

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)
Кафедра конструирования узлов и деталей радиоэлектронной аппаратуры
(КУДР)

УТВЕРЖДАЮ
Заведующий кафедрой КУДР
_____ А.Г. Лоцилов
” ____ ” _____ 2018 г.

А.Г. Лоцилов

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ,
КУРСОВОЙ И САМОСТОЯТЕЛЬНОЙ РАБОТЕ
по дисциплине
«ОСНОВЫ ПРОГРАММИРОВАНИЯ»
для студентов направления 11.03.03
«Проектирование и технология электронно-вычислительных средств»

Томск 2018

СОДЕРЖАНИЕ

1 Цели и задачи дисциплины.....	3
1.1 Цели дисциплины	3
1.2 Задачи дисциплины	3
2 Требования к результатам освоения дисциплины.....	3
3 Содержание и порядок освоения дисциплины	4
3.1. Содержание лекционных занятий	4
3.2 Содержание и порядок выполнения лабораторных работ.....	5
3.3 Содержание и порядок выполнения курсовой работы	6
4 Вопросы для самостоятельного изучения	9
5 Учебно-методическое и информационное обеспечение дисциплины	10

1 Цели и задачи дисциплины

1.1 Цели дисциплины

Целью изучения дисциплины «Основы программирования» является обеспечение необходимого уровня компетенций студентов-бакалавров специальности 11.03.03 - «Конструирование и технология электронных средств» в области современных информационных технологий обработки и хранения информации, основ алгоритмизации и прикладного программирования с использованием языков программирования высокого уровня.

1.2 Задачи дисциплины

- знакомство с основными принципами организации записи, хранения и чтения информации;
- овладение компьютерными методами сбора, хранения и обработки информации;
- знакомство с понятием алгоритма и алгоритмическими системами;
- получение практических навыков программирования на языках программирования высокого уровня.

2 Требования к результатам освоения дисциплины

Процесс изучения дисциплины направлен на формирование следующих компетенций:

- ОПК-6 способностью осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий;
- ОПК-9 способностью использовать навыки работы с компьютером, владеть методами информационных технологий, соблюдать основные требования информационной безопасности;

В результате изучения дисциплины обучающийся должен:

- **знать** основы информационных технологий и требования к информационной безопасности; принципы организации записи, хранения и чтения информации в ЭВМ; принципы алгоритмизации и программирования; синтаксис языка программирования С/С++; основы алгоритмического и структурного программирования; основы информационных технологий и требования к информационной безопасности.
- **уметь** разрабатывать алгоритмы решения задач; работать с программными средствами общего назначения; разрабатывать программное обеспечение средствами объектно-ориентированного языка С++ и использовать его на практике.
- **владеть** навыками работы с компьютером; навыками программирования на языке высокого уровня.

3 Содержание и порядок освоения дисциплины

3.1. Содержание лекционных занятий

На лекциях преподаватель рассматривает вопросы программы курса, составленной в соответствии с государственным образовательным стандартом. Из-за недостаточного количества аудиторных часов некоторые темы не удастся осветить в полном объеме, поэтому преподаватель, по своему усмотрению, некоторые вопросы выносит на самостоятельную работу студентов, рекомендуя ту или иную литературу. Кроме этого, для лучшего освоения материала и систематизации знаний по дисциплине, необходимо постоянно разбирать материалы лекций по конспектам и литературе, рекомендуемой к курсу [1-5]. Содержание разделов лекционных занятий приведено в таблице 3.1.

Содержание разделов лекционных занятий приведено в таблице 3.1.

Таблица 3.1 – Содержание разделов лекций

№	Название раздела	Содержание раздела дисциплины по лекциям
2 семестр		
1	Основы алгоритмического и структурного программирования	Понятие алгоритма. Принципы структурного программирования. Основные понятия языка. Базовые типы данных. Переменные и константы. Имена переменных. Выражения. Ввод и вывод информации. Примеры простейших программ.
2	Операторы	Логические значения True и False в языке C. Условные операторы. Вложенные условные операторы if. Лестница if-else-if. Оператор цикла. Оператор перехода. Оператор-выражение. Блок операторов. Примеры программ с использованием операторов.
3	Массивы и строки	Одномерные массивы. Создание указателя на массив. Передача одномерного массива в функцию. Строки. Двухмерные массивы. Многомерные массивы. Индексация указателей. Инициализация массивов. Массивы переменной длины. Приемы использования массивов и строк.
4	Функции	Определение функции. Область действия функции. Аргументы функции. Оператор return. Рекурсия. Прототип функции. Объявление списков параметров переменной длины. Ключевое слово inline. Примеры использования функций.
5	Файловый ввод-вывод	Файловый ввод / вывод в C и C++. Поток и файлы. Основы файловой системы. Функции fread() и fwrite(). Ввод/вывод при прямом доступе: функция fseek(). Функции fprintf() и fscanf(). Стандартные потоки. Примеры работы с файлами

