

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМЕНИ В.Ф. УТКИНА»

Факультет вычислительной техники
Кафедра «Вычислительная и прикладная математика»

СОГЛАСОВАНО


Декан факультета
вычислительной техники


Д.А. Перепелкин
« 26 » 06 2020 г.



УТВЕРЖДАЮ

Проректор РОПиМД


А.В. Корячко
« 26 » 06 2020 г.

Заведующий кафедрой:

ВПМ  Г.В. Овечкин
« 26 » 06 2020 г.

РАБОЧАЯ ПРОГРАММА

ДИСЦИПЛИНЫ

Б1.В.10 «Низко-уровневое программирование»

Направление подготовки — 09.03.04 «Программная инженерия»

Уровень подготовки — академический бакалавриат
Квалификация (степень) выпускника — бакалавр

Форма обучения — очная

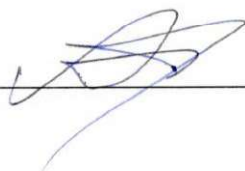
Рязань 2020 г.

ЛИСТ СОГЛАСОВАНИЙ

Рабочая программа составлена с учетом требований федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.04 – «Программная инженерия»,
утвержденного _____ №920 от 19.09.2017 г. _____
(дата утверждения ФГОС ВО)

Разработчик:

доцент кафедры ВПМ



О.В. Антипов

Программа рассмотрена и одобрена на заседании кафедры ВПМ «11» июня 2020 г.,
протокол № 11.

Заведующий кафедрой ВПМ

Г.В. Овечкин

1. ЦЕЛЬ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью изучения дисциплины является ознакомить студентов с базовыми низкоуровневыми возможностями языка C, с его сильными и слабыми сторонами, дать понятие об области его применения и подходах к решению стандартных алгоритмических задач с его помощью. Познакомить студентов с моделями памяти и принципами работы с динамической памятью, с данными на уровне байтов и битов, включая изучение битовых операций и области их применения, динамическое построение структур данных и способы их представления в памяти, а также изучить принципы компиляции программ, применение специальных директив препроцессора, основы модульного программирования, работу с бинарными и текстовыми файлами, использование этих структур для организации данных.

Основные задачи освоения учебной дисциплины:

1. Получение системы базовых знаний о низкоуровневом программировании.
2. Формирование представления о языке C и его использовании в решении различных предметных задач.
3. Систематизация и закрепление практических навыков и умений по программированию на языке C.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина относится к вариативной части профессиональных дисциплин блока № 1 (Б1.В.09), изучается по очной форме обучения на 1 курсе во 2 семестре, базируется на знаниях, полученных по программе среднего (полного) общего образования или среднего профессионального образования, а также в ходе изучения дисциплин: «Информатика», «Алгоритмические языки и программирование», «Основы программной инженерии».

Пререквизиты дисциплины. Для освоения дисциплины обучающиеся должны

- *знать:*
 - основные понятия теории алгоритмов и базовые принципы алгоритмизации;
 - основы процедурного программирования, архитектуру программ, стандартные управляющие конструкции, иметь понятие о структурах и типах данных;
 - стандартные алгоритмы обработки структур данных;
 - способы представления данных в памяти компьютера;
 - принципы нисходящего проектирования и структурного программирования;
- *уметь:*
 - применять свои знания при решении различных предметных задач;
 - работать в средах программирования;
 - разрабатывать, отлаживать и выполнять императивные программы;
- *владеть:*
 - навыками алгоритмизации, разработки и отладки императивных программ;
 - методами и приемами анализа и структурирования сложных программ.

Постреквизиты дисциплины. Полученные знания используются при изучении дисциплин «Операционные системы», «Объектно-ориентированное программирование», «Конструирование программного обеспечения», «Разработка компиляторов», «Разработка системных утилит», при выполнении НИР и при подготовке выпускной квалификационной работы.

В результате изучения дисциплины студенты должны

- *знать:*
 - назначение и области применения языка C;
 - синтаксис и семантику языка, операторы и операции языка, специальные значения, стандартные типы данных;
 - модели памяти, принципы и способы управления динамической памятью;

- способы динамического построения сложных структур данных непосредственно в памяти компьютера, побитовых операции обработки данных и сферу их применения;
- *уметь*:
 - применить свои знания к решению предметных задач;
 - составлять, отлаживать и выполнять низкоуровневые программы;
- *владеть*:
 - навыками разработки и отладки С программ с помощью встроенных средств отладки;
 - навыками работы с данными в памяти компьютера на уровне байт и битов;
 - навыками работы с данными из бинарных и текстовых файлов.

