

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

Кафедра вычислительной и прикладной математики

СОГЛАСОВАНО

УТВЕРЖДАЮ

Декан ФАИТУ

С.И. Холопов
« 25 » 06 2020 г.

Проректор РОПиМД



А.В. Корячко
« 06 » 2020 г.

Заведующий кафедрой АСУ

С.И. Холопов
« 25 » 06 2020 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.О.01.18 «Алгоритмические языки и программирование»

Направление подготовки

09.03.02 «Информационные системы и технологии»

Уровень подготовки – академический бакалавриат

Квалификация выпускника – бакалавр

Формы обучения – очная, заочная


Рязань 2020 г.

ЛИСТ СОГЛАСОВАНИЯ


Рабочая программа составлена с учетом требований федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.02 «Информационные системы и технологии», утвержденного приказом Минобрнауки России от 19.09.2017 г. № 926.

Разработчики:

доцент кафедры ВПМ

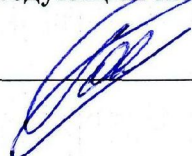

_____ Н.И. Парфилова
(подпись)

старший преподаватель кафедры ВПМ


_____ О.А. Москвитина
(подпись)

Рассмотрена и утверждена на заседании кафедры 11 июня 2020 г., протокол № 11.

Заведующий кафедрой ВПМ


_____ Г.В. Овечкин

1. ЦЕЛЬ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель изучения дисциплины – обучение студентов основам алгоритмизации и программирования предметных задач на современных ЭВМ, что, в частности, включает:

- формирование логического и алгоритмического мышления, культуры алгоритмизации и структурного программирования;
- формирование теоретической базы и практических умений и навыков разработки алгоритмов и программ решения различных предметных задач на ЭВМ в императивных системах программирования.

Основная задача дисциплины – обеспечить базовое образование студентов основам алгоритмизации и структурного программирования, дающее возможность их дальнейшего обучения в области применения вычислительной техники при проектировании, разработке, внедрять выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности и эксплуатации современных программных средств и информационных систем различного назначения, при создании и использовании новых информационных технологий.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина относится к обязательной части Блока 1 «Дисциплины (модули)» основной профессиональной образовательной программы (далее – образовательной программы) бакалавриата направления подготовки 09.03.02 «Информационные системы и технологии», изучается по очной и заочной форме обучения на 1 курсе в 1 и 2 семестрах.

Для успешного освоения дисциплины студент должен иметь компетенции, формирование которых предусмотрено программой среднего (полного) общего образования или среднего профессионального образования.

Пререквизиты дисциплины. Для освоения дисциплины обучающиеся должны

- *знать:*
 - основные понятия информатики и математики;
 - первичные основы алгоритмизации и программирования;
- *уметь:*
 - применять свои знания при решении типовых предметных задач;
 - иметь навык работы в среде программирования;
 - разрабатывать, отлаживать и выполнять простейшие программы;
- *владеть:*
 - навыками анализа предметных задач вычислительного характера,
 - навыками разработки и отладки простейших программ;
 - методами и приемами анализа и структурирования программ.

Постреквизиты дисциплины. Освоение данной дисциплины необходимо как предшествующее для большинства дисциплин профессионального цикла, и, в первую очередь, для дисциплин, связанных с освоением различных технологий и языков программирования, например, «Объектно-ориентированное программирование», «Технологии программирования», «Архитектура информационных систем», «Разработка и анализ требований», «Программное обеспечение информационных систем», а также таких дисциплин как «Компьютерная графика», «Операционные системы», «Базы данных», «Сети и телекоммуникации», «Защита информации» и других специальных дисциплин, требующих навыков программирования.

В результате изучения дисциплины студенты должны

- *знать:*
 - современные информационные технологии и программные средства, в том числе отечественного производства;
 - структуру современного программного обеспечения;

- основные парадигмы программирования и номенклатуру языков программирования различного уровня и их возможности;
- основы императивного стиля программирования;
- основы структурного проектирования алгоритмов и программ;
- теоретические основы алгоритмизации и программирования типовых задач на конкретном алгоритмическом языке;
- перспективы и тенденции развития информационных технологий и вычислительной техники;
- приемы алгоритмизации и программирования задач профессиональной деятельности;
- методы автономной отладки и тестирования программ;
- *уметь*:
 - работать в современной среде программирования;
 - использовать современные информационные технологии и инструментальные средства для решения различных задач в своей профессиональной деятельности;
 - разрабатывать алгоритмы и программы в соответствии с принципами структурного программирования, анализировать структуру алгоритмов и программ;
 - выполнять алгоритмизацию и программирование типовых инженерных расчетов с применением различных вычислительных методов и современных технологий программирования;
 - выполнять тестирование и отладку программ; оформлять программную документацию;
- *владеть*:
 - основными методами, способами и средствами обработки информации на ЭВМ;
 - навыками работы с конкретными программными системами;
 - навыками проектирования, разработки, отладки и тестирования программ в современных средах программирования императивного стиля;
 - навыками работы с научно-технической литературой и технической документацией.

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.

Общепрофессиональные компетенции выпускников и индикаторы их достижения

Категория (группа) общепрофессиональных компетенций	Код и наименование общепрофессиональной компетенции	Код и наименование индикатора достижения общепрофессиональной компетенции
	ОПК-2. Способен использовать современные информационные технологии и программные средства, в том числе, отечественного производства, при решении задач профессиональной деятельности	ИД – 1 _{ОПК-2} <i>Знать</i> : современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности; ИД – 2 _{ОПК-2}

		<p><i>Уметь:</i> выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности;</p> <p>ИД – 3_{ОПК-2}</p> <p><i>Владеть:</i> навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности.</p>
--	--	--

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 5 зачетных единиц (ЗЕ), 180 академических часов.

Вид учебной работы	Всего часов	Семестры	
		1	2
Аудиторные занятия (всего)	64	32	32
В том числе:			
Лекции	32	16	16
Лабораторные работы (ЛР)	32	16	16
Практические занятия (ПЗ)			
Семинары (С)			
Курсовой проект/работа (аудиторная нагрузка)			
<i>Другие виды аудиторной работы</i>			
Самостоятельная работа (всего)	80	31	49
В том числе:			
Курсовая работа (КР)			12
Расчетно-графические работы			
Расчетные задания			
Реферат			
<i>Другие виды самостоятельной работы</i>		31	37
Контроль	36	9	27
Вид промежуточной аттестации (зачет, дифференцированный зачет, экзамен)	экзамен, зачет, КР	зачет	экзамен, КР
Общая трудоемкость (часы)	180	72	108
Зачетные единицы трудоемкости	5	2	3
Контактная работа (по учебным занятиям)	64	32	32

Общая трудоемкость изучения дисциплины для заочной формы обучения составляет 5 зачетных единиц (ЗЕ), 180 академических часов.

Вид учебной работы	Часы
Установочная сессия	
Аудиторные занятия (всего)	8
В том числе:	
Лекции	4
Лабораторные работы (ЛР)	4
Практические занятия (ПЗ)	
Самостоятельная работа (всего)	28
В том числе:	
Контрольные работы	28
Вид промежуточной аттестации (зачет)	-
Зимняя сессия	
Аудиторные занятия (всего)	8
В том числе:	
Лекции	4
Лабораторные работы (ЛР)	4
Практические занятия (ПЗ)	
Самостоятельная работа (всего)	60
В том числе:	
Контрольные работы	60
Вид промежуточной аттестации (проверка к.р.)	4
Вид промежуточной аттестации (зачет)	зачет
Летняя сессия	
Аудиторные занятия (всего)	
В том числе:	
Лекции	
Лабораторные работы (ЛР)	
Практические занятия (ПЗ)	
Самостоятельная работа (всего)	63
В том числе:	
Курсовая работа (КР)	12
Вид промежуточной аттестации (проверка КР)	9
Вид промежуточной аттестации (экзамен)	экзамен
Общая трудоемкость, час	180
Общее количество зачетных единиц трудоемкости	5

4.2. Содержание дисциплины, структурированное по разделам и темам

В структурном отношении программа дисциплины представлена следующими разделами и темами:

I раздел. Создание алгоритмов и программ. Теория алгоритмизации.