Продолжение таблицы 3.1

№	Название раздела	Содержание раздела дисциплины по лекциям
6	Структуры, объединения, перечисления и декларации	Структуры. Массивы структур. Передача структур функциям. Указатели на структуры. Массивы и структуры внутри структур. Объединения. Битовые поля. Перечисления. Важное различие между C и C++ в описании структур. Использование sizeof для обеспечения переносимости. Средство typedef. Пример работы со структурами.
7	Динамические структуры данных	Понятие указателя. Указательные переменные. Операции для работы с указателями. Указательные выражения. Указатели и массивы. Многоуровневая адресация. Инициализация указателей. Указатели на функции. Трудности при работе с указателями. Списки. Основные функции для работы со списками.
8	Основы объектно-ориентированного программирования	Основы объектно-ориентированного программирования. Описание класса. Инкапсуляция. Описание объектов. Указатель this. Конструкторы. Деструкторы. Перегрузка операций. Рекомендации по составу класса.
9	Разработка приложений с графическим интерфейсом	Окна. Класс QMainWindow. Разработка интерфейса при помощи Qt Designer. Программирование формы, созданной в Qt Designer. Ресурсы программы. Стандартные диалоги. Создание собственных диалогов. Сохранение настроек приложения

3.2 Содержание и порядок выполнения лабораторных работ

Для успешного освоения дисциплины студент должен получить необходимые навыки практической работы в рамках выполнения 12 (двенадцати) лабораторных работ.

Из них во втором семестре:

- 1) Разработка простейших программ;
- 2) Линейные алгоритмы;
- 3) Ветвления;
- 4) Циклические программы;
- 5) Массивы;
- 6) Циклические программы.

Из них в третьем семестре:

- 7) Функции;
- 8) Работа с файлами;
- 9) Структуры;
- 10) Динамические структуры данных;
- 11) Основы объектно-ориентированного программирования;
- 12) Разработка приложений с графическим интерфейсом.

Лабораторные работы выполняются с использованием электронной системы

управления обучением кафедры КУДР [6], построенной на основе системы управления курсами Moodle. Перед началом обучения студентам создаются учетные записи и осуществляется инструктаж по правилам работы в системе.

Каждая из лабораторных работ по дисциплине содержит:

1) Документ с теоретическими сведениями, необходимыми для выполнения лабораторной работы;

2) Тестирование, содержащий от 10 до 15 вопросов по теме лабораторной работы;

3) Индивидуальное задание, состоящее из 5 задач на заданную тему.

Тестирование – форма контроля, позволяющая оценить степень владения теоретическим материалом по теме лабораторной работы. Контроль правильности ответов студентов выполняется автоматически, средствами системы Moodle.

Индивидуальное задание – форма контроля навыков составления программ. Контроль программ выполняется преподавателем в режиме online, при необходимости преподаватель дает комментарии по тексту программы.

Результаты прохождения каждой из форм контроля оцениваются баллом от 0 до 5.

Критерием выполнения лабораторной работы является преодоление порогового уровня, равного 2,5 балла по каждой из форм контроля.

3.3 Содержание и порядок выполнения курсовой работы

Задачей курсового проектирования является разработка приложения в заданной предметной области, включающей в себя:

- анализ предметной области;
- разработку алгоритмов;
- разработку интерфейсов;
- программную реализацию;
- тестирование и отладку программного кода;
- подготовку и написание отчета;
- защиту проекта.

Выполнение курсового проекта предполагает самостоятельное изучение дополнительных вопросов по объектно-ориентированным языкам программирования (в частности, C++), средам проектирования Windows-приложений, а также получение практического опыта объектно-ориентированного анализа и программирования и оформления соответствующей документации на программную разработку.