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Процесс изучения дисциплины направлен на формирование следующих компетенций в соответствии с ФГОС ВО, ПООП (при наличии) по данному направлению подготовки, а также компетенций (при наличии), установленных университетом.

Рекомендуемые профессиональные компетенции выпускников и индикаторы их достижения (при наличии)

Задача ПД	Объект или область знания	Код и наименование профессиональной компетенции	Код и наименование индикатора достижения профессиональной компетенции	Обоснование (ПС, анализ опыта)
Направленность (профиль), специализация: Программная инженерия				
Тип задач профессиональной деятельности: проектный				
Формирование требований к информатизации и автоматизации прикладных процессов, формализация предметной области проекта; технико-экономическое обоснование проектных решений и составление технического задания на разработку ПП; проектирование программно-аппаратных средств в соответствии с техническим заданием; применение современных инструментальных средств при разработке ПО; документирование компонентов ИС на всех стадиях ЖЦ	Прикладные и информационные процессы Информационные технологии Программное обеспечение	ПК-4. Владение навыками моделирования, анализа и использования формальных методов конструирования ПО	ИД – 1 _{ПК-4} <i>Знать</i> : основы моделирования и формальные методы конструирования ПО. ИД – 3 _{ПК-4} <i>Уметь</i> : использовать формальные методы моделирования и конструирования ПО. ИД – 3 _{ПК-4} <i>Владеть</i> : методами формализации, моделирования и конструирования ПО.	06.001 Программист 06.004 Специалист по тестированию в области информационных технологий 06.022 Системный аналитик 06.028 Системный программист
Тип задач профессиональной деятельности: производственно-технологический				
Проведение работ по инсталляции ПО АС и загрузки БД; настройка параметров ИС и тестирование результатов настройки; ведение технической документации;	Информационные технологии Программное обеспечение	ПК-10. Владение навыками использования различных технологий разработ-	ИД – 1 _{ПК-10} <i>Знать</i> : современные технологии разработки ПО. ИД – 2 _{ПК-10} <i>Уметь</i> : использовать современные	06.001 Программист 06.004 Специалист по тестированию в области информаци-

техническое сопровождение ИС в процессе эксплуатации; применение Web-технологий при реализации удаленного доступа в системах клиент-сервер и распределенных вычислений		ки ПО	технологии разработки ПО. ИД – 3ПК-10 <i>Владеть:</i> навыками использования современных технологий разработки ПО.	онных технологий 06.022 Системный аналитик 06.028 Системный программист
--	--	-------	--	---

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1. Объем дисциплины по семестрам (курсам) и видам занятий в зачетных единицах

с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам занятий) и на самостоятельную работу обучающихся

Общая трудоемкость изучения дисциплины для очной формы обучения составляет 4 зачетных единиц (ЗЕ), 144 академических часа.

Объем дисциплины	Всего часов	Семестр 2
Общая трудоемкость дисциплины, в том числе:	144	144
1. Контактная работа обучающихся с преподавателем (всего), в том числе:	48,25	48,25
Лекции	24	24
лабораторные работы	16	16
практические занятия	8	8
иная контактная работа (ИКР)	0,25	0,25
консультация	-	-
2. Самостоятельная работа	87	87
3. Курсовой проект	-	-
4. Контроль	8,75	8,75
Вид промежуточной аттестации		Зачет с оценкой

4.2. Содержание дисциплины, структурированное по разделам и темам

В структурном отношении программа дисциплины представлена следующими разделами и темами:

I раздел. Абстракция и «низкоуровневость» языков программирования.

Основные концепции языков программирования. Архитектура фон Неймана. Эволюция языков программирования. Уровни абстракции. Низкоуровневые, высокоуровневые и сверхвысокоуровневые языки. Поколения языков программирования. Проблема трансляции языка программирования. Компиляторы и интерпретаторы. Система обработки языка.

II раздел. Синтаксис и семантика языка C.

Тема 1. Общие сведения о языке C. Назначение языка. Основные особенности. Структура программы. Комментарии. Идентификаторы. Переменные и константы. Простые типы. Автоматическое приведение типов. Явное преобразование типов. Стандартные потоки ввода-вывода. Вывод данных. Специальные символы. Ввод данных. Базовые унарные операторы. Базовые бинарные операторы. Тернарный оператор. Выражения и приоритет операций. Стандартные математические функции.

Тема 2. Управляющие операторы. Пустой оператор. Инструкции. Условный оператор. Оператор выбора. Операторы цикла. Функции и параметры. Массивы. Одномерные массивы. Многомерные массивы.

III раздел. Указатели и адреса памяти.

