

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:

ВПиМ  Г.В. Овечкин

« 26 » 06 2020 г.

РАБОЧАЯ ПРОГРАММА

дисциплины

Б1.В.15 «Функциональное программирование»

Направление подготовки — 09.03.04 «Программная инженерия»

Уровень подготовки — академический бакалавриат

Направленность (профиль) подготовки
Программная инженерия

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
(дата утверждения ФГОС ВО)

Разработчик:
доцент кафедры ВПМ


_____ Н.И. Парфилова

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 формирование у будущих специалистов твердых теоретических знаний и практических навыков по технологии функционального и рекурсивного программирования для решения задач представления и использования знаний в системах искусственного интеллекта, освоение языка функционального программирования *Лисп*, приобретение навыков составления и отладки функциональных программ.

Основные задачи освоения учебной дисциплины:

1. Получение системы базовых знаний о функциональном программировании как об одном из подходов в области декларативного программирования.
2. Формирование представления о существующих *Лисп*-системах и их использовании в решении различных задач.
3. Систематизация и закрепление практических навыков и умений по программированию на языке *Лисп*.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина относится к вариативной части профессиональных дисциплин блока № 1 (Б1.В.15), изучается по очной форме обучения на 3 курсе в 5 семестре, базируется на знаниях, полученных в ходе изучения дисциплин: «Математическая логика и теория алгоритмов», «Дискретная математика», «Алгоритмические языки и программирование», «Алгоритмы и структуры данных».

Пререквизиты дисциплины. Для освоения дисциплины обучающиеся должны

- *знать:*
 - основные понятия теории алгоритмов и дискретной математики;
 - управляющие конструкции и структуры данных императивного программирования;
- *уметь:*
 - применять свои знания при решении различных предметных задач;
 - работать в средах программирования;
 - разрабатывать, отлаживать и выполнять императивные программы;
- *владеть:*
 - навыками разработки и отладки императивных программ;
 - методами и приемами анализа и структурирования сложных программ.

Постреквизиты дисциплины. Полученные знания используются при изучении дисциплин «Web-программирование», «Логическое программирование», «Разработка компиляторов», «Разработка системных утилит», при выполнении НИР и при подготовке выпускной квалификационной работы, а также при обучении в магистратуре в таких дисциплинах как, например, «Системы искусственного интеллекта», «Онтология знаний».

В результате изучения дисциплины студенты должны

- *знать:*
 - основы функционального и рекурсивного программирования, основные приемы разработки и отладки функциональных программ;
 - перспективы и тенденции развития функционального программирования;
- *уметь:*
 - применить свои знания по функциональному и рекурсивному программированию к решению предметных задач, в частности, задач представления и использования знаний в системах искусственного интеллекта;
 - работать в среде функционального программирования;
 - составлять, отлаживать и выполнять функциональные программы;
- *владеть:*
 - навыками разработки и отладки функциональных программ;
 - методами и приемами анализа и структурирования сложных программных продуктов (ПП) в стиле декларативного программирования.

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; техническое сопровождение ИС в процессе эксплуатации; применение Web-технологий при реализации удаленного доступа в системах клиент-сервер и распределенных вычислений	Информационные технологии Программное обеспечение	ПК-10. Владение навыками использования различных технологий разработки ПО	ИД – 1 _{ПК-10} <i>Знать:</i> современные технологии разработки ПО. ИД – 2 _{ПК-10} <i>Уметь:</i> использовать современные технологии разработки ПО. ИД – 3 _{ПК-10} <i>Владеть:</i> навыками использования современных технологий разработки ПО.	06.001 Программист 06.004 Специалист по тестированию в области информационных 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 3 зачетных единиц (ЗЕ), 108 академических часов.

Объем дисциплины	Всего часов	Семестр 5
Общая трудоемкость дисциплины, в том числе:	108	108
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	51	51
3. Курсовой проект	-	-
4. Контроль	8,75	8,75
Вид промежуточной аттестации		Зачет

4.2. Содержание дисциплины, структурированное по разделам и темам

В структурном отношении программа дисциплины представлена следующими разделами и темами:

I раздел. История и перспективы развития функционального программирования.