Курсовой проект выполняется по типовым заданиям, студент может предложить собственный вариант задания.

Курсовая работа должна быть подготовлена к защите в срок, устанавливаемый деканатом. К защите курсовой работы представляется своевременно выполненная студентом и отрецензированная преподавателем работа.

3.3.1 Тематика курсовой работы

В ходе выполнения курсовой работы студенты должны практически освоить общий методологический подход, используемый при проектировании и программной реализации прикладного программного обеспечения.

Примерная тематика курсовых работ (проектов):

- Длинные числа.
- Матричная арифметика.
- Обращение матрицы.
- Вычисление собственных значений матрицы.
- Решение систем линейных алгебраических уравнений методом Гаусса.
- Решение систем линейных алгебраических уравнений методом Гаусса - Зейделя.
- Вычисление определённых интегралов методом прямоугольников.
- Вычисление определённых интегралов методом трапеций.
- Вычисление определённых интегралов методом Симпсона.
- Вычисление определённых интегралов методом Гаусса.
- Оптимизация функции методом золотого сечения.
- Оптимизация функции методом Ньютона.
- Оптимизация функции методом координатного спуска.
- Оптимизация функции методом градиентного спуска.
- Расчет погонной электрической емкости микрополосковой линии передачи с использованием метода сеток.
- Аппроксимация экспериментальных данных методом наименьших квадратов.
- Разработка Web-сайта.
- Компонент для отображения графиков.
- Другая тема, по согласованию с преподавателем.

3.3.2 Этапы и график выполнения курсовой работы

Курсовая работа представляет собой результат выполнения следующих взаимосвязанных этапов:

1. Этап постановки и формулирования задачи.
2. Этап проектирования.
3. Этап кодирования.

4. Этап отладки и тестирования.

5. Документирование созданного программного продукта и оформление пояснительной записки к курсовому проекту.

6. Защита.

Работа выполняется самостоятельно в свободное время и сдается в строго оговоренные сроки.

3.3.3 Порядок выполнения работы

В рамках первого этапа следует:

- произвести изучение предметной области и осуществить сбор материала в проблемно-ориентированном контексте;
- определить назначение программы, выработать требования к ней и представить их, если это возможно, в формализованном виде;
- сформулировать требования к представлению исходных данных и выходных результатов;
- определить структуры входных и выходных данных;
- сформулировать ограничения и допущения на исходные и выходные данные.

В рамках второго этапа следует:

- осуществить выбор метод реализации решения задачи;
- разработать алгоритм реализации задачи.

В рамках третьего этапа следует:

- произвести уточнение структуры входных и выходных данных и определение формата их представления;
- произвести программирование решения задачи;
- произвести комментирование текста программы и составления предварительного описания программы.

В рамках четвертого этапа следует:

- составить тесты для проверки правильности работы программы;
- произвести обнаружение, локализацию и устранение ошибок в программе, выявленных с помощью тестов;
- скорректировать код программы и ее описание.

В рамках пятого этапа следует:

- оформить курсовой проект в виде пояснительной записки согласно требованиям к курсовому проекту.

В рамках шестого этапа следует:

1) защитить работу, ответив на вопросы преподавателя.

Оценками курсовой работы могут быть: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

4 Вопросы для самостоятельного изучения

В ходе самостоятельной работе по дисциплине и при подготовке к экзамену должны быть проработаны следующие типовые вопросы, раскрывающие содержание курса.

1) Что такое алгоритм? Перечислите и объясните свойства алгоритма. Каковы правила построения имён переменных в программах на языке C/C++?

2) Какие преимущества дает объектный подход в программировании? Что такое конструктор? Может ли быть несколько конструкторов у одного класса? Что такое точечная нотация? Как она используется при работе с объектами?

3) Какие типы данных вы знаете? Что такое приоритет операций? Зачем он нужен?

В каком порядке выполняются операции, если они имеют одинаковый приоритет? Что происходит, если в выражения входят переменные разных типов? Какого типа будет результат?

4) Что такое список? Какие операции он допускает? Что такое узел? Как создать узел списка? Как добавить элемент в начало линейного списка?