Указатель. Тип указателя. Разыменовывание указателя. Взятие адреса памяти. Нулевые указатели. Передача параметров функции. Массивы и указатели. Представление в памяти статического одномерного и многомерного массива. Передача массивов в функцию в качестве параметра. Указатель типа *void*. Адреса функций.

IV раздел. Представление памяти процесса. Управление памятью.

Тема 1. Область неинициализированных данных. Область инициализированных данных. Сегмент текста программы. Область стека (*stack*). Область кучи (*heap*). Модели памяти. Автоматическая память. Регистры процессора. Статическая память. Динамическая память. Сравнение динамического и статического выделения памяти. Работа с динамической памятью. Выделение динамической памяти. Освобождение динамической памяти. Выделение памяти под массив.

Тема 2. Массивы и указатели. Изменение размера ранее выделенной памяти. Копирование и заполнение памяти. Работа с динамическими многомерными массивами и способ их представления в памяти. «Утечки» памяти.

V раздел. Строковый тип данных.

Строковые константы. Массивы символов. Указатель на символьный тип. Массивы строк. Операции со строками. Инициализация строк. Ввод и вывод строк. Основные функции стандартной библиотеки *string.h*.

VI раздел. Компиляция, сборка, препроцессор.

Принципы компиляции программ на языке C. Структура компилятора. Исходная программа. Препроцессор. Модифицированная исходная программа. Компилятор. Целевая ассемблерная программа. Ассемблер. Перемещаемый машинный код. Компоновщик/загрузчик. Библиотечные файлы. Перемещаемые объектные файлы. Целевой машинный код. Директивы препроцессора. Директивы *#include*, *#define*, *#undef*. Условная компиляция.

VII раздел. Модульное программирование.

Модуль. Интерфейс модуля. Реализация модуля. Принципы компиляции модуля. Раздельная компиляция. Компоновщик и модули. Глобальные переменные. Статические переменные. Спецификатор *static*.

VIII раздел. Собственные типы данных.

Тема 1. Объявление типа с помощью *typedef*. Перечисления (*enum*). Перечисления и читаемость кода. *typedef* и перечисления. Задание значений перечислений. Побитовые операции. Побитовые логические операции. Операции битового сдвига. Битовые флаги и маски. Флаги и перечисления. Упаковка значений с помощью битовых операций.

Тема 2. Структуры. Структуры и функции. Массивы структур. Указатели и структуры. Представление структур в памяти компьютера. «Выравнивание» структур. Структуры, ссылающиеся на себя. Поля. Объединения.

IX раздел. Работа с файлами.

Бинарные и текстовые файлы. Файловая система. Устройство внешней памяти. Файловая структура *FAT32*. Блочная адресация. Сектора. Кластеры. Загрузочный сектор. Таблица *FAT*. Работа с файлами в языке C. Поток ввода-вывода. Открытие файлов. Режимы доступа. Закрытие файла. Чтение и запись в файл. Произвольный доступ к файлу.

**4.3. Разделы дисциплины и трудоемкость по видам учебных занятий
(в академических часах)**

№ п/п	Тема	Общая трудоемкость, всего часов	Контактная работа обучающихся с преподавателем				Самостоятельная работа обучающихся
			всего	лекции	лабораторные работы	практические занятия	
1	2	3	4	5	6	7	8
1.	Абстракция и «низкоуровневость» языков программирования	9	2	2			7
2.	Язык C	25	10	4	4	2	15
2.1.	Структура программы. Простые типы. Потoki ввода-вывода. Базовые операторы. Выражения и операции. Стандартные математические функции	8	3	2		1	5
2.2.	Управляющие операторы. Функции. Массивы	17	7	2	4	1	10
3.	Указатели и адреса памяти	10	5	2	2	1	5
4.	Представление памяти процесса. Управление памятью	22	7	4	2	1	15
4.1.	Области данных, текста программы, стека, кучи. Модели памяти. Работа с динамической памятью	7	2	2			5
4.2.	Массивы и указатели. Динамические массивы	15	5	2	2	1	10
5.	Строковый тип данных	9	4	2	2		5
6.	Компиляция, сборка, препроцессор	7	2	2			5
7.	Модульное программирование	15	5	2	2	1	10
8.	Собственные типы данных	22	7	4	2	1	15
8.1.	Объявление типа. Перечисления. Побитовые операции	7	2	2			5
8.2.	Структуры и функции. Массивы структур. Указатели и структуры	15	5	2	2	1	10
9.	Работа с файлами.	16	6	2	2	2	10
	<i>Зачет</i>	9					9
	Всего:	144	48	24	16	8	96

