суммы выручки в каждом округе.

Округа\месяцы	Июнь	Июль	Август	Сумма по округу	В процентах
Центральный	140	160	120	=сумма чисел левее	=сумма всего по округам (*) / число левее * 100
Западный	85	80	100	=сумма чисел левее	=сумма всего по округам (*) / число левее * 100
Северный	120	135	140	=сумма чисел левее	=сумма всего по округам (*) / число левее * 100
Южный	110	115	105	=сумма чисел левее	=сумма всего по округам (*) / число левее * 100
Всего по округам	= сумма чисел выше (*)	100%			

Задание №2

Составьте таблицу значений линейной функции $y=kx+b$, выбрав по своему усмотрению коэффициент k и свободный член b .

x	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
y													

4.2.6. ТЕМА: Создание электронных презентаций. Основы работы в MS PowerPoint

Данное практическое занятие предполагает подготовку студентом презентации в MS PowerPoint на заданную тему.

Возможные темы презентаций:

- история города Томска;
- история ТУСУРа;
- принцип работы волоконно-оптических линий связи;
- интерференция и дифракция электромагнитных волн;
- типы волоконно-оптических кабелей.

5. Курсовая работа

Курсовая работа по информатике является первой во время обучения студентов в вузе. Поэтому от студентов при оформлении пояснительной записки требуется освоение основных правил оформления текстовых документов (в соответствии с требованиями ОС ТУСУР 6.1.2013).

Задания на курсовую работу носят индивидуальный характер. Как правило, они содержат элементы алгоритмизации, программирования и графического представления информации. В процессе выполнения курсовой работы закрепляются и углубляются навыки программирования, приобретается опыт использования возможностей вычислительной техники и программного обеспечения.

Задание на курсовую работу выдается либо в начале весеннего семестра, либо в конце ноября – начале декабря в осеннем семестре. В последнем случае - это производится для студентов, успешно справляющихся с программой лабораторного практикума или перевыполняющих ее.

5.1. Цель и задачи курсовой работы

Целью курсовой работы по дисциплине «Информатика» является развитие и закрепление навыков структурированного, объектно-ориентированного и обобщенного программирования. В процессе самостоятельной работы студентом (под руководством преподавателя) решаются следующие задачи:

- составление последовательности организационных мер и приемов при выполнении работ большого объема;
- изучение вопросов теории, необходимых для выполнения задания;
- наработка навыков составления алгоритмов решения задач и проектирования программного средства;
- развитие навыков использования справочной литературы и поиска необходимых данных (включая источники в Интернет);

За проводимые в работе решения и правильность использования формул или математических моделей, за правильность расчета основных параметров и характеристик, за качество подготовки текстовых документов, а также за своевременность подготовки и защиты курсовой работы в целом отвечает студент. В процессе выполнения работы студент должен правильно организовать свой труд, регулярно работать над заданием, проявлять максимум инициативы и самостоятельности для решения поставленных задач.

5.2. Тематика курсовых работ

Тематика курсовых работ определяется программой курса, связана последующими дисциплинами и включает в себя следующие темы:

- Вычисление коэффициента ошибок цифровой системы передачи
- Вычисление модового распределения оптического волокна

- Вычисление дифракционного светового поля
- Решения дисперсионного уравнения многомодового волокна
- Вычисление энергетического бюджета цифровой ВОСП
- Статистическая обработка экспериментальных данных
- Вычисление определителей методом Гаусса
- Сплайн-аппроксимация, интерполяция и экстраполяция
- Решение дифференциальных уравнений методом Эйлера
- Решение систем нелинейных уравнений
- Операции с матрицами
- Программирование интерфейса COM порта
- Программирование интерфейса LPT порта
- Программирование интерфейса порта USB
- База данных на основе СУБД MS Access
- База данных на основе СУБД MySQL
- Создание Web-сайта
- Создание сетевых приложений
- Создание приложений для мобильных устройств.

5.3. Содержание курсовой работы

В результате выполнения курсовой работы студент должен получить представление обо всех этапах решения поставленной задачи: от проектирования (разработка блок-схем и UML-моделей) до написания исходного кода программы, его компиляции и анализа работы созданного приложения.