Тема 1. Решение предметной задачи на ЭВМ. Этапы решения задачи на ЭВМ. Алгоритм и его свойства. Алгоритмическая система. Алгоритмизация – процесс разработки и описания алгоритма. Средства записи алгоритмов.

Тема 2. Технология разработки алгоритмов. Структурный подход. Основные принципы. Теорема о структурировании. Базовые управляющие структуры. Проектирование алгоритмов: нисходящее и восходящее проектирование.

Тема 3. Структуры алгоритмов. Алгоритмы линейной структуры. Ветвления. Циклы. Циклы с пред- и пост-условием. Циклы с параметром. Итерационные циклы. Вложенные циклы. Вспомогательные/подчиненные алгоритмы. Данные. Простые данные. Константы и переменные. Структуры данных.

II раздел. Технология структурного программирования.

Тема 1. Парадигмы и языки программирования. Эволюция языков программирования. Методы и стили программирования. Императивный стиль программирования. Структурное программирование. Программирование алгоритмов различной структуры. Реализация базовых управляющих структур в алгоритмических языках.

Тема 2. Программирование алгоритмов линейной структуры.

Тема 3. Программирование алгоритмов с ветвлениями. Операторы, реализующие простое (дихотомическое) и многоальтернативное ветвление.

Тема 4. Программирование циклических алгоритмов. Операторы цикла.

Тема 5. Итерационные циклы – вычисления с заданной точностью.

Тема 6. Вложенные циклы.

Тема 7. Организация ввода-вывода данных. Процедуры ввода и вывода данных из стандартных файлов. Управление выводом.

Тема 8. Отладка программ. Способы и средства их отладки в интегрированной среде.

III раздел. Типы и структуры данных.

Тема 1. Концепция действий и данных в программировании. Понятие типа. Понятие типизированного языка программирования. Иерархия типов. Скалярные и структурированные типы. Понятие статических и динамических данных. Ссылочный тип. Динамические структуры данных.

Тема 2. Регулярный тип. Массивы. Одномерные и двумерные массивы.

Тема 3. Символьный и строковый типы. Строки. Обработка текстов.

Тема 4. Комбинированный тип. Записи. Табличное представление данных. Простые и вложенные записи. Записи с вариантами.

Тема 5. Множественный тип. Множества. Представление и использование множеств в Паскале.

Тема 6. Файловый тип. Файлы.

IV раздел. Организация блочно-модульных программ.

Тема 1. Блочный принцип организации программ. Подпрограммы. Процедуры и функции. Процедуры. Описание и использование. Функции. Описание и использование.

Тема 2. Параметры процедур и функций. Параметры-значения и параметры-переменные, параметры константы. Процедурный тип.

Тема 3. Понятие рекурсии. Программирование рекурсивных функций.

Тема 4. Понятие модуля. Создание и использование модулей. Основные стандартные модули в Паскале.

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

Для очной формы обучения

№ 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
Семестр 1							
1.	<i>Теория алгоритмизации</i>	3					3
2.	<i>Технология структурного программирования</i>	60	32	16	16		28
2.1	Парадигмы и языки программирования. Методы и стили программирования. Императивный стиль. Структурное программирование	8	2	2			6
2.2	Программирование линейных алгоритмов	10	4	2	2		6
2.3	Программирование алгоритмов с ветвлениями	12	6	2	4		6
2.4	Программирование циклических алгоритмов	18	12	6	6		6
2.5	Организация ввода-вывода	6	4	2	2		2
2.6	Отладка программ	6	4	2	2		2
	Консультации в семестре						
	Зачет и консультации	9					
	Итого в семестре:	72	32	16	16		31
Семестр 2							
3.	<i>Типы и структуры данных</i>	42	20	10	10		22
3.1	Концепция данных и концепция типа. Иерархия типов. Скалярные и структурированные типы	5	2	2			3
3.2	Регулярный тип. Массивы	8	4	2	2		4
3.3	Символьный и строковый типы. Строки	8	4	2	2		4
3.4	Комбинированный тип. Записи	6	3	1	2		3
3.5	Множественный тип. Множества	6	3	1	2		3
3.6	Файловый тип. Файлы	9	4	2	2		5
4.	<i>Организация блочно-модульных программ</i>	39	12	6	6		27
4.1	Блочный принцип организации	9	4	2	2		5

	программ. Подпрограммы. Процедуры и функции						
1	2	3	4	5	6	7	8
4.2	Параметры процедур и функций.	12	4	2	2		8
4.3	Программирование рекурсивных функций	6					6
4.4	Понятие модуля. Создание и использование модулей. Стандартные модули	12	4	2	2		8
	Консультации в семестре						
	Экзамены и консультации	27					
	Итого в семестре:	108	32	16	16		49
	Всего:	180	64	32	16		80

Для заочной формы обучения

№ п/п	Тема	Общая трудоемкость, всего часов	Контактная работа обучающихся с преподавателем			Самостоятельная работа обучающихся
			всего	лекции	лабораторные работы	
Семестр 1. Установочная сессия						
1	Методы и стили программирования. Императивный стиль. Структурное программирование. Линейные и разветвляющиеся алгоритмы и программы	18	4	2	2	14
2	Циклические алгоритмы и программы. Организация ввода-вывода	18	4	2	2	14
	Самостоятельная работа					28
	Итого:	36	8	4	4	28
Зимняя сессия						
3	Сложные типы и структуры данных. Массивы. Строки. Записи. Файлы	34	4	2	2	30
4	Блочный принцип организации программ. Подпрограммы. Процедуры и функции. Модули	34	4	2	2	30
	Самостоятельная работа					60
	Зачет и консультация	4				4
	Итого:	72	8	4	4	64
Семестр 2. Летняя сессия						
	Самостоятельная работа	51				51
	Экзамен, консультация, КР	9+12				9+12
	Итого:	72				72
	Всего:	180	16	8	8	164

4.4. Лекционные занятия

4.4.1. Лекционные занятия для очной формы обучения

№ п/п	Темы лекционных занятий	Трудоемкость (час.)	Формируемые компетенции	Форма контроля
1	Парадигмы и языки программирования. Методы и стили программирования. Императивный стиль. Структурное программирование	2	ОПК-2	зачет
2	Программирование линейных алгоритмов	2	ОПК-2	зачет
3	Программирование алгоритмов с ветвлениями	2	ОПК-2	зачет
4	Программирование циклических алгоритмов	6	ОПК-2	зачет
5	Организация ввода-вывода	2	ОПК-2	зачет
6	Отладка программ	2	ОПК-2	зачет
7	Концепция данных и концепция типа. Иерархия типов. Скалярные и структурированные типы	2	ОПК-2	экзамен
8	Регулярный тип. Массивы	2	ОПК-2	экзамен
9	Символьный и строковый типы. Строки	2	ОПК-2	экзамен
10	Комбинированный тип. Записи	1	ОПК-2	экзамен
11	Множественный тип. Множества	1	ОПК-2	экзамен
12	Файловый тип. Файлы	2	ОПК-2	экзамен
13	Блочный принцип организации программ. Подпрограммы. Процедуры и функции	2	ОПК-2	экзамен
14	Параметры процедур и функций	2	ОПК-2	экзамен
15	Понятие модуля. Создание и использование модулей. Стандартные модули	2	ОПК-2	экзамен

4.4.2. Лекционные занятия для заочной формы обучения

№ п/п	Темы лекционных занятий	Трудоемкость (час.)	Формируемые компетенции	Форма контроля
1	Методы и стили программирования. Императивный стиль. Структурное программирование. Линейные и разветвляющиеся алгоритмы и программы	2	ОПК-2	зачет
2	Циклические алгоритмы и программы. Организация ввода-вывода	2	ОПК-2	зачет
3	Сложные типы и структуры данных. Массивы. Строки. Записи. Файлы	2	ОПК-2	экзамен
4	Блочный принцип организации программ. Подпрограммы. Процедуры и функции. Модули	2	ОПК-2	экзамен

4.5. Лабораторные занятия

Лабораторные работы по дисциплине «Алгоритмические языки и программирование» имеют своей целью изучение технологии, основных методов и алгоритмов структурного программирования, освоение алгоритмического языка структурного программирования, приобретение навыков составления и отладки программ.