4.3.1. Лекционные занятия

№ п/п	Темы лекционных занятий	Трудоемкость (час.)	Формируемые компетенции	Форма контроля
1.	Абстракция и «низкоуровневость» языков программирования	2	ПК-4, ПК-10	зачет
2.	Язык C. Структура программы. Простые типы. Потоки ввода-вывода. Базовые операторы. Выражения и операции. Стандартные математические функции	2	ПК-4, ПК-10	зачет
3.	Управляющие операторы. Функции. Массивы	2	ПК-4, ПК-10	зачет
4.	Указатели и адреса памяти	2	ПК-4, ПК-10	зачет
5.	Области данных, текста программы, стека, кучи. Модели памяти. Работа с динамической памятью	2	ПК-4, ПК-10	зачет
6.	Массивы и указатели. Динамические массивы	2	ПК-4, ПК-10	зачет
7.	Строковый тип данных	2	ПК-4, ПК-10	зачет
8.	Компиляция, сборка, препроцессор	2	ПК-4, ПК-10	зачет
9.	Модульное программирование	2	ПК-4, ПК-10	зачет
10.	Объявление типа. Перечисления. Побитовые операции	2	ПК-4, ПК-10	зачет
11.	Структуры и функции. Массивы структур. Указатели и структуры	2	ПК-4, ПК-10	зачет
12.	Работа с файлами	2	ПК-4, ПК-10	зачет

4.3.2. Практические занятия (семинары)

№ п/п	Тематика практических занятий (семинаров)	Трудоемкость (час.)	Формируемые компетенции	Форма контроля
1.	Язык C. Структура программы. Простые типы. Базовые операторы. Выражения и операции. Стандартные математические функции	2	ПК-4, ПК-10	зачет
2.	Указатели и адреса памяти. Представление памяти процесса. Управление памятью. Динамические массивы	2	ПК-4, ПК-10	зачет
3.	Модульное программирование. Структуры и функции. Массивы структур. Указатели и структуры	2	ПК-4, ПК-10	зачет
4.	Работа с файлами. Бинарные и текстовые файлы.	2	ПК-4, ПК-10	зачет

4.3.3. Лабораторные занятия

Лабораторные работы по дисциплине «Низкоуровневое программирование» для направления подготовки 09.03.04 – «Программная инженерия» имеют своей целью изучение технологии и основных приемов низкоуровневого программирования, освоение языка программирования C, приобретение навыков разработки и отладки C программ.

В результате выполнения лабораторного практикума студенты должны уметь:

- разрабатывать, отлаживать с помощью встроенных средств отладки и выполнять низкоуровневые программы на языке программирования C;
- работать с данными в памяти компьютера на уровне байт и битов;
- работать с данными из бинарных и текстовых файлов;
- применить свои знания к решению определенных предметных задач.

№ п/п	Наименование лабораторных работ	Трудоемкость (час.)	Формируемые компетенции	Форма контроля
1.	Управляющие структуры языка C	4	ПК-4, ПК-10	зачет
2.	Массивы и функции языка C	2	ПК-4, ПК-10	зачет
3.	Указатели и ссылки. Передача параметров в функцию по указателю	2	ПК-4, ПК-10	зачет
4.	Динамические массивы	2	ПК-4, ПК-10	зачет
5.	Обработка символов и строк	2	ПК-4, ПК-10	зачет
6.	Структуры. Модули	2	ПК-4, ПК-10	зачет
7.	Работа с файлами в языке C. Потоки ввода-вывода	2	ПК-4, ПК-10	зачет

4.3.4. Самостоятельная работа

Самостоятельное изучение тем учебной дисциплины способствует закреплению знаний, умений и навыков, полученных в ходе аудиторных занятий, углублению и расширению знаний по отдельным вопросам и темам дисциплины; освоению умений практического использования полученных знаний по разработке программ различного прикладного характера.

Самостоятельная работа обучающихся по данному курсу заключается:

- при подготовке к лекциям и практическим занятиям в изучении и доработке конспекта лекции с применением учебно-методической литературы, подборе дополнительных примеров к теоретическим положениям курса по данной теме;
- при подготовке к лабораторным работам в разработке, отладке и выполнении программы своего варианта задания по данной теме;
- при самостоятельном изучении отдельных вопросов и тем курса с применением рекомендуемой учебно-методической литературы;
- при подготовке к зачету в изучении, осмыслении и повторении пройденного теоретического материала и выполненных практических заданий с применением конспекта лекций и учебно-методической литературы.