После получения задания по курсовой работе студент последовательно выполняет следующие этапы, входящие в рейтинговую систему оценки:

- анализ технического задания;
- составление алгоритма решения задачи;
- проектирование разрабатываемого программного средства;
- написание исходного кода вычислительного модуля программы;
- разработка графического пользовательского интерфейса (в случае, если это предусмотрено заданием);
- компиляция исходного кода программы;
- анализ работы приложения;
- оформление курсовой работы;
- защита работы.

6. Контрольные работы

В процессе изучения дисциплины «Информатика» студенты выполняют **две контрольные работы**.

Самостоятельное выполнение контрольных работ способствует более глубокому и конструктивному усвоению теоретического материала курса и

развитию необходимых инженеру навыков применения теории для практических задач.

Контрольная работа выполняется по индивидуальным билетам, содержащим три теоретических вопроса и одно практическое задание.

Ответы на теоретические вопросы даются в письменной форме, практическое задание выполняется на компьютере с последующей проверкой преподавателем.

6.1. Контрольная работа №1

Примеры заданий

Билет №1

1. Дайте определение понятиям «программа», «программирование», «язык программирования». Что такое «алгоритм»?
2. Что такое заголовочные файлы или файлы включения. Для чего предназначены файлы включения или заголовочные файлы?
3. Опишите правила инициализации массива в языке C++.
4. Задача: написать программу на языке C++, которая последовательно выводит на экран коды ASCII пяти символов, введенных с клавиатуры. (считать с клавиатуры пять символов, затем последовательно вывести на экран код ASCII для каждого введенного символа).

Билет №2

1. Дайте определение компилятору и интерпретатору исходных кодов программ.
2. Назовите основные типы целочисленных данных в языке C++. Почему в языке C++ несколько типов целочисленных данных?
3. Операторы объявления и присвоения. Что выполняет каждый из них?
4. Задача: написать программу на языке C++, которая вычисляет произведение пяти элементов массива. Массив заполнить данными с клавиатуры. (считать с клавиатуры пять чисел, записать их в массив, вывести на экран произведение всех пяти элементов).

Билет №3

1. Область имен. Назначение и возможности. Что делает следующий оператор:
`using namespace std;`

2. В каком виде информация хранится в памяти компьютера? Назовите единицы измерения информации.
3. С какого номера начинается индексация в массивах? Как соотносятся индекс последнего элемента и размер массива?
4. Задача: написать программу на языке C++, которая анализирует два введенных с клавиатуры числа. Если сумма чисел больше 10, на экран выводится надпись «Summ is more than 10». Если сумма меньше 10, на экран выводится надпись «Summ is less than 10». Если числа равны, то выводится надпись «Numbers are equal». Все три условия должны быть реализованы одним условным оператором. (используя логические операции и служебное слово else).

Билет №4

1. Что в языке C++ называется оператором? Опишите правила записи переменных в языке C++.
2. Числа с плавающей точкой. Запись чисел с плавающей точкой. Назовите типы данных с плавающей точкой.
3. Что такое заголовочные файлы или файлы включения. Для чего предназначены файлы включения или заголовочные файлы?
4. Задача: написать программу на языке C++, которая считывает с клавиатуры десять чисел с плавающей запятой, формирует из них массив, сортирует его по убыванию и выводит на экран отсортированный массив.

Билет №5

1. Арифметические операции в языке C++, запись, приоритеты. Разновидности операций деления для разных типов данных. Операция деления по модулю.
2. Опишите правила форматирования исходного кода в языке C++.
3. Дайте определение производному (пользовательскому) типу данных. Дайте определение типу массив.
4. Задача: написать программу на языке C++, которая последовательно выводит на экран строку заданное количество раз. Строку и количество раз вывода на экран считать с клавиатуры.

6.2. Контрольная работа №2

Примеры заданий

Билет №1

1. Дайте определение понятию «производный тип данных». Приведите примеры.
2. Что такое структура? В чем принципиальное отличие структуры от массива?
3. Что содержится в заголовочных файлах «.h» и файлах исходных текстов «.cpp»? Каким образом можно использовать два этих типа файлов?
4. Задача: Предположим, что `treacle` является массивом из 10 переменных типа `float`. Определите указатель, который указывает на первый элемент массива `treacle`, и используйте указатель для вывода первого и последнего элементов массива.