В результате выполнения предусмотренного лабораторного практикума студенты должны уметь:

- работать в современной среде программирования;
- составлять, отлаживать и выполнять программы средней сложности;
- применить свои знания по структурному программированию к решению различных прикладных задач.

4.5.1. Лабораторные работы для очной формы обучения

№ п/п	Наименование лабораторных работ	Трудоемкость (час.)	Формируемые компетенции	Форма контроля
1	Программирование линейных алгоритмов	2	ОПК-2	зачет
2	Программирование алгоритмов с ветвлениями	4	ОПК-2	зачет
3	Программирование циклических алгоритмов	6	ОПК-2	зачет
4	Организация ввода-вывода	2	ОПК-2	зачет
5	Отладка программ	2	ОПК-2	зачет
6	Регулярный тип. Массивы	2	ОПК-2	экзамен
7	Символьный и строковый типы. Строки	2	ОПК-2	экзамен
8	Комбинированный тип. Записи	2	ОПК-2	экзамен
9	Множественный тип. Множества	2	ОПК-2	экзамен
10	Файловый тип. Файлы	2	ОПК-2	экзамен
11	Подпрограммы. Процедуры и функции	2	ОПК-2	экзамен
12	Параметры процедур и функций	2	ОПК-2	экзамен
13	Создание и использование модулей. Стандартные модули	2	ОПК-2	экзамен

4.5.2. Лабораторные работы для заочной формы обучения

№ п/п	Наименование лабораторных работ	Трудоемкость (час.)	Формируемые компетенции	Форма контроля
1	Линейные и разветвляющиеся алгоритмы и программы	2	ОПК-2	зачет
2	Циклические алгоритмы и программы. Организация ввода-вывода	2	ОПК-2	зачет
3	Массивы. Строки. Записи. Файлы	2	ОПК-2	экзамен
4	Подпрограммы. Процедуры и функции. Модули	2	ОПК-2	экзамен

4.6. 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 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 выполнении курсового проектирования в изучении необходимого теоретического материала, в проектировании, разработке, отладке и выполнении программы своего задания, подготовке отчета и подготовке к защите курсовой работы;
- при самостоятельном изучении отдельных вопросов и тем курса с применением рекомендуемой учебно-методической литературы;
- при подготовке к зачету и экзамену в изучении, осмыслении и повторении пройденного теоретического материала и выполненных практических заданий с применением конспекта лекций и учебно-методической литературы.

4.6.1. Самостоятельная работа для очной формы обучения

№ п/п	Тематика самостоятельной работы	Трудоемкость (час.)	Формируемые компетенции	Форма контроля
1	Теория алгоритмизации	3	ОПК-2	зачет
2	Парадигмы, языки, методы и стили программирования. Императивный стиль. Структурное программирование	6	ОПК-2	зачет
3	Линейные алгоритмы и программы	6	ОПК-2	зачет
4	Алгоритмы и программы с ветвлениями	6	ОПК-2	зачет
5	Циклические алгоритмы и программы	6	ОПК-2	зачет
6	Организация ввода-вывода	2	ОПК-2	зачет
7	Отладка программ	2	ОПК-2	зачет
8	Концепция данных и концепция типа. Скалярные и структурированные типы	3	ОПК-2	экзамен
9	Регулярный тип. Массивы	4	ОПК-2	экзамен
10	Символьный и строковый типы. Строки	4	ОПК-2	экзамен
11	Комбинированный тип. Записи	3	ОПК-2	экзамен
12	Множественный тип. Множества	3	ОПК-2	экзамен
13	Файловый тип. Файлы	5	ОПК-2	экзамен
14	Подпрограммы. Процедуры и функции	5	ОПК-2	экзамен
15	Параметры процедур и функций.	8	ОПК-2	экзамен
16	Рекурсивные функции	6	ОПК-2	экзамен
17	Создание и использование модулей. Стандартные модули	8	ОПК-2	экзамен
18	Экзамены и консультации	27	ОПК-2	экзамен

4.6.2. Самостоятельная работа для заочной формы обучения

№ п/п	Тематика самостоятельной работы	Трудоемкость (час.)	Формируемые компетенции	Форма контроля
1	Методы и стили программирования. Императивный стиль. Структурное программирование. Линейные и разветвляющиеся алгоритмы и программы	14	ОПК-2	зачет
2	Циклические алгоритмы и программы. Организация ввода-вывода	14	ОПК-2	зачет
3	Зачет и консультация	4	ОПК-2	зачет
4	Сложные типы и структуры данных. Массивы. Строки. Записи. Файлы	30	ОПК-2	экзамен
5	Блочный принцип организации программ. Подпрограммы. Процедуры и функции. Модули	30	ОПК-2	экзамен
6	Экзамен, консультация, КР	72	ОПК-2	экзамен

4.7. Темы курсовых работ

Целью курсового проектирования по дисциплине «Алгоритмические языки и программирование» является углубление теоретических знаний студентов в области алгоритмизации и программирования, приобретение практических навыков разработки, отладки, тестирования и документирования достаточно сложных программных продуктов.

Исходя из этого, курсовое проектирование по данной дисциплине посвящено проектированию и разработке программ сложной модульной структуры с использованием различных типов и структур данных для различных предметных областей.

Основой к выполнению курсовой работы является лекционные занятия и лабораторный практикум, посвященные изучению студентами технологии структурного программирования и средствам его реализации в определенном алгоритмическом языке.

Тема 1. Проектирование обучающе-контролирующих программ.

При проектировании обучающе-контролирующих программ следует учесть следующие требования.

1. Программа должна иметь систему меню и графический интерфейс пользователя (с использованием графических операторов языка или псевдографики), который позволяет осуществлять удобную навигацию по системе меню с помощью клавиш управления курсором. Допускается использования манипулятора типа «мышь».

2. Программа должна иметь заставку, в которой выводится название программы и имя автора, главное меню с пунктами получения справки по программе и выходом из программы.

3. Программа должна всегда подсказывать пользователю на экране каждого меню, какие возможности и действия доступны в текущий момент времени.

4. Методический материал подбирается студентом по нескольким изучаемым темам, количество которых оговаривается в задании.

5. Программы могут быть двух типов: контролирующие и обучающе-контролирующие, предусматривающие работу в двух режимах: 1 - контроль; 2 - обучение. В режиме контроля на вопрос предусматривается однозначный ответ, фиксируется количество верных и неверных ответов. В режиме обучения в случае неверного ответа предусматривается подсказка – выдача на экран обучающего текста. После этого вопрос задается по данной теме повторно. Вопрос может быть такой же категории сложности либо проще. При этом сложность вопроса должна быть учтена введением весовых коэффициентов. Выбор вопроса по теме осуществляется автоматически, с использованием датчика случайных чисел.

6. Вся база контролирующих вопросов должна храниться в файле, и загружаться во время запуска программы. Все изменения также должны сохраняться в файле.