№ п/п	Тематика самостоятельной работы	Трудоемкость (час.)	Формируемые компетенции	Форма контроля
1.	Абстракция и «низкоуровневость» языков программирования	7	ПК-4, ПК-10	зачет
2.	Язык С. Структура программы. Простые типы. Потоки ввода-вывода. Базовые операторы. Выражения и операции. Стандартные математические функции	5	ПК-4, ПК-10	зачет
3.	Управляющие операторы. Функции. Массивы	10	ПК-4, ПК-10	зачет
4.	Указатели и адреса памяти	5	ПК-4, ПК-10	зачет
5.	Области данных, текста программы, стека, кучи. Модели памяти. Работа с динамической памятью	5	ПК-4, ПК-10	зачет
6.	Массивы и указатели. Динамические массивы	10	ПК-4, ПК-10	зачет
7.	Строковый тип данных	5	ПК-4, ПК-10	зачет
8.	Компиляция, сборка, препроцессор	5	ПК-4, ПК-10	зачет
9.	Модульное программирование	10	ПК-4, ПК-10	зачет
10.	Объявление типа. Перечисления. Побитовые операции	5	ПК-4, ПК-10	зачет
11.	Структуры и функции. Массивы структур. Указатели и структуры	10	ПК-4, ПК-10	зачет
12.	Работа с файлами. Бинарные и текстовые файлы.	10	ПК-4, ПК-10	зачет

5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине представлен в виде оценочных материалов и приведен в **Приложении** к рабочей программе дисциплины (см. документ «Оценочные материалы по дисциплине «Низкоуровневое программирование»).

6. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Основная литература

1. Керниган Б., Ритчи Д. Язык программирования Си. — 2-е изд. — М.: Вильямс, 2007. — С. 304. — ISBN 0-13-110362-8.
2. Столяров А. В. Язык Си и начальное обучение программированию. — Издательский отдел факультета ВМК МГУ, 2010. — № 7. — С. 78—90.
3. Подбельский В. В., Фомин С. С. Курс программирования на языке Си: учебник. — М.: ДМК Пресс, 2012. — 318 с. — ISBN 978-5-94074-449-8.
4. Прата С. Язык программирования С: Лекции и упражнения. — М.: Вильямс, 2006. — С. 960. — ISBN 5-8459-0986-4.
5. Прата С. Язык программирования С (С11). Лекции и упражнения, 6-е издание. — М.: Вильямс, 2015. — 928 с. — ISBN 978-5-8459-1950-2.

6.2. Дополнительная литература

6. Шилдт Г. С: полное руководство, классическое издание. — М.: Вильямс, 2010. — С. 704. — ISBN 978-5-8459-1709-6.
7. Гукин Д. Язык программирования Си. — М.: Диалектика, 2006. — С. 352. — ISBN 0-7645-7068-4.
8. Языки программирования Ада, Си, Паскаль / А. Фьюэр, Н. Джехани. — М.: Радио и Связь, 1989. — 368 с. — 50 000 экз. — ISBN 5-256-00309-7.

6.3. Методические указания для обучающихся по освоению дисциплины

6.3.1. Методические указания к освоению лекционного материала

Для освоения лекционного материала следует: изучить конспект лекции в тот же день, после лекции: 10 – 15 минут, повторно прочитать конспект лекции за день перед следующей лекцией: 10 – 15 минут. Также следует изучить теоретический лекционный материал по рекомендованному учебнику/учебному пособию: 1 час в неделю.

Следует максимально использовать лекционное время для изучения дисциплины, понимания лекционного материала и написания конспекта лекций. В процессе лекционного занятия студент должен уметь выделять важные моменты и основные положения. При написании *конспекта лекций* следует придерживаться следующих правил и рекомендаций.

1. При ведении конспекта рекомендуется структурировать материал по разделам, главам, темам. Вести нумерацию формул, схем, рисунков. Выделять по каждой теме постановку задачи, основные положения, выводы. Кратко записывать те пояснения лектора, которые оказались особенно важными. Это позволит при подготовке к сдаче зачёта и экзамена не запутаться в структуре лекционного материала.

2. Лекционный материал следует записывать в конспект лишь после того, как излагаемый лектором тезис будет дослушан до конца и понят.

3. При конспектировании следует отмечать непонятные, на данном этапе, положения, доказательства и пр.

6.3.2. Методические указания к практическим/лабораторным занятиям

Практические занятия по дисциплине существенно дополняют лекции. В процессе анализа теоретических положений и решения практических задач студенты расширяют и углубляют свои знания, полученные из лекционного курса и учебников, приобретают умение применять общие закономерности к конкретным случаям. В процессе решения задач развивается логическое мышление, и вырабатываются навыки вычислений, работы со справочной литературой. Практические занятия способствуют закреплению знаний и практических навыков, формированию конструктивного стиля мышления, расширению кругозора.

При подготовке к практическому занятию необходимо внимательно ознакомиться с соответствующим теоретическим материалом по конспекту лекций и рекомендованному учебнику, затем изучить конспект или материалы предыдущего практического занятия и выполнить заданное расчетное задание: 1 – 2 часа в неделю.