Билет №2

1. Дайте определение типу «указатель». Для хранения какой именно информации предназначен данный тип? Приведите примеры.
2. Опишите правила создания функции (процедуры). Приведите пример: напишите листинг функции, возвращающей произвольную целую степень дробного числа.
3. Чем в C++ строка “S” отличается от символа ‘S’?
4. Задача: Создайте программу, которая просит пользователя вводить положительные целые числа, а затем создает динамический массив из этих значений типа `int`. После этого, увеличивает размер массива вдвое и дублирует введенные значения.

Билет №3

1. Каким образом, используя синтаксис C++, можно узнать адрес области памяти, выделенной для целочисленной переменной `var`? Как называется данная операция?
2. Дайте определение производному типу данных «строка». В чем особенность обработки строк в C++?
3. Приведите пример обращения к полям структуры. Используйте структуру, описывающую работника какой-либо компании. Она должна содержать информацию о его имени, возрасте, семейном положении (состоит в браке или нет), рост (в метрах).

4. Задача: Напишите программу, которая динамически определяет структуру описывающую рыбу (структура должна включать вид рыбы, вес в целых граммах и длину в долях метра), а затем считывает с клавиатуры значения полей для двух разных рыб.

Билет №4

1. Опишите операторы (функции) строчно-ориентированного ввода в C++.
2. Приведите правила записи прототипа функции. Дайте пояснения, что означает каждая составляющая прототипа.
3. Опишите, каким образом можно обрабатывать массивы данных, используя «указатели».
4. Задача: Напишите программу, которая сначала спрашивает имя и фамилию пользователя, затем выводит на экран общее количество введенных букв.

Билет №5

1. Какие операции определены для «указателей»?
2. Функции `cin.getline()` и `cin.get()`. Назначение и отличия.
3. В чем преимущества использования функций в программах C++?
4. Задача: Опишите функцию, которая возвращает произвольную целую степень целого числа, при этом определяет, является ли число четным или нечетным, а также находит сумму цифр числа (для числа из двух разрядов: $10 < x < 99$).

7. Экзаменационные вопросы

Подготовка к экзаменам содействует систематизации, обобщению и закреплению знаний, устранению пробелов, возникающих в процессе учебных занятий, и должна вестись в течение всего семестра. Организация самостоятельной работы в семестре является залогом успешной сдачи зачетов и экзаменов.

7.1. Перечень экзаменационных вопросов

1. Дайте определение понятиям «программа», «программирование», «язык программирования». Что такое «алгоритм»? Языки программирования высокого и низкого уровней. Чем они отличаются, приведите примеры.
2. Дайте определение компилятору и интерпретатору исходных кодов программ.

3. Что такое заголовочные файлы. Для чего они предназначены? Для чего нужен препроцессор? Что делает следующая директива препроцессора: `#include <iostream>`
Объекты `cin`, `cout`. Назначение, правила использования, записи.
4. Область имен. Назначение и возможности. Что делает следующий оператор:
`using namespace std;`
Для чего предназначена следующая запись:
`int main()`
`{`
`.....`
`Return 0;`
`}`
5. Что в языке C++ называется оператором? Перечислите правила составления программ на языке C++. Порядок выполнения операторов, правила форматирования кода программы. Опишите правила записи переменных в языке C++.
6. Назовите основные типы целочисленных данных в языке C++. Почему в языке C++ несколько типов целочисленных данных? Числа с плавающей точкой. Запись чисел с плавающей точкой. Назовите типы данных с плавающей точкой. Точность типов данных с плавающей запятой. Почему при расчетах необходимо ее учитывать?
7. Тип данных `char`. Назначение, правила записи, особенности. Как с помощью программы C++ найти, какому символу соответствует код ASCII 88?
8. Операторы C++. Операторы объявления и присваивания. Что выполняет каждый из них? Арифметические операции в языке C++, запись, приоритеты. Разновидности операций деления для разных типов данных. Операция деления по модулю.
9. Логические операции в языке C++, запись, приоритеты. Тип данных `bool`. Для чего он используется? Опишите операторы цикла в языке C++ (синтаксис, принцип работы, примеры). Опишите операторы условия в языке C++ (синтаксис, принцип работы, примеры)
10. Дайте определение производному (пользовательскому) типу данных. Дайте определение типам: массив, строка, структура. Опишите правила инициализации массива в языке C++. С какого номера начинается индексация в массивах? Как соотносятся индекс последнего элемента и размер массива?