7. Программа должна содержать учетную информацию для преподавателя: фамилию студента, количество опросов, количество верных и неверных ответов. Весь протокол тестирования студентов должен записываться в файл, с возможностью его просмотра преподавателем (вход в режим преподавателя должен осуществляться по паролю).

8. Для ввода данных надо предусмотреть интерфейс пользователя и систему меню.

9. Важно, чтобы программа контролировала вводимые пользователем данные и не позволяла ему вводить недопустимые значения (например, отрицательные расстояния и массы, неправильный формат дат, слишком большой возраст людей и т.д.)

Задание 1.1. Математика

Разработать обучающе-контролирующую программу по отдельным разделам математики: алгебре, геометрии, тригонометрии и т.д. Количество контролируемых разделов должно быть не менее пяти, в каждом из разделов предусмотреть не менее 10 вопросов. В случае неверного ответа обеспечить анализ ошибки и выдачу различного рода подсказок, например, "Вы, пропустили знак", "Эта формула представляет собой разность квадратов" и т.д. При подсчете общего балла для каждой ошибки ввести весовые коэффициенты. На каждый вопрос предусмотреть возможность повторного ответа.

Задание 1.2. Логические схемы

Разработать программу, контролирующую знание логических схем *AND*, *OR*, *NOT*, *XOR* и т.д. Программа должна воспроизводить на экране рисунок одной из логических схем. Задавая случайным образом различные комбинации входных сигналов, проанализировать правильность вводимого обучающимся значения выходного сигнала. Количество анализируемых логических схем должно быть не менее 30. С целью введения элементов обучения при ошибочном ответе предусмотреть возможность повторного ответа, выдав какую-либо подсказку, например, таблицу истинности анализируемой схемы.

Задание 1.3. Вступительные экзамены

Разработать контролирующую программу, обеспечивающую прием вступительных экзаменов в вуз по математике, физике и информатике. По каждой из дисциплин случайным образом должно задаваться не менее трех вопросов, имеющих различные весовые коэффициенты, отражающие уровень сложности вопросов. После достижения суммы этих коэффициентов определенной величины выставляется оценка по пятибалльной системе. Программа определяет, набрал ли экзаменуемый заданный проходной балл, и, в случае успеха, его фамилия вносится в список студентов. Банк программы должен содержать по каждому предмету не менее 15 вопросов.

Задание 1.4. Транслятор

Разработать программу, производящую синтаксический анализ различных синтаксических конструкций заданного алгоритмического языка. В программе должен быть предусмотрен анализ не менее 50 ошибочных ситуаций, при этом каждая ошибка должна помечаться и сопровождаться пояснительным текстом. Выбор ошибочных ситуаций и комментариев к ним произвести произвольно. Предусмотреть возможность повторного ввода исправлений в конструкциях.

Задание 1.5. Программирование

Разработать обучающе-контролирующую программу по заданному алгоритмическому языку. Программа должна производить обучение и контроль знаний студента не менее чем по пяти темам. По каждой теме студенту выдается обучающий текст и затем случайным образом выбирается для контроля один из пяти вопросов по данной теме. Если ответ обучаемого не совпадает с заданным эталоном, предусмотреть возможность выдачи текста, разъясняющего характер ошибки и дополнительного вопроса. По окончании работы обучаемому

выставляется оценка с учетом числа выданных вопросов и правильных ответов. Банк программы должен содержать по каждой теме не менее 15 вопросов.

Задание 1.6. Начинающий программист

Разработать обучающе-контролирующую программу для изучения графических символов схем алгоритмов. Программа должна выдавать на экран выбранные случайным образом графические символы схем алгоритмов и предлагать ввести оператор, соответствующий данному символу на определенном алгоритмическом языке. В случае ошибочного ответа необходимо реализовать подсказку, анализировать надо не только логику ответа, но и его синтаксис. Оценка рассчитывается в зависимости от количества верных и неверных ответов. Банк программы должен содержать не менее 40 вопросов различной категории сложности.

Задание 1.7. Алгоритмизация

Разработать обучающе-контролирующую программу для изучения графических блок-схем алгоритмов. Программа должна выдавать на экран дисплея графическую схему одного из вычислительных процессов: линейного, разветвляющегося, циклического. По заданной схеме обучающий должен написать программу. Программа анализирует соответствие каждого оператора языка блок-схеме и выдает по результатам анализа соответствующий комментарий. В зависимости от количества и характера допущенных ошибок выставляется оценка. Банк программы должен содержать не менее 40 вопросов различной категории сложности.

Задание 1.8. Справочник

Разработать обучающую программу для изучения основных элементов определенного алгоритмического языка. По введенному ключевому слову или оператору языка программа должна выдавать соответствующий обучающий текст и пример использования данного оператора, сопроводить обучающий текст рисунком графической блок-схемы. Предусмотреть возможность дополнения банка данных, удаления информации и сортировки ключевых слов по алфавиту. Банк программы должен содержать не менее 40 записей.

Задание 1.9. Двоичная арифметика

Разработать обучающе-контролирующую программу для изучения способов перевода десятичных чисел в двоичные и обратно, выполнения арифметических операций в двоичной системе счисления. По каждой теме программа должна выдавать обучающий текст и затем предлагать вопрос для контроля, выбрав его случайным образом из группы вопросов по теме. Если обучаемый допускает ошибку в ответе, то должно быть выдано соответствующее сообщение и предоставлена возможность повторного ответа. В зависимости от количества заданных вопросов и допущенных при ответах ошибок обучаемому выставляется оценка. Банк программы должен содержать не менее 50 вопросов различной категории сложности.

Задание 1.10. Работа с файлами

Разработать обучающую программу, обеспечивающую получение навыков работы с файловой системой. Программа должна эмулировать работу файловой системы со следующим перечнем работ:

- справка по системе и её командам;
- создание текстового файла с заданным именем;
- переименование файла;
- копирование файла;
- удаление файла;
- редактирование созданных файлов;
- объединение файлов;
- создание подкаталогов;
- удаление подкаталогов;
- навигация по каталогам с файлами.

Тема 2. Проектирование баз данных и файловых структур

Целью проектирования данного раздела заданий является создание базы данных или файловой системы в рассматриваемой предметной области. В качестве модели данных принята одна из "классических" моделей – реляционная модель данных. При такой форме организации данных вся информация представляется в табличной форме – значениями данных в системе взаимосвязанных таблиц. При проектировании программ работы с базами данных следует учесть следующие требования.

1. Программа должна иметь систему меню и графический интерфейс пользователя (с использованием графических операторов языка или псевдографики), который позволяет осуществлять удобную навигацию по системе меню с помощью клавиш управления курсором. Допускается использования манипулятора типа «мышь».

2. Предусмотреть заставку, на которой выводится название программы и имя автора, главное меню с пунктами получения справки по программе и выходом из программы.

3. Программа должна всегда подсказывать пользователю на экране каждого меню, какие возможности и действия доступны в текущий момент времени.

4. Вход в систему осуществляется по логину и паролю, все пользователи делятся на два типа – клиент и администратор. Клиент способен только просматривать данные и пользоваться ограниченным функционалом, администратору же доступны все функции системы, включая редактирование и удаление данных. При входе в систему программа сравнивает введенные логин и пароль с таблицей пользователей, которая хранится в отдельном файле. Пример таблицы пользователей:

Логин	Пароль	Администратор
Иванов	Fgdkl2342nk	1
Петров	DFgfDFG34	0
Сидоров	DFffET33d	0
Кузнецов	sfSSFkjsd3	1

5. Программа должна иметь удобный функционал по созданию, редактированию и удалению записей в таблицах (в соответствии с заданием). При каждом удалении данных программа должна предупреждать пользователя о необратимости операции и спрашивать подтверждение действия.

6. Вся база данных должна храниться в отдельном файле, и загружаться во время запуска программы. Все изменения также должны сохраняться в файле.