Следует максимально использовать аудиторное время практических занятий. В процессе занятия студент должен активно участвовать в дискуссиях, обсуждениях и решениях практических задач и вести *конспект практических занятий* отдельно от конспекта лекций.

Дополнительно в часы самостоятельной работы студенты могут повторно решить задачи, с которыми они плохо освоились во время аудиторных занятий, и обязательно те задачи, которые не получились дома при предыдущей подготовке к практическим занятиям.

Подготовка к лабораторным работам. Перед началом лабораторной работы необходимо ознакомиться с методическими указаниями к данной лабораторной работе, внимательно ознакомиться с заданием и желательно заранее выполнить подготовку программного проекта в используемой инструментальной среде, чтобы время лабораторного занятия использовать для исправления ошибок, модификации проекта и защиты данной работы.

Выполнение каждой из запланированных работ заканчивается предоставлением отчета. Требования к форме и содержанию отчета приведены в методических указаниях к лабораторным работам или определяются преподавателем на первом занятии.

Отчет по лабораторной работе студент должен начать оформлять еще на этапе подготовки к ее выполнению. Для допуска к лабораторной работе, студент должен представить преподавателю «заготовку» отчета, содержащую: оформленный титульный лист или название и номер работы при ведении общего конспекта, цель работы, задание, проект решения, и при наличии полученные результаты, выводы.

Изучение методических указаний к лабораторной работе – 2 часа перед выполнением лабораторной работы и в ходе разработки проекта и 2 часа для оформления отчета, отладки проекта и подготовки к сдаче работы.

После выполнения лабораторной работы необходимо согласовать полученные результаты с преподавателем. Важным этапом является *защита лабораторной работы*. В процессе защиты студент отвечает на вопросы преподавателя, касающиеся теоретического материала, относящегося к данной работе, и проекта, реализующего его задание, комментирует полученные в ходе работы результаты. При подготовке к защите лабораторной работы рекомендуется ознакомиться со списком вопросов по изучаемой теме и попытаться самостоятельно на них ответить, используя конспект лекций и рекомендуемую литературу. Кроме чтения учебной литературы рекомендуется активно использовать информационные ресурсы сети Интернет по изучаемой теме.

6.3.3. Методические указания к подготовке и сдаче теоретического зачета

Теоретический зачет – форма промежуточной проверки знаний, умений, навыков, степени освоения дисциплины. Главная задача зачета состоит в том, чтобы у студента по окончании изучения данной дисциплины сформировались определенное представление об общем содержании дисциплины, определенные теоретические знания и практические навыки, определенный кругозор. Готовясь к зачету, студент приводит в систему знания, полученные на лекциях, на практических и лабораторных занятиях, разбирается в том, что осталось непонятным, и тогда изучаемая им дисциплина может быть воспринята в полном объеме с присущей ей строгостью и логичностью, ее практической направленностью.

Зачеты дают возможность преподавателю определить теоретические знания студента и его практические навыки при решении определенных прикладных задач. Оцениваются: понимание и степень усвоения теоретического материала; степень знакомства с основной и дополнительно литературой, а также с современными публикациями; умение применить теорию к практике, решать определенные практические задачи данной предметной области, правильно проводить расчеты и т. д.; знакомство с историей данной науки; логика, структура и стиль ответа, умение защищать выдвигаемые положения.

Значение зачетов не ограничивается проверкой знаний, являясь естественным завершением обучения студента по данной дисциплине, они способствуют обобщению и закреплению знаний и умений, приведению их в стройную систему, а также устранению возникших в процессе обучения пробелов.

Подготовка к зачету – это тщательное изучение и систематизация учебного материала, осмысление и запоминание теоретических положений, формулировок, формул, установление и осмысление внутри предметных связей между различными темами и разделами дисциплины, закрепление теоретических знаний путем решения определенных задач.

Перед зачетом преподаватель может назначить *консультацию*, ее цель – дать ответы на вопросы, возникшие в ходе самостоятельной подготовки студента, студент имеет возможность получить ответ на все неясные ему вопросы, кроме того, преподаватель будет отвечать на вопросы других студентов, что будет способствовать повторению и закреплению знаний всех присутствующих. Преподаватель на консультации, как правило, обращает внимание на те разделы, по которым на предыдущих зачетах ответы были неудовлетворительными, а также фиксирует внимание на наиболее трудных разделах или темах курса.