Тема 1. Императивное и функциональное программирование. Сравнительный анализ основ императивного и функционального стилей программирования. Функциональные эквиваленты императивных программ. Строго функциональный язык.

Тема 2. История возникновения и развития функционального программирования. Работы Дж. Маккарти, язык Лисп и его последующие диалекты. Существующие языки и системы функционального программирования. Общая характеристика. Аппаратное обеспечение функциональных программ. Лисп-машины. Сферы приложения Лисп-систем.

II раздел. Основы функционального программирования.

Тема 1. Теоретические основы функционального программирования. Понятие функции. Определение и вызов функции. Лямбда-исчисление А. Чёрча. Префиксная нотация. Диалог с интерпретатором Лиспа. Исполнение функциональных программ. Иерархия вызовов.

Тема 2. Символьные выражения. S-выражения как форма представления программ и данных. Атомы и списки. Числовые и символьные атомы, их назначение. Внутреннее представление атомов. Списки. Представление списков в программе и памяти. Точечная нотация.

Тема 3. Значение S-выражения. Значение атома. Функции Set и Setq. Значение функции. Вычисление значения S-выражения. Функция Eval. Блокирование механизма вычисления значения S-выражения – функция Quote. Работа интерпретатора Лиспа. Распределение памяти, управление памятью и сборка мусора.

III раздел. Базовые функции.

Тема 1. Основные функции обработки списков. Селекторы и конструкторы обработки списков. Предикаты идентификации и сравнения атомов и списков. Логические функции. Арифметические функции и предикаты. Представление в теоретическом Лиспе и в существующих диалектах.

Тема 2. Условное выражение. Понятие условного выражения как альтернативы структуре ветвления. Базовые функции и их реализация в существующих диалектах.

IV раздел. Определение новых функций.

Встроенные и определяемые функции. Средства определения новых функций. Функция *Defun*. Тип определяемой функции. Вычисляемые и невычисляемые функции. *Лямбда*-выражение – его роль и использование в языке. Задание параметров в *Лямбда*-списке. Передача параметров и область их действия. Статические и динамические *Лисп* системы.

V раздел. Основы рекурсивного программирования.

Тема 1. Понятие рекурсии. Теория рекурсивных функций. Использование рекурсии в императивных и функциональных программах. Различные формы рекурсии. Простая рекурсия. Параллельная рекурсия. Взаимная рекурсия. Рекурсия более высокого порядка.

Тема 2. Техника построения рекурсивных программ. Выбор подфункций. Накапливающие параметры. Ассоциативные списки. Использование функций более высокого порядка. Применяющие и отображающие функционалы. Отладка рекурсивных программ. Встроенные средства трассировки.

VI раздел. Средства императивного программирования в *Лисп*-системах.

Основные средства императивного программирования и их реализация в *Лисп*-системах. Понятие *PROG*-выражения. Основные управляющие структуры: последовательные вычисления, ветвления, циклы и их реализация в различных диалектах *Лиспа*. Использование средств императивного программирования в *Лисп*-программах.

VII раздел. Ввод и вывод данных в *Лисп*-программах. Работа с файлами.

Тема 1. Функции ввода и вывода данных. Ввод *S*-выражений. Функции ввода. Вывод значений *S*-выражений. Функции вывода. Управление выводом.

Тема 2. Работа с файлами. Функции работы с файлами. Управление входным и выходным потоками.

VIII раздел. Области приложения *Лисп*-систем.

Искусственный интеллект и технология знаний. Экспертные системы. Системы аналитических преобразований. Доказательства и логическое программирование. Процедуры синтаксического анализа. Программы перевода. Обучающие программы. Задачи поиска. Программирование игр. Распознавание образов и обработка изображений.