5) Как выполнить обмен значений двух переменных с помощью третьей переменной? Можно ли выполнить обмен значений двух переменных без использования третьей переменной? Если да, то каким образом? Чем отличаются условные операторы в полной и неполной формах?

6) Что такое указатель? Как объявить указатель на целое число? Вещественное число? Символ? Какие данные хранятся в указателе? Как вывести адрес памяти на который ссылается указатель? Как вывести значение на которое ссылается указатель?

7) Что такое цикл? Сравните цикл с переменной и цикл с условием. Какие преимущества и недостатки есть у каждого из них? Верно ли, что любой цикл с переменной можно заменить циклом с условием? Верно ли обратное утверждение?

8) Что такое структура? В чём её отличие от массива? В каких случаях использование структур дает преимущества? Какие именно? Как объявляется новый тип данных на языке Си? Выделяется ли при этом память? Как обращаются к полю структуры?

9) Что такое символьная строка? Как хранятся строки в языке C? Как обращаться к элементу строки с заданным номером? Как вычисляется длина строки? Перечислите основные операции со строками и соответствующие им стандартные функции.

10) Какие функции для получения псевдослучайных чисел вы знаете? Как получить псевдослучайное целое число в диапазоне [a,b]? Как получить псевдослучайное вещественное число в диапазоне [a,b]?

11) Что такое сложное условие? Как определяется порядок вычислений в сложном условии? Зачем нужен оператор выбора? Как можно обойтись без него? Как в операторе выбора определить, что нужно делать, если ни один вариант не подошёл?

12) Что такое динамические структуры данных? Где выделяется память под эти данные? Как объявить в программе динамический массив и задать его размер? Как расширить массив в ходе работы программы?

13) Что такое инкапсуляция? Каковы ее цели? Чем отличаются секции public и private в описании классов? Как определить, в какую из них поместить свойство или метод? Что такое иерархия классов? Дайте полное определение ООП и объясните его.

14) С помощью каких функций осуществляется чтение/запись в двоичные файлы? Каковы преимущества (недостатки) работы с двоичными файлами? Какие параметры существуют у функции fopen и каково их назначение? Как узнать, что файл не удалось открыть? Каковы возможные причины такой ошибки?

15) Чем отличаются текстовые и двоичные файлы по внутреннему содержанию? Объясните последовательность действий при работе с файлами. Что такое файловая переменная? В каком случае одна и та же файловая переменная может быть использована для работы с несколькими файлами, а в каком – нет?

5 Учебно-методическое и информационное обеспечение дисциплины

1. Язык программирования C++. Лекции и упражнения [Текст] : научно-популярное издание / С. Прата ; пер.: Ю. И. Корниенко, А. А. Моргунова. - 6-е изд. - М. : Вильямс, 2013. - 1248 с. (наличие в библиотеке ТУСУР - 15 экз.)

2. C/C++. Программирование на языке высокого уровня [Текст] : учебник для вузов / Т. А. Павловская. - СПб. : ПИТЕР, 2013. - 461 с. (наличие в библиотеке ТУСУР - 35 экз.)

3. Объектно-ориентированное программирование: Курс лекций / Панов С. А., Ганджа Т. В. - 2015. 110 с. [Электронный ресурс] - Режим доступа: <https://edu.tusur.ru/publications/5009> (дата обращения: 18.06.2018).

4. Кудинов Ю. И. Пашенко Ф. Ф. Основы современной информатики [Электронный ресурс]: учеб. пособие - Изд. 2-е, стер. - СПб. : Лань, 2011. - 256 с. – Электрон. текстовые дан. [Электронный ресурс] - Режим доступа: <https://e.lanbook.com/reader/book/68468/#1> (дата обращения: 18.06.2018).

5. Язык программирования C++. Лекции и упражнения [Текст] : научно-популярное издание / С. Прата ; пер.: Ю. И. Корниенко, А. А. Моргунова. - 6-е изд. - М. : Вильямс, 2013. - 1248 с. (наличие в библиотеке ТУСУР - 15 экз.)

6. Основы программирования // [Электронный курс в системе Moodle] / Томский государственный университет систем управления и радиоэлектроники. [Томск, 2017]. [Электронный ресурс] - Режим доступа: <http://kudr.tusur.ru:81/moodle/course/view.php?id=3> (дата обращения: 18.06.2018).