Планируя подготовку к сдаче зачета, обучаемый должен учитывать сразу несколько факторов: сложность учебного материала и степень его проработки в ходе обучения, свои индивидуальные способности. Подготовка к зачету следует начинать с общего планирования деятельности, с определения объема материала, подлежащего проработке, необходимо внимательно сверить конспекты с программой дисциплины, чтобы убедиться, все ли разделы отражены в лекциях, отсутствующие темы изучить по учебнику. Второй этап предусматривает системное изучение материала по предмету с обязательной записью всех выкладок, выводов, формул. На третьем этапе – этапе закрепления – полезно чередовать повторение особенно сложных вопросов с беглым повторением всего материала.

6.3.4. Рекомендации по работе с литературой

Теоретический материал курса становится более понятным, когда дополнительно к прослушиванию лекции и изучению конспекта изучаются и книги по данному предмету. Литературу по дисциплине рекомендуется читать как в бумажном, так и в электронном виде (если отсутствует бумажный аналог). Полезно использовать несколько учебников и пособий по дисциплине. Рекомендуется после изучения очередного параграфа ответить на несколько вопросов по данной теме. Полезно мысленно задать себе следующие вопросы (и попробовать ответить на них): «о чем этот параграф?», «какие новые понятия введены, каков их смысл?», «зачем мне это нужно по специальности?».

Рекомендуется самостоятельно изучать материал, который еще не прочитан на лекции и не применялся на лабораторном или практическом занятии, тогда занятия будут гораздо понятнее. В течение недели рекомендуется выбрать время (1 час) для работы с литературой.

6.3.5. Образовательные технологии, используемые в учебном процессе

Изучение дисциплины построено на методике *когнитивного или проблемного обучения*, как наиболее эффективной технологии при подготовке специалиста технического профиля. В рамках такого обучения студенты с самого начала учатся мыслить как программные инженеры, концентрировать внимание на решаемой проблеме, изучать требования и особенности проектирования перед созданием кода программы, работать по итеративной схеме, т.е. вырабатывают в себе привычку думать обо всех деталях, необходимых для разработки больших программ. При подготовке программных инженеров программирование остается основополагающим навыком, чем раньше и дольше студенты будут программировать, тем лучшими специалистами они смогут стать в дальнейшем, большинство возможностей их будущей работы требуют наличия профессиональных навыков программирования, поэтому необходимо делать основной упор на получение таких навыков. Применение данного подхода способствует тому, что студенты с самого начала чувствуют, что обучаются выбранной специальности.

Для обеспечения результативности профессиональной практики рекомендуется поддерживать рейтинговую систему оценки знаний текущей успеваемости, выполнения лабораторных заданий. Активно поощряется профессионально выполненная работа с обязательным профессиональным оформлением отчетов.

Принятая технология обучения предусматривает применение активных форм обучения, таких как интерактивные методы чтения лекций и проведения практических и лабораторных занятий, организация устных презентаций в группе для конструктивного обсуждения и оценивания выполненных студенческих работ с целью демонстрации прогресса в обучении и для выработки у студентов лучшего понимания изучаемой проблематики, использование стандартизированных тестов для отслеживания и оценивания общего прогресса в группе и у каждого студента в отдельности.

Чтение лекций и проведение практических и лабораторных занятий осуществляется с использованием мультимедийных средств и раздаточных материалов, с привлечением обучающихся и контролирующих учебных курсов.

7. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ ИНТЕРНЕТ, НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1. Электронно-библиотечная система «Лань». – Режим доступа: с любого компьютера РГРТУ без пароля. – URL: <https://e.lanbook.com/>
2. Электронно-библиотечная система «IPRbooks». – Режим доступа: с любого компьютера РГРТУ без пароля, из сети Интернет по паролю. – URL: <https://iprbookshop.ru/>.
3. Электронная библиотека РГРТУ. – URL: <http://weblib.rrtu/ebs>.
4. Научная электронная библиотека eLibrary. – URL: <http://e.lib/vlsu.ru/www.uisrussia.msu.ru/elibrary.ru>
5. Библиотека и форум по программированию. – URL: <http://www.cyberforum.ru>
6. Национальный открытый университет ИНТУИТ. – URL: <http://www.intuit.ru/>
7. Информационно-справочная система. – URL: <http://window.edu.ru>

8. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ

1. Операционная система Windows XP (MicrosoftImagine, номерподписки 700102019, бессрочно).
2. Операционная система Windows XP (MicrosoftImagine, номерподписки ID 700565239, бессрочно).
3. OpenOffice (OpenOffice.org, OO.org, OO.o, OOo) – свободно распространяемое ПО.
4. MicrosoftOfficeVisio (MicrosoftImagine:Номер подписки 700102019, бессрочно).
5. Dev-C++ – свободная интегрированная среда разработки приложений для языков программирования C/C++. В дистрибутив входит компилятор GCC. Распространяется согласно GPL.