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

№ п/п	Тема	Общая трудо- емкость, всего часов	Контактная работа обучающихся с преподавателем				Само- стоя- тельная работа обучаю- щихся
			всего	лекции	лабора- торные работы	прак- тичес кие заня- тия	
1	2	3	4	5	6	7	8
1.	<i>История и перспективы развития функционального программирования</i>	8	2	2			6
1.1.	Императивное и функциональное программирование	4	1	1			3
1.2.	История возникновения и развития функционального программирования	4	1	1			3
2.	<i>Основы функционального программирования</i>	16	10	6		4	6
2.1.	Теоретические основы	4	2	2			2
2.2.	Символьные выражения	6	4	2		2	2
2.3.	Значение S-выражения	6	4	2		2	2
3.	<i>Базовые функции Лиспа</i>	12	8	4	2	2	4
3.1.	Функции обработки списков	6	4	2		2	2
3.2.	Условное выражение	6	4	2	2		2
4.	<i>Определение новых функций</i>	11	6	2	2	2	5
5.	<i>Основы рекурсивного программирования</i>	25	10	4	6		15
5.1.	Понятие рекурсии	9	4	2	2		5
5.2.	Техника построения рекурсивных программ	16	6	2	4		10
6.	<i>Средства императивного программирования в Лисп-системах</i>	9	4	2	2		5
7.	<i>Ввод и вывод данных</i>	11	6	2	4		5
7.1.	Функции ввода и вывода	5	3	1	2		2
7.2.	Работа с файлами	6	3	1	2		3
8.	<i>Области приложения Лисп-систем</i>	7	2	2			5
	<i>Зачет</i>	9					9
	Всего:	108	48	24	16	8	60

4.3.1. Лекционные занятия

№ п/п	Темы лекционных занятий	Трудоемкость (час.)	Формируемые компетенции	Форма контроля
1.	Императивное и функциональное программирование. История и перспективы развития функционального программирования	2	ПК-4, ПК-10	зачет
2.	Теоретические основы функционального программирования	2	ПК-4, ПК-10	зачет
3.	Символьные выражения: <i>S</i> -выражения	2	ПК-4, ПК-10	зачет
4.	Значение <i>S</i> -выражения	2	ПК-4, ПК-10	зачет
5.	Базовые функции <i>Лиспа</i> . Функции обработки списков	2	ПК-4, ПК-10	зачет
6.	Условное выражение	2	ПК-4, ПК-10	зачет
7.	Определение новых функций	2	ПК-4, ПК-10	зачет
8.	Понятие рекурсии	2	ПК-4, ПК-10	зачет
9.	Техника построения рекурсивных программ	2	ПК-4, ПК-10	зачет
10.	Средства императивного программирования в <i>Лисп</i> -системах	2	ПК-4, ПК-10	зачет
11.	Ввод и вывод данных. Функции ввода и вывода. Работа с файлами	2	ПК-4, ПК-10	зачет
12.	Области приложения <i>Лисп</i> -систем	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.	Основы функционального программирования. Символьные выражения	2	ПК-4, ПК-10	зачет
2.	Основы функционального программирования. Значение <i>S</i> -выражения	2	ПК-4, ПК-10	зачет
3.	Базовые функции <i>Лиспа</i> . Функции обработки списков	2	ПК-4, ПК-10	зачет
4.	Определение новых функций	2	ПК-4, ПК-10	зачет

4.3.3. Лабораторные занятия

Лабораторные работы по дисциплине «Функциональное программирование» для направления 09.03.04 – «Программная инженерия» имеют своей целью изучение технологии основных методов функционального и рекурсивного программирования, освоение языка функционального программирования *Лисп*, приобретение навыков разработки и отладки функциональных программ.

В результате выполнения лабораторного практикума студенты должны уметь:

- работать в среде программирования *Лисп*;
- составлять, отлаживать и выполнять функциональные программы средней сложности;
- применить свои знания по функциональному и рекурсивному программированию к решению предметных задач, в частности, задач представления и использования знаний в системах искусственного интеллекта.

№ п/п	Наименование лабораторных работ	Трудоемкость (час.)	Формируемые компетенции	Форма контроля
1.	Простые функциональные программы. Определение новых функций. Условные выражения	4	ПК-4, ПК-10	зачет
2.	Простая рекурсия в задачах обработки списков. Параллельная рекурсия в задачах обработки списков	2	ПК-4, ПК-10	зачет
3.	Разработка функциональных программ с использованием различных форм рекурсии	4	ПК-4, ПК-10	зачет
4.	Разработка функциональных программ с использованием <i>PROG</i> -выражений	2	ПК-4, ПК-10	зачет
5.	Ввод и вывод данных.	2	ПК-4, ПК-10	зачет
6.	Разработка и отладка сложных функциональных программ при решении предметных задач. Работа с файлами.	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	2	3	4	5
1.	Императивное и функциональное программирование. История и перспективы развития функционального программирования	6	ПК-4, ПК-10	зачет
2.	Теоретические основы функционального программирования	2	ПК-4, ПК-10	зачет
3.	Символьные выражения	2	ПК-4, ПК-10	зачет
4.	Значение <i>S</i> -выражения	2	ПК-4, ПК-10	зачет
5.	Базовые функции <i>Лиспа</i> . Функции обработки списков	2	ПК-4, ПК-10	зачет