7. Для ввода данных предусмотреть комфортный интерфейс пользователя и систему меню. Пользователь должен иметь возможность в любой момент прекратить ввод данных, или вернуться к редактированию ранее введенных полей.

8. Важно чтобы программа контролировала вводимые пользователем данные и не позволяла ему вводить недопустимые значения (например, отрицательные расстояния и массы, неправильный формат дат, слишком большой возраст людей и т.д.)

Задание 2.1. Расписание движения поездов

Разработать программу создания и поддержки таблицы расписания движения поездов через некоторую станцию. Столбцы таблицы должны содержать сведения о номере поезда, его типе, о пунктах отправления-прибытия, о времени отправления-прибытия, о промежуточных станциях, доступных классах вагонов, цене за билет. Предусмотреть возможность выдачи информации о конечной и промежуточных станциях, времени прибытия и стоянки, о возможном альтернативном пути на конечную станцию, подборе самого дешёвого билета из пункта А в пункт Б. Предусмотреть возможность добавления и удаления строк таблицы расписания. Необходимо хранить информацию не менее чем о 50-ти рейсах.

Задание 2.2. Диспетчер

Разработать программу, регулирующую загрузку компьютера в учебном классе. В таблицах должны быть указаны тип компьютера, расписание загрузки для учебных занятий и предусмотрена возможность индивидуальной записи на каждое рабочее место. Предусмот-

реть возможность добавления и удаления строк таблиц. По запросу пользователя программа должна выдавать информацию о наличии свободных рабочих мест и статистическую информацию о загрузке компьютера во время учебного процесса. Таблица должна содержать не менее чем 50 записей.

Задание 2.3. База данных

Разработать программу создания, редактирования и удаления таблиц произвольной длины с произвольным числом столбцов. Для каждого столбца задаётся тип данных – текстовый или числовой. Должны быть предусмотрены функции добавления и удаления строк в созданных таблицах, поиск информации по заданному ключу, а также сортировка строк по некоторому атрибуту и суммирование элементов столбцов (если это позволяет сделать тип данных столбца). Программа должна адекватно масштабировать ширину столбцов при отображении таблиц в зависимости от введённых данных. В программе должно быть создано не менее 4 таблиц, с количеством строк не менее 10, и различным числом столбцов от 4 до 10.

Задание 2.4. Метрология

Разработать программу, создающую в диалоговом режиме таблицу "Прибор", содержащую графы с информацией о номере прибора, его типе (вольтметр, амперметр и т.д.), годе выпуска, цене, месте нахождения и дате поверки. Предусмотреть возможность создания, добавления и удаления строк таблицы, сортировки строк по заданному ключу, подсчета итога: цены приборов, количества по типам, количество старых и новых приборов, количество приборов находящихся в каждом месте. Таблица должна содержать не менее чем 50 записей.

Задание 2.5. Анкета

Составить программу, хранящую анкетные данные о студентах группы, такие, например, как фамилия, год и место рождения, средний балл на вступительных экзаменах и сессии, участие в общественной, научной и спортивной работах. Каждый пункт оценивается определенным количеством баллов. Программа должна уметь добавлять, редактировать и удалять записи о студентах, подсчитывать и выводить общий балл для каждого студента, выбирать лучшего и худшего студентов по общему показателю, выдавать информацию по заданному ключу. Таблица должна содержать не менее чем 50 записей.

Задание 2.6. Регистратура

Разработать программу, обеспечивающую регистрацию и учет больных, посещающих поликлинику. В картах больных должна содержаться информация, такая как номер карточки, ФИО больного, дата рождения, ФИО лечащего врача, дата последнего посещения врача, текущая дата. Программа должна обеспечивать поиск информации по любому из заданных ключей, создавать, редактировать и удалять записи. Предусмотреть выдачу информации о больных посетивших данного врача или посетивших врача в определенный день. Таблица должна содержать не менее чем 50 записей.

Задание 2.7. Словарь технических терминов

Разработать программу пояснения терминов, принятых в ВТ. Словарь должен содержать не менее чем 60 терминов. Предусмотреть систему меню, содержащую пункты:

- поиск по ключу (сокращению, аббревиатуре) полной расшифровки понятия;
- пояснение понятия;
- область применения;
- поиск синонимов;
- вывод логически связанных терминов;
- расширение, редактирование, сокращение словаря;
- сортировка наименований понятий по алфавиту.

Задание 2.8. Англо-русский словарь

Разработать программу, обеспечивающую выдачу перевода английских слов на русский язык. Предусмотреть возможность расширения, редактирования, сокращения словаря, поиска синонимов, английских и русских слов, определения части речи, времени, числа и

падежа. По запросу пользователя программа должна выдавать пример использования слова в предложении и возможный его перевод. Объем словаря должен составлять не менее 5 слов на каждую букву.

Задание 2.9. Каталог

Разработать программу поиска названия книги, её автора и года издания по заданному шифру (ключу). Предусмотреть возможность расширения, редактирования, сокращения библиотеки и её сортировки по названию книги, по авторам, по году издания. Программа должна сообщать о наличии книги в библиотеке, кому выдана книга, если её нет на месте, фиксировать все текущие запросы, количество выдаваемых книг в день. Объем библиотеки должен составлять не менее 50 книг.

Задание 2.10. Зарплата

Разработать программу выдачи ведомости, содержащей такие графы, как фамилия, должность, оклад, налог, отчисления на соцстрах, в пенсионный фонд, больничные листы, профсоюзные взносы, сумма к выдаче на руки. Фамилию, должность и величину оклада вводить в режиме диалога, остальные графы программа должна заполнять самостоятельно в соответствии с заданными процентными ставками. Предусмотреть возможности добавления, редактирования и удаления строк (с пересчётом автоматических полей), сортировки фамилий и по величине оклада, суммирования столбцов таблицы с денежным типом. Ведомость должна содержать не менее чем 50 записей.

Задание 2.11. Стипендия

Разработать программу выдачи стипендиального протокола группы, содержащего такие графы, как фамилия, имя и отчество студента, экзаменационные оценки по четырем предметам и сумма начисленной стипендии. Программа на основе заранее сформированного преподавателем списка группы должна обеспечивать ввод экзаменационных оценок в диалоговом режиме и расчет суммы стипендии в соответствии со следующим алгоритмом: основной размер стипендии назначается при наличии смешанных оценок; при наличии оценок только "хорошо" и "отлично" начисляется надбавка в размере 25 % от основного размера стипендии, а отлично успевающему студенту надбавка начисляется в размере 50 %, в случае получения неудовлетворительной оценки студенту стипендия не начисляется; для губернаторских стипендиатов основной размер стипендии должен быть увеличен на 30%. Предусмотреть возможность коррекции результатов экзаменационной сессии и начисленной суммы стипендии, а также редактирование, удаление и добавление списка студентов группы в режиме преподавателя. Список группы должен состоять не менее чем из 15 студентов.

Задание 2.12. Передача

Разработать программу выдачи экзаменационного листа для передачи неудовлетворительной оценки, полученной в ходе сессии. Программа на основе заранее сформированного преподавателем списка студентов с результатами экзаменационной сессии по четырем предметам должна выдавать экзаменационный лист на каждого неуспевающего студента, в котором должны быть указаны наименование предмета, фамилия экзаменатора, фамилия и номер группы студента, номер зачетной книжки и дата экзамена, а также предусмотрены графы для визы преподавателя и декана и оценки, полученной на переэкзаменовке. Предусмотреть возможность коррекции преподавателем результатов экзаменационной сессии. Список группы должен состоять не менее чем из 15 студентов.

Тема 3. Игровые программы

Разработать программы, обеспечивающие досуг человека. При проектировании игровых программ следует учесть следующие требования.