11. Дайте определение производному типу данных «строка». В чем особенность обработки строк в C++? Чем в C++ “S” отличается от ‘S’? Опишите операторы (функции) строчно-ориентированного ввода в C++.
12. Дайте определение производному типу данных «структура». Приведите пример обращения к полям структуры. Используйте структуру, описывающую работника какой-либо компании. Она должна содержать информацию о его имени, возрасте, семейном положении (состоит в браке или нет), рост (в метрах).
13. Опишите правила создания функции (процедуры). Приведите правила записи прототипа функции. Дайте пояснения, что означает каждая составляющая прототипа. Приведите пример: напишите листинг функции, возвращающей произвольную целую степень дробного числа.
14. Функции C++. Функции, возвращающие результат и не возвращающие его. Назначение, синтаксис, основные отличия. Что содержится в заголовочных файлах «.h» и файлах исходных текстов «.cpp»? Каким образом можно использовать два этих типа файлов?
15. Дайте определение типу «указатель». Для хранения какой именно информации предназначен данный тип? Приведите примеры. Каким образом, используя синтаксис C++, можно узнать адрес области памяти, выделенной для целочисленной переменной **var**? Как называется данная операция?
16. Какие операции определены для «указателей»? Опишите, каким образом можно обрабатывать массивы данных, используя «указатели».
17. Каким образом возможно выделение и освобождение памяти в C++ в процессе выполнения программы? Приведите пример объявления (листинг) динамического массива дробных чисел.
18. Дайте определение понятиям: ООП, класс, объект. Какими являются отношения между объектом и классом? Опишите общую структуру класса. Общедоступный и приватный разделы класса.
19. Функции-элементы класса. Встроенные методы. Диапазон доступа. Опишите взаимодействие функций-элементов класса с элементами разных разделов класса.
20. Деструкторы и конструкторы классов. Определение. Когда вызываются конструкторы класса? Когда вызываются деструкторы класса? Можно ли использовать имена элементов класса в качестве аргументов конструктора?
21. Опишите явный и неявный способ инициализации объекта с использованием конструкторов различных типов. Приведите пример конструктора со спецификатором **new**.

22. Особенности применения конструкторов от других методов класса. Что такое конструктор, определенный по умолчанию, и какую пользу приносит его наличие?
23. Что происходит когда вы присваиваете один объект другому объекту того же класса? Опишите процедуру в двух случаях: когда объект создан в статической и в динамической памяти компьютера.
24. Перегрузка операций. Вид операторной функции. Ограничения при выполнении перегрузки операций.
25. Виды и назначение дружественных структур. Создание дружественных конструкций. Какими правами доступа обладает дружественная функция?
26. Преобразование производных типов данных. Каким образом возможно выполнить преобразование стандартного типа данных в пользовательский? Каким образом возможно выполнить обратное преобразование?
27. Численные методы. Методы решения систем линейных уравнений. Метод Гаусса. Метод Гаусса-Жордана.
28. Численные методы. Методы решения систем линейных уравнений. Метод простых итераций.
29. Численные методы. Методы решения нелинейных уравнений. Метод половинного деления.
30. Численные методы. Методы решения нелинейных уравнений. Метод простых итераций.
31. Численные методы. Аппроксимация функций. Интерполяционные формулы Ньютона.
32. Численные методы. Аппроксимация функций. Сплайн-интерполяция.
33. Численные методы. Интегрирование. Методы прямоугольников и трапеций.
34. Численные методы. Интегрирование. Методы прямоугольников и трапеций.
35. Численные методы. Обыкновенные дифференциальные уравнения. Методы Рунге-Кутты.
36. Объектно-ориентированное программирование. Перечислите основные принципы объектной модели.
37. Объектно-ориентированное программирование. Дайте определение понятию «абстракция».
38. Объектно-ориентированное программирование. Дайте определение понятию «инкапсуляция».
39. Объектно-ориентированное программирование. Дайте определение понятию «модульность».