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для освоения дисциплины необходимы следующие материально-технические ресурсы:

- 1) аудитория для проведения лекционных и практических занятий, групповых и индивидуальных консультаций, текущего контроля, промежуточной аттестации, оборудованная проектором и маркерной (меловой) доской;
- 2) классы, оснащенные персональными компьютерами для проведения лабораторных занятий;
- 3) аудитория для самостоятельной работы, оснащенная компьютерной техникой с подключением к локальной вычислительной сети и сети Интернет.

Наименование специальных помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы	Перечень лицензированного ПО. Реквизиты подтверждающего документа
Учебная аудитория для проведения практический занятий, лабораторных работ и самостоятельной работы №206-3	Проектор: InFocus LP640 18 ПК с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду: ЦП: Intel Pentium 4 class 2800 ОЗУ: 1 Гб ПЗУ: 50 Гб (11 шт.)	1. Операционная система Windows XP/Vista/7/8/10 (Microsoft Imagine: Номер подписки 700102019, бессрочно). 2. Microsoft Visual Studio (Microsoft Imagine: Номер подписки 700102019, бессрочно). 3. Microsoft Office Access (Microsoft Imagine: Номер подписки 700102019, бессрочно).

	<p>ЦП: Intel Pentium 4 class 3200 ОЗУ: 1 Гб ПЗУ: 50 Гб (5 шт.) ЦП: Intel Pentium 4 class 2800 ОЗУ: 500 Мб ПЗУ: 50 Гб (1 шт.) ЦП: Intel Pentium 4 class 2800 ОЗУ: 2 Гб ПЗУ: 50 Гб (1 шт.)</p>	<p>4. Microsoft Office Visio (Microsoft Imagine: Номер подписки 700102019, бессрочно). 5. Microsoft SQL Server (Microsoft Imagine: Номер подписки 700102019, бессрочно). 6. Microsoft Project (Microsoft Imagine: Номер подписки 700102019, бессрочно). 7. Свободно распространяемое программное обеспечение под лицензиями GNU, Apache, Oracle, Mozilla, CeCILL.</p>
<p>Учебная аудитория для проведения практической занятий, лабораторных работ и самостоятельной работы №206-4</p>	<p>18 ПК с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду: ЦП: Pentium 4 class 2800 ОЗУ: 1 Гб ПЗУ: 50 Гб (8 шт.) ЦП: Intel Pentium II/III class 2327 ОЗУ: 2 Гб ПЗУ: 50 Гб (10 шт.)</p>	<p>1. Операционная система Windows XP/Vista/7/8/10 (Microsoft Imagine: Номер подписки 700102019, бессрочно). 2. Microsoft Visual Studio (Microsoft Imagine: Номер подписки 700102019, бессрочно). 3. Microsoft Office Access (Microsoft Imagine: Номер подписки 700102019, бессрочно). 4. Microsoft Office Visio (Microsoft Imagine: Номер подписки 700102019, бессрочно). 5. Microsoft SQL Server (Microsoft Imagine: Номер подписки 700102019, бессрочно). 6. Microsoft Project (Microsoft Imagine: Номер подписки 700102019, бессрочно). 7. Свободно распространяемое программное обеспечение под лицензиями GNU, Apache, Oracle, Mozilla, CeCILL.</p>
<p>Учебная аудитория для проведения практической занятий, лабораторных работ и самостоятельной работы №206-5</p>	<p>24 ПК с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду: ЦП: Intel Pentium II/III class 2394 ОЗУ: 2 Гб ПЗУ: 70 Гб (17 шт.) ЦП: Intel Pentium II/III class 2327 ОЗУ: 2 Гб ПЗУ: 100 Гб (1 шт.) ЦП: Intel Pentium III Xeon 3093 ОЗУ: 4 Гб ПЗУ: 300 Гб (6 шт.)</p>	<p>1. Операционная система Windows XP/Vista/7/8/10 (Microsoft Imagine: Номер подписки 700102019, бессрочно). 2. Microsoft Visual Studio (Microsoft Imagine: Номер подписки 700102019, бессрочно). 3. Microsoft Office Access (Microsoft Imagine: Номер подписки 700102019, бессрочно). 4. Microsoft Office Visio (Microsoft Imagine: Номер подписки 700102019, бессрочно). 5. Microsoft SQL Server (Microsoft Imagine: Номер подписки 700102019, бессрочно). 6. Microsoft Project (Microsoft Imagine: Номер подписки 700102019, бессрочно). 7. Свободно распространяемое программное обеспечение под лицензиями GNU, Apache, Oracle, Mozilla, CeCILL.</p>