1. Каждая игровая программа должна иметь систему меню, а также графический интерфейс пользователя, выполненный с помощью графических возможностей определенного алгоритмического языка, который позволяет осуществлять удобную навигацию по системе меню с помощью клавиш управления курсором. Допускается использования манипулятора

типа «мышь».

2. Предусмотреть заставку, в которой выводится название игры и имя автора, иметь главное меню с пунктами получения справки по правилам игры и выходом из программы.

3. Программа должна всегда подсказывать пользователю на экране каждого меню, какие возможности и действия доступны в текущий момент времени.

4. Для ввода данных предусмотреть интерфейс пользователя и систему меню.

5. Если в программе есть какие-либо предустановленные данные, которые используются в программе, то их необходимо хранить в отдельном файле, и загружать во время запуска. Все изменения должны сохраняться в файле.

6. Если результатом игры является набранное количество очков или определенное число шагов, необходимое для выигрыша, то нужно организовать файл, хранящий фамилию победителей, достигших лучших результатов, а в самой игре дать возможность просмотра таблицы рекордов из главного меню.

7. Важно чтобы программа контролировала вводимые пользователем данные, и не позволяла ему вводить недопустимые значения (например, отрицательные расстояния и массы, неправильный формат дат, слишком большой возраст людей и т.д.)

Задание 3.1. "Что, где, когда?"

Разработать программу, моделирующую популярную телевизионную игру "Что, где, когда?". Программа должна на предварительном этапе выводить на экран игровое поле рулетки с указанием цифр секторов и их цветов, а также величины ставки для каждого сектора. Из банка вопросов для каждого сектора случайным образом выбирается вопрос, сложность которого пропорциональна величине ставки. В ходе игры по игровому полю запускается шарик, определяющий номер некоторого сектора, и выдается текст соответствующего вопроса. Текст вопроса должен быть сформулирован таким образом, чтобы был предусмотрен четкий однозначный ответ. В тексте вопроса можно определять и форму ответа. В процессе игры фиксировать количество верных и неверных ответов и сумму соответствующих выигрышей. Победитель, определяемый по максимальной сумме выигрыша, награждается символом мудрости – совой. Банк вопросов должен содержать не менее 50 вопросов.

Задание 3.2. Искатель кладов

На экране дисплея изображается игровое поле и табло, фиксирующее количество набранных очков. Случайным образом располагается определенное число "кладов", причем с течением времени их количество может увеличиваться. Задача играющего состоит в том, чтобы набрать наибольшее количество очков – "кладов". При этом за кладоискателем гонится разбойник, который движется быстрее кладоискателя, но выбор направления движения является случайным. Если собраны все клады, то происходит переход на более высокий уровень игры (например, увеличивается число разбойников), если же кладоискатель пойман разбойником, то игра заканчивается, фиксируется набранное количество очков и определяется место играющего в таблице рекордов. Количество уровней должно быть не менее 10.

Задание 3.3. Пасьянс

Задача играющего состоит в том, чтобы за минимальное число ходов собрать по порядку от шестерки до туза в каждом из четырех рядов карты соответствующей масти. С этой целью программа на экране дисплея располагает случайным образом игральные карты в четыре ряда рубашками вниз. Играющий указывает одну из карт, и программа открывает её. Выбранная карта автоматически перемещается на её соответствующее масти и номиналу место, та карта, на место которой положена данная, открывается и перемещается в соответствующий ряд. Если открытая карта перемещается на свободное место или остается на месте, то играющий делает очередной ход, открывая новую карту. Игра заканчивается, когда все карты окажутся открытыми. Программа фиксирует количество сделанных играющим ходов и заносит его фамилию в список победителей, если это число минимально.

Задание 3.4. Морской бой

На экране дисплея изображается два игровых поля. На одном из них играющий располагает произвольным образом определенное число кораблей различных классов. На другом поле свои корабли располагает компьютер, не отображая их на экране. В процессе игры противники (играющий и компьютер) поочередно производят выстрелы по полю соперника. Противник фиксирует результативность выстрела и выдает соответствующую реплику: "Мимо", "Попал", "Убил", клеточка, по которой произведен выстрел, закрашивается определенным образом. Если произведенный выстрел был удачным, то предоставляется право на внеочередной поощрительный выстрел. Игра считается законченной, когда потоплены все корабли одного из противников. Если победил игрок, а не компьютер, то его результат (количество выстрелов) заносится в таблицу рекордов. Лучшим считается адмирал, добившийся победы за наименьшее количество ходов.

Задание 3.5. Игра "15"

Программа случайным образом располагает на игровом поле фишки с пятнадцатью цифрами и одну пустую клеточку (часто в данной игре при случайной генерации поля получается не собираемая комбинация, поэтому лучшим образом будет, сначала расположить все цифры по порядку, а затем программным способом перемешать их, совершая большое количество случайных перестановок, как делает это сам игрок). Играющий, используя резервную клеточку, с помощью клавиш перемещает фишки с цифрами таким образом, чтобы расположить их по порядку. Программа фиксирует количество перемещений. Лучшим считается тот игрок, который упорядочит все фишки за наименьшее число ходов.

Задание 3.6. Лабиринт

На экране формируется лабиринт по определенному закону. Играющий за минимальное число ходов, перемещая курсор с помощью клавиш, должен найти путь от входа к выходу. При этом прокладываемый путь отображается на экране. Предусмотреть возможность ввода лабиринта играющим, сохранение и загрузка созданного лабиринта в файл, и поиск оптимального пути в лабиринте с помощью компьютера.

Задание 3.7. Тир

На игровом поле случайным образом располагается мишень произвольного размера на короткий отрезок времени. Играющий, управляя клавишами прицеливания или мышью, должен за отведенное время поразить мишень, которая двигается по экрану. За каждую сбитую мишень начисляются премиальные очки пропорционально, например, размеру и расстоянию до мишени. За каждой промах назначаются штрафные очки. Игра заканчивается по истечении определенного времени или после поражения заданного количества мишеней. Программа фиксирует список победителей и вносит их в таблицу рекордов.

Задание 3.8. XONIX

По всей площади экрана изображается игровое поле, по которому движется шарик, подобно теннисному мячу отражаясь от стенок поля. Играющий, управляя движением курсора, должен отобразить у шарика не менее 70% площади игрового поля. Если траектория курсора представляет собой замкнутый контур, то площадь, ограниченная им, считается отобранной у шарика. Когда отсекается 70% площади поля, то осуществляется переход к более сложному уровню игры, при котором увеличивается количество шариков или их скорость. На всю игру играющему отводится ограниченное число попыток, например, пять. Попытка считается использованной, если во время движения курсора его траектория пересекается одним из шариков. После использования всех попыток фиксируется количество площади, отобранной игроком у шариков, и результат заносится в таблицу, если он достоин этого. Количество уровней должно быть не менее 10.

Задание 3.9. Карточная игра «Дурак»

Разработать программу, моделирующую карточную игру "Дурак" на двух игроков, один из которых управляется компьютером. Каждому раздаётся по 6 карт, следующая карта

открывается и её масть устанавливает козырь для данной игры, и остальная колода кладётся сверху так, чтобы козырная карта была всем видна. Туз не может быть козырем. Козырный туз не отбивается в игре. Цель игры – избавиться от всех карт. Последний игрок, не избавившийся от карт, остаётся «в дураках». Запрещается забирать карты, которые отбили. Отбитые карты идут в отбой.

Задание 3.10. Домино

Смоделировать игру, в "козла" между человеком и компьютером. Программа раздает случайным образом по 7 фишек игроющему и компьютеру. Фишки игрока располагаются на экране в открытом виде, а фишки компьютера – в закрытом. Игру начинает тот, у кого находится меньший дупль. Ход игры должен отображаться на охроне. Если у одной из сторон нет необходимой фишки для очередного хода, то делается заем из "базара" (фишки, оставшиеся не розданными) до появления нужной фишки, в противном же случае ход пропускается. Когда у одной из сторон не остается фишек, фиксируется количество оставшихся очков у противоположной стороны и вновь раздаются фишки. Проигрывает тот, кто раньше наберет в сумме не менее 101 очка.