1	2	3	4	5
10.	Условное выражение	2	ПК-4, ПК-10	зачет
	Определение новых функций	5	ПК-4, ПК-10	зачет
	Основы рекурсивного программирования	5	ПК-4, ПК-10	зачет
	Техника построения рекурсивных программ	10	ПК-4, ПК-10	зачет
	Средства императивного программирования в <i>Лисп</i> -системах	5	ПК-4, ПК-10	зачет
	Ввод и вывод данных. Функции ввода и вывода	2	ПК-4, ПК-10	зачет
	Работа с файлами	3	ПК-4, ПК-10	зачет
	Области приложения <i>Лисп</i> -систем	5	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 «Функциональное программирование»).

6. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Основная литература

1. Хювёнен Э., Селлянен И. Мир Лиспа. Т.1: Введение в язык Лисп и функциональное программирование. – М.: Мир, 1990. – 447 с.
2. Засорин С.В. Функциональное программирование: метод. ук. к лаб. работам. – РГРТУ, Рязань, 2011. – 56 с.
3. Городняя Л.В. Введение в программирование на Лиспе [Электронный ресурс] / Л.В. Городняя, Н.А. Березин. – 2-е изд. – Электрон. текстовые данные. – М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. – 134 с. – 2227-8397. – Режим доступа: <http://www.iprbookshop.ru/73668.html>
4. Рогозин О.В. Функциональное и рекурсивно-логическое программирование [Электронный ресурс]: учебное пособие / О.В. Рогозин. – Электрон. текстовые данные. – М.: Евразийский открытый институт, 2009. – 139 с. – 978-5-374-00182-2. – Режим доступа: <http://www.iprbookshop.ru/11119.html>
5. Практикум по дисциплине логическое и функциональное программирование [Электронный ресурс] – Электрон. текстовые данные. – М.: Московский технический университет связи и информатики, 2016. – 16 с. – 2227-8397. – Режим доступа: <http://www.iprbookshop.ru/61489.html>

6.2. Дополнительная литература

6. Хювёнен Э., Селлянен И. Мир Лиспа. Т.2: Методы и системы программирования. – М.: Мир, 1990. – 319 с.
7. Салмина Н.Ю. Функциональное программирование и интеллектуальные системы [Электронный ресурс]: учебное пособие / Н.Ю. Салмина. – Электрон. текстовые данные. – Томск: Томский государственный университет систем управления и радиоэлектроники, 2016. – 100 с. – 2227-8397. – Режим доступа: <http://www.iprbookshop.ru/72216.html>

8. Галкина М.Ю. Функциональное и логическое программирование [Электронный ресурс]: практикум / М.Ю. Галкина. – Электрон. текстовые данные. – Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2008. – 107 с. – 2227-8397. – Режим доступа: <http://www.iprbookshop.ru/55464.html>

9. Городняя Л.В. История развития программного обеспечения. Первые реализации языка Lisp в СССР. – www.computer-museum.ru/histsoft/lisp_sorucum_2011.htm

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 рекомендуемому 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 рекомендуемому 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, работать по итеративной схеме, т.е. вырабатывают в себе привычку думать обо всех деталях, необходимых для разработки больших программ и информационных систем. При этом программирование остается основополагающим навыком, чем раньше и дольше студенты будут программировать, тем лучшими специалистами они смогут стать в дальнейшем. Кроме того, 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. HomeLisp 13.4.1, HomeLisp 1.11.1, HomeLisp 1.13.4 (Freeware) – свободно распространяемое ПО (сайт <http://homtelisp.ru>).

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.) ЦП: Intel Pentium 4 class 3200	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 Imag-

	<p>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>ine: 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>