40. Объектно-ориентированное программирование. Дайте определение понятию «иерархия».
41. Объектно-ориентированное программирование. Дайте определение понятию «типизация».
42. Объектно-ориентированное программирование. Дайте определение понятию «объект».
43. Объектно-ориентированное программирование. Состояние и поведение объектов.
44. Объектно-ориентированное программирование. Опишите жизненный цикл объекта.
45. Объектно-ориентированное программирование. Перечислите типы отношений между объектами.
46. Объектно-ориентированное программирование. Дайте определение понятию «класс».
47. Опишите структуру класса.
48. Объектно-ориентированное программирование. Что такое абстрактный класс? Класс-интерфейс? Класс-утилита?
49. Объектно-ориентированное программирование. Перечислите типы отношений между классами.
50. Дайте определение графическому пользовательскому интерфейсу программы (GUI).
51. Библиотека Qt. Класс QApplication. Назначение, основные методы.
52. Библиотека Qt. Виджеты. Классы QLabel, QPushButton. Назначение, основные методы, сигналы, слоты.
53. Библиотека Qt. Компоновка виджетов. Менеджеры компоновки.
54. Библиотека Qt. Диалоговые окна. Класс QDialog. Назначение, основные возможности.
55. Библиотека Qt. Технология сигналов и слотов. Принцип, синтаксис, пример использования.
56. Библиотека Qt. Главное окно программы. Класс QMainWindow. Центральный виджет.
57. Библиотека Qt. Главное окно программы. Классы Qmenu, QAction.
58. Библиотека Qt. Создание пользовательских виджетов. Подходы, реализация.
59. Математический пакет MathCAD. Синтаксис, возможности программирования.
60. Математический пакет MathCAD. Функции, графические построения.
61. Среда программирования и моделирования Matlab. Синтаксис, работа с файлами.

62. Среда программирования и моделирования Matlab. Работа с массивами данных, графические построения.

7.2. Структура экзаменационного билета.

Экзаменационный билет на экзамене по дисциплине «Информатика» в первом семестре состоит из двух теоретических вопросов и одной задачи. Задачи аналогичны приведенным в разделах 6.1-6.2.

Экзаменационный билет на экзамене по дисциплине «Информатика» во втором семестре состоит из двух теоретических вопросов

8. Распределение самостоятельной работы

№ п/п	Раздел дисциплины	Тематика самостоятельной работы (детализация)	Трудоемкость (час.)	Компетенции ОК, ПК	Контроль выполнения работы
1.	Модели решения функциональных и вычислительных задач	Изучение теоретического материала. Подготовка к практическим занятиям и тестовой контрольной работе.	28	ОК-9, ПК-2	Контрольная работа.
2.	Языки и системы программирования	Изучение теоретического материала. Подготовка к практическим занятиям и тестовой контрольной работе.	122	ПК-2	Контрольная работа.
3.	Операционные системы, базы данных и локальные сети	Изучение теоретического материала. Подготовка к практическим занятиям.	8	ПК-1, ПК-2, ПК-14	Опрос на практическом занятии.

9. Методика текущего контроля освоения дисциплины

Контроль освоения дисциплины осуществляется путем применения рейтинговой системы оценки успеваемости и включает текущий контроль выполнения элементов объема дисциплины по элементам контроля с подведением текущего рейтинга.

Итоговый контроль освоения дисциплины в I семестре осуществляется на экзамене. Итоговый контроль освоения дисциплины в II семестре осуществляется на защите курсовой работы и экзамене.

Текущий контроль освоения дисциплины осуществляется в соответствии с Положением о порядке использования рейтинговой системы для оценки успеваемости студентов (приказ ректора 25.02.2010 № 1902) и основана на бально-рейтинговой системы оценки успеваемости, действующей с 2009 г., которая включает текущий контроль выполнения элементов объема дисциплины по элементам контроля с подведением текущего рейтинга и итоговый контроль.

Правила формирования пятибалльных оценок за каждую контрольную точку (КТ1, КТ2) осуществляется путем округления величины, рассчитанной по формуле:

$$KTx \Big|_{x=1,2} = \frac{(Сумма \text{ _ баллов , _ набранная \text{ _ к \text{ _ } KTx) * 5}}{Требуемая \text{ _ сумма \text{ _ баллов \text{ _ по \text{ _ } бальной \text{ _ раскладке}}}}.$$

Итоговый контроль освоения дисциплины осуществляется на экзамене по традиционной пятибалльной шкале. Обязательным условием перед сдачей экзамена является выполнение студентом необходимых по рабочей программе для дисциплины видов занятий: выполнение и защита результатов лабораторных работ, в том числе теоретических тестов (контрольных работ) и выполнение индивидуальных заданий.

Формирование итоговой суммы баллов осуществляется путем суммирования семестровой (до 70 баллов) и экзаменационной составляющих (до 30 баллов).