Тема 4. Графические программы

В данном разделе должны разрабатываться программы, использующие графические возможности алгоритмического языка. При проектировании графических программ следует учесть следующие требования.

1. Каждая игровая программа должна иметь систему меню, а также графический интерфейс пользователя выполненный с помощью графических возможностей языка, который позволяет осуществлять удобную навигацию по системе меню с помощью клавиш управления курсором. Допускается использования манипулятора типа «мышь».

2. Программа должна иметь заставку, в которой выводится название программы и имя автора, иметь главное меню с пунктами получения справки и выходом из программы.

3. Программа должна всегда подсказывать пользователю на экране каждого меню, какие возможности и действия доступны в текущий момент времени.

4. Для ввода данных предусмотреть интерфейс пользователя и систему меню.

5. Если в программе есть какие-либо предустановленные данные, которые использует программа, то их необходимо хранить в отдельном файле, и загружать во время запуска. Все их изменения также должны сохраняться в файле.

6. Важно чтобы программа контролировала вводимые пользователем данные, и не позволяла ему вводить недопустимые значения (например, отрицательные расстояния и массы, неправильный формат дат, слишком большой возраст людей и т.д.)

Задание 4.1. Построение графиков аналитических функций

Разработать программу графического представления семейства произвольных функций в количестве не менее 15, заданных математическими формулами. При этом конкретный вид формулы определяется пользователем в диалоговом режиме. В программе должно быть предусмотрено масштабирование по осям X и Y , а также расчёт значений функций в заданном диапазоне с заданным шагом и вывод их на экран в виде таблицы, сохранение результатов табулирования в файле.

Задание 4.2. Построение графиков табличных функций

Разработать программу графического представления семейства произвольных функций в количестве не менее 15, заданных табличным способом. При этом таблицы, задающие функции, должны храниться в файле. Предусмотреть масштабирование по осям X и Y и интерполяцию между соседними значениями. Также должна быть предусмотрена возможность создания своих таблиц функций.

Задание 4.3. Табло для голосования

Разработать программу, выдающую на экран дисплея табло для голосования. На табло должны на полный экран индцироваться текущее время, отведенное для голосования, по

истечении которого в строках "за", "против", "воздержалось", "итога" должны появляться цифры, соответствующие результату голосования, и сообщение "решение принято" или "решение не принято". Предусмотреть два режима ввода данных о голосовании: вручную с помощью функциональных клавиш, автоматически с помощью датчика случайных чисел, из файла с заранее сгенерированными данными.

Задание 4.4. Графический редактор

Разработать программу, обеспечивающую реализацию следующих функций.

1. Изменение видеохарактеристик экрана.
2. Построение вертикальных и горизонтальных линий одинарной и двойной толщины.
3. Построение прямоугольника одинарной и двойной толщины.
4. Построение круга, эллипса.
5. Построение кривых линий, рисуемых произвольным образом.
6. Установка текущего цвета рисования.
7. Позиционирование курсора в любое место экрана.
8. Сохранение результатов рисования в файл, как последовательность действий по рисованию линий, прямоугольников, овалов и т.д.

9. Загрузка ранее нарисованных изображений из файла.

Должно быть сохранено не менее 5 файлов с рисунками.

Задание 4.5. Изометрия

Разработать программу, обеспечивающую изображение на экране дисплея определенного набора пространственных геометрических фигур: параллелепипед, сфера, прямоугольная пирамида, конус, усеченный конус. Предусмотреть возможности масштабирования фигур по заданным размерам и построения трех проекций для каждой фигуры. Все предоставляемые услуги оформить в виде сервисного меню с библиотекой геометрических фигур.

Задание 4.6. Графическая интерпретация математических методов

Разработать программу, иллюстрирующую графически различные математические методы, такие как интегрирование по методу прямоугольников (левых, правых, средних), трапеций и методом Симпсона, решение уравнений методом половинного деления, методом касательных, методом хорд, методом итераций. Графическое построение по каждому методу должно вестись по шагам и сопровождаться пояснениями в словесной форме. Создать библиотеку математических методов и меню для выбора вида работ по каждому методу.

Задание 4.7. Сортировка

Разработать программу, реализующую основные методы сортировки: метод "пузырька", метод линейного выбора и т.д. (не менее 6 алгоритмов) и иллюстрирующую каждый из них графически. Сравнить эффективность их работы, дать оценку каждому из них, сделать вывод об их применимости в разных случаях, об их сильных и слабых сторонах, выразить скорость сортировки графически с применением элементов мультипликации. Оформить библиотеку алгоритмов в виде сервисного меню. Предусмотреть загрузку данных из файла.

Задание 4.8. Клавиатура

Разработать программу для изучения клавиатуры. Для этой цели необходимо изобразить общий вид клавиатуры на дисплее, перемещая курсор, выбрать необходимую клавишу и по введенному запросу изобразить на экране рисунок выбранной клавиши и выдать текст, поясняющий её назначение. Предусмотреть режим тестирования пользователя: на экране отображается клавиатура и на ней случайным образом на короткое время подсвечиваются клавиши. Пользователь должен как можно быстрее нажать их на самой клавиатуре. Тестирование происходит заданное время, в течение которого программа фиксирует скорость нажатия, а также процент ошибок и, в конце, выводит данную информацию на экран. Лучшие результаты заносятся в таблицу рекордов.

Задание 4.9. Калейдоскоп

Создать программу, моделирующую детскую игрушку "Калейдоскоп". Программа в центре экрана должна крупно изобразить контур одной из геометрических фигур: круга, эллипса, квадрата, ромба, треугольника и продублировать внутри неё несколько изображений этой фигуры с различной толщиной линий и миганием изображения. С течением времени случайным образом должны меняться типы фигур, их величина, цвет и количество.

Тема 5. Работа с матрицами.

Задания этой темы ориентированы, прежде всего, на студентов заочной формы обучения. При проектировании программ следует учесть следующие требования.

1. Программа должна быть выполнена на определенном алгоритмическом языке и может иметь консольный интерфейс взаимодействия с пользователем.

2. Все неизвестные данные должны вводиться пользователем с клавиатуры.

3. Важно чтобы программа контролировала вводимые пользователем данные и не позволяла ему вводить недопустимые значения (например, отрицательные размеры матриц, индексы строк/столбцов, выходящие за границы массива).

4. При вводе пользователем недопустимого значения программа должна сообщить, какие сообщения следует вводить и предоставить возможность повторного ввода.

5. При заполнении элементов матриц случайным образом следует генерировать весь допустимый диапазон значений (вещественные, отрицательные, нулевые, положительные), если иное не оговорено в задании.

6. Для промежуточных вычислений для контроля выполнения программы необходимо предусмотреть эхо-печать данных, также выводить на экран по шагам все изменения в матрицах и все промежуточные результаты вычислений.

Задание 5.1

Составить функцию нахождения максимального значения элемента в одномерном массиве. Используя эту функцию, найти сумму максимальных элементов в первой строке матрицы B и во втором столбце матрицы $A = B^2 + C^T B$, где A, B, C – квадратные матрицы размером $n \times n$. Элементы матрицы C задать случайным образом. Элементы матрицы B определить по заданной формуле.

Задание 5.2

Составить функцию нахождения минимального значения элемента в одномерном массиве. Используя функцию, найти сумму минимальных элементов в k -ой строке матрицы B и в l -ом столбце матрицы $A = B + B^T C B$, где A, B, C – квадратные матрицы размером $n \times n$. Элементы матрицы C задать случайным образом. Элементы матрицы B определить по заданной формуле.