Таблица пересчета суммы баллов в традиционную и международную оценку

Оценка (ГОС)	Итоговая сумма баллов, учитывает успешно сданный экзамен	Оценка (ECTS)
5 (отлично)	90 - 100	A (отлично)
4 (хорошо)	85 – 89	B (очень хорошо)
	75 – 84	C (хорошо)
	70 - 74	D (удовлетворительно)
65 – 69		
3 (удовлетворительно)	60 - 64	E (посредственно)
	Ниже 60 баллов	F (неудовлетворительно)

Распределение баллов по элементам контроля в 1 семестре

Элементы учебной деятельности	Кол-во элементов	Длительность элемента, час.	Кол - во баллов за 1 элемент контроля	Срок контроля, (неделя с начала семестра)	Кол - во баллов (всего)
Посещение лекций	18	2	0.556	1-18	10
Выполнение контрольных работ (тестов)	16	1/3	1.25	1-18	20
Выполнение домашних и индивидуальных заданий	16		1.25	1-18	20
Выполнение и защита результатов лабораторных работ	16	4	1.25	1-18	20
Сдача экзамена(максимум)					30
Итого					100

Таблица распределения баллов в течение 1 семестра

Элементы учебной деятельности	Максимальный балл на 1-ую контрольную точку с начала семестра	Максимальный балл за период между 1КТ и 2КТ	Максимальный балл за период между 2КТ и на конец семестра	Всего за семестр
Посещение лекций	3.336	3.336	3.336	10
Выполнение контрольных работ (тестов)	6.25	7.5	6.25	20
Выполнение домашних и индивидуальных заданий	6.25	7.5	6.25	20
Выполнение и защита результатов лабораторных работ	6.25	7.5	6.25	20
Итого максимум за период	22	26	22	70
Сдача экзамена(максимум)				30
Нарастающим итогом	22	48	70	100

Распределение баллов по элементам контроля во 2 семестре

Элементы учебной деятельности	Кол-во элементов	Длительность элемента, час.	Кол - во баллов за 1 элемент контроля	Срок контроля, (неделя с начала семестра)	Кол - во баллов (всего)
Посещение лекций	14	2	0.714	1-17	10
Выполнение контрольных работ (тестов)	12	1/3	1.667	1-17	20
Выполнение домашних и индивидуальных заданий	12		1.667	1-17	20
Выполнение и защита результатов лабораторных работ	12	4	1.667	1-17	20
Сдача экзамена (максимум)					30
Итого					100

Таблица распределения баллов в течение 2 семестра

Элементы учебной деятельности	Максимальный балл на 1-ую контрольную точку с начала семестра	Максимальный балл за период между 1КТ и 2КТ	Максимальный балл за период между 2КТ и на конец семестра	Всего за семестр
Посещение лекций	5	5	-	10
Выполнение контрольных работ (тестов)	10	10	-	20
Выполнение домашних и индивидуальных заданий	10	10	-	20
Выполнение и защита результатов лабораторных работ	10	10	-	20
Итого максимум за период	35	35	-	70
Сдача экзамена(максимум)				30
Нарастающим итогом	35	70	70	100

Таблица распределения баллов за курсовую работу

Элементы учебной деятельности	Кол - во баллов за 1 элемент контроля	Срок контроля, (неделя с начала семестра)	Кол - во баллов (всего)
Подбор и обзор литературы	10	5	10
Написание программы	40	6-17	40
Полнота изложения ПЗ	10	на защите	10
Творческий подход	10	на защите	10
Защита работы	30	на защите	30
Итого максимум за период:			100

Таблица распределения баллов за курсовую работу в течение 2 семестра

Элементы учебной деятельности	Максимальный балл на 1-ую контрольную точку с начала семестра	Максимальный балл за период между 1КТ и 2КТ	Максимальный балл за период между 2КТ и на конец семестра	Всего за семестр
Подбор и обзор литературы	10			10
Написание программы		20	20	40
Полнота изложения ПЗ			10	5
Творческий подход			10	5
Защита работы			30	30
Итого максимум за период:	10	20	70	100
Итого	10	30	100	100

10. Учебно-методические материалы по дисциплине

При изучении дисциплины “Информатика” рекомендуется использовать учебные пособия [1-3] из списка основной учебной литературы, охватывающие все разделы программы, а также использовать дополнительную литературу для более глубокого изучения отдельных разделов курса [4-10]. Основные учебные и учебно-методические пособия размещены на научно-образовательном портале ТУСУРа, в системе управления курсами каф. СВЧиКР и в ЛВС каф.СВЧиКР.

10.1. Основная литература

1. Информатика. Базовый курс: Учебник для вузов / С. В. Симонович [и др.]; ред.: С. В. Симонович. – 2-е изд. – СПб. : Питер, 2007. – 639 с.
2. Кудинов Ю.И., Пащенко Ф.Ф. Основы современной информатики [электронный ресурс] учеб. Пособие/. - Изд. 2-е , стер,- СПб.: Лань, 2011.- 256с. Режим доступа
<http://e.lanbook.com/view/book/2024/>
3. Гловацкая А.П. Методы и алгоритмы вычислительной математики. Учеб. Пособие для вузов. – Радио и связь, 1999. – 408 с.

10.2. Дополнительная литература:

4. Мудров А.Е. Численные методы для ПЭВМ на языках Бейсик, Фортран и Паскаль. – Томск: МП "РАСКО", 1991. -272 с.
5. Прата С. Язык программирования С++. Лекции и упражнения, 6-е изд.: пер. с англ. – М.: ООО «И.Д. Вильямс», 2012. – 1248 с.
6. С/С++ для студента: [учебное пособие] / А. П. Побегайло. – СПб. : БХВ-Петербург, 2006. – 526 с.
7. С++. Объектно-ориентированное программирование. Задачи и упражнения : учебное пособие для вузов / В. В. Лаптев, А. В. Морозов, А. В. Бокова. - СПб. : Питер, 2007. – 287 с.
8. Qt. Профессиональное программирование на С++: Наиболее полное руководство / М. Шлее. - СПб. : БХВ-Петербург, 2005. – 544 с.
9. Бланшет Ж., Саммерфилд М. Qt 4: программирование GUI на С++. Пер. с англ. 2-е изд., доп. – М.: КУДИЦ-ПРЕСС, 2008. – 736 с.
10. Android за 24 часа. Программирование приложений под операционную систему Google/ Дэрсси Л., Кондер Ш. — М.: Рид Групп, 2011. — 464 с.

10.3. Перечень методических указаний по курсовой работе, практическим занятиям и лабораторным работам, самостоятельной работе студентов

11. П.А. Карпушин, С.Н. Шарангович Информатика: Методические указания по выполнению курсовой работы . Томск: [Электронный ресурс]. ТУСУР, 2011. – 29 с. Режим доступа: <http://edu.tusur.ru/training/publications/124>
12. Лавыгина А. В. Информатика: учебно-методическое пособие. Методические указания к лабораторным и практическим работам / А. В. Лавыгина; Федеральное агентство по образованию, Томский государственный университет систем управления и радиоэлектроники, Кафедра электронных приборов. – Томск : ТУСУР, 2007. – 86 с.
13. Методические указания по практическим занятиям и самостоятельной работе студентов приведены в Учебно-методическом пособии [9] – раздел 5, стр. 18-21, [10] – раздел 2, стр. 25-31.
14. Кудинов Ю. И., Пашенко Ф. Ф., Келина А. Ю. Практикум по основам современной информатики [Электронный ресурс]: учеб. пособие /. Изд. 2-е, стер. – СПб.: Лань, 2011. – 256 с. Режим доступа <http://e.lanbook.com/view/book/1799/>
15. Дубинин Д.В. Информатика [Электронный ресурс]: Описание лабораторных работ. – Томск: ТУСУР, 2009. – 60 с. Режим доступа: <http://edu.tusur.ru/training/publications/1055>
16. Дубинин Д. В. Информатика [Электронный ресурс]: Методические указания по самостоятельной работе. – Томск: ТУСУР, 2012. – 13 с. Режим доступа: <http://edu.tusur.ru/training/publications/1851>

Учебное издание

**Семкин Артем Олегович
Шарангович Сергей Николаевич**

ИНФОРМАТИКА

Учебно-методическое пособие по практическим занятиям и организации
самостоятельной работы студентов по дисциплине
«Информатика»

Формат 60x84 1/16. Усл. печ. л. _____.

Тираж _____ экз. Заказ _____.

Томский государственный университет систем управления и
радиоэлектроники.

634050, Томск, пр. Ленина, 40. Тел. (3822) 533018.