Задание 5.3

Составить процедуру вычисления обратной матрицы методом Жордана и с ее помощью решить заданные системы линейных уравнений. Предусмотреть возможность ввода коэффициентов и свободных членов уравнений.

Задание 5.4

Составить процедуру вычисления обратной матрицы методом клеточных матриц и с ее помощью решить заданные системы линейных уравнений. Предусмотреть возможность ввода коэффициентов и свободных членов уравнений.

Задание 5.5

Составить процедуру вычисления обратной матрицы заданным методом и с ее помощью найти матрицы $A = X + X^{-1}$; $B = H + H^{-1}$. Определить сумму их элементов, расположенных в нечетных строках, где X, H – заданные матрицы размером $n \times n$.

Задание 5.6

Составить функцию для нахождения суммы элементов квадратной матрицы, расположенных выше главной диагонали. Определить максимальную из указанных сумм для матриц $X = A^2 + B^3$; $Y = AA^T + BB^T$, где A, B – исходные заданные матрицы размером $n \times n$.

Задание 5.7

Составить функцию для нахождения суммы элементов квадратной матрицы, расположенных ниже главной диагонали. Определить минимальную из указанных сумм для матриц $X = A^2 + B^3$; $Y = AA^T + BB^T$, где A, B – исходные заданные матрицы размером $n \times n$.

Задание 5.8

Составить процедуру определения числа положительных, нулевых и отрицательных элементов произвольной квадратной матрицы. Найти суммарное количество соответствующих элементов в матрицах: $A = X^T X + X^2$ и $B = E^T E + E^2$, где X, E – квадратные матрицы размером $n \times n$, элементы которых вычисляются по заданным формулам.

Задание 5.9

Составить функцию для нахождения суммы элементов квадратной матрицы, у которых сумма индексов является нечетным числом. Используя эту функцию, определить указанную сумму для матриц: $A, B, E = A^2 B + A^T B^T$. Элементы матриц A и B задать случайным образом.

Задание 5.10

Составить функцию для нахождения произведения элементов некоторой квадратной матрицы, у которых сумма индексов является четным числом. Используя эту функцию, определить указанное произведение для матриц: $A, B, E = A^2 B + A^T B^T$. Элементы матриц A и B задать случайным образом.

Задание 5.11.

Составить функцию нахождения минимального элемента главной диагонали произвольной квадратной матрицы. Используя указанную функцию, определить сумму минимальных элементов главных диагоналей матриц $D = P + K$ и $L = P^T - KP$, где P, K – исходные матрицы размером $n \times n$. Элементы матрицы K задаются случайным образом, а элементы матрицы P вычисляются по заданной формуле.

Задание 5.12

Составить функцию нахождения максимального элемента побочной диагонали произвольной квадратной матрицы. Используя указанную функцию, определить сумму максимальных элементов побочных диагоналей матриц $D = P + K$ и $L = P^T - KP$, где P, K – исходные матрицы размером $n \times n$. Элементы матрицы K задаются случайным образом, а элементы матрицы P вычисляются по заданной формуле.

Задание 5.13

Составить процедуру упорядочения по убыванию элементов одномерного массива. Используя процедуру, упорядочить элементы k -ой строки матрицы $A = XX^T + X^3 + X^2 Y$, где A, X, Y – квадратные матрицы размером $n \times n$. Элементы матриц X и Y задать случайным образом.

Задание 5.14

Составить процедуру упорядочения по возрастанию элементов одномерного массива. Используя процедуру, упорядочить элементы k -ой строки матрицы $A = X^4 + Y^2 X$, где A, X, Y – квадратные матрицы размером $n \times n$. Элементы матриц X и Y задать случайным образом.

Задание 5.15

Составить подпрограмму, позволяющую поменять местами k -ые строки в матрицах A и B , а также l -ые строки в матрицах A и C . После замены строк определить минимальный элемент в матрице $D = AB + C^T B^T$, где A, B, C – заданные матрицы.

Задание 5.16

Составить процедуру перестановки соответствующих элементов заданных столбцов квадратной матрицы. Используя эту процедуру, поменять местами столбцы k и l матрицы A размером $n \times n$, вычислить матрицу $D = A^3 + BA^T$ и поменять в ней местами столбцы q и r , B – квадратная матрица размером $m \times m$. Элементы матрицы A задать случайным образом, а элементы матрицы B вычислить по формуле $b_{ij} = i/j$.

Задание 5.17

Составить процедуру перестановки соответствующих элементов строк с номерами k и l в квадратной матрице размером $n \times n$. Используя эту процедуру, поменять местами строки k и l матрицы A размером $n \times n$, вычислить матрицу $D = B^2 B^T + A^T$ и поменять в ней местами строки q и r , B – квадратная матрица размером $m \times m$. Элементы матрицы A задать произвольно, а элементы матрицы B вычислить по формуле $b_{ij} = i/j$.

Задание 5.18.

Составить процедуру упорядочения по убыванию элементов квадратной матрицы, расположенных по строкам. Используя эту подпрограмму, найти минимальный из элементов матриц $A, B, C = A^T + A^2 + A^3$. Элементы матрицы A задать случайным образом, а элементы матрицы B вычислить по заданной формуле.

Задание 5.19

Составить процедуру упорядочения по возрастанию элементов квадратной матрицы, расположенных по столбцам. Используя эту подпрограмму, найти максимальный из элементов матриц $A, B, C = A^T + B^T + A^2 + B^2$. Элементы матрицы A задать случайным образом, а элементы матрицы B вычислить по заданной формуле.

Задание 5.20

Составить процедуру преобразования прямоугольной матрицы размером $m \times n$ в одномерный массив в порядке следования строк. Используя эту процедуру, преобразовать исходные матрицы A и C . Найти сумму минимальных элементов этих массивов. Элементы матрицы A задать случайным образом, а элементы матрицы C вычислить по заданной формуле.

4.8. Темы рефератов

Реферативная форма обучения и контроля по дисциплине «Алгоритмические языки и программирование» имеет своей целью текущий контроль самостоятельного изучения отдельных тем учебной дисциплины, подготовки выступлений и коллективной дискуссии. Предлагаются следующие темы рефератов.

Тема 1. Создание алгоритмов и программ. Теория алгоритмизации. Решение предметной задачи на ЭВМ. Этапы решения задачи на ЭВМ. Постановка задачи. Построение математической модели. Выбор метода решения. Алгоритмизация. Программирование. Отладка и тестирование программ. Документирование программ.

Тема 2. Создание алгоритмов и программ. Теория алгоритмизации. Алгоритм и его свойства. Алгоритмическая система. Алгоритмизация – процесс разработки и описания алгоритма. Средства записи алгоритмов. Словесная запись алгоритма. Схемы алгоритмов. Правила построения схем алгоритмов. Структурограммы. Правила построения структурограмм.

Тема 3. Создание алгоритмов и программ. Теория алгоритмизации. Технология разработки алгоритмов. Структурный подход. Основные принципы. Теорема о структуриро-

вании. Базовые управляющие структуры. Проектирование алгоритмов. Нисходящее и восходящее проектирование. Оформление алгоритмов.

Тема 4. Создание алгоритмов и программ. Теория алгоритмизации. Структуры алгоритмов. Алгоритмы линейной структуры. Ветвления. Циклы. Циклы с пред- и постусловием. Циклы с параметром. Итерационные циклы. Вложенные циклы. Вспомогательные/подчиненные алгоритмы. Данные. Простые данные. Константы и переменные. Структуры данных.

Тема 5. Парадигмы и языки программирования. Эволюция языков программирования. Методы и стили программирования. Императивное и декларативное программирование. Императивное программирование и алгоритмические языки. Структура языка императивного стиля программирования. Концепция действий и концепция данных. История развития алгоритмических языков.

Тема 6. Структурное программирование. Основные принципы. Программирование алгоритмов различной структуры. Реализация базовых управляющих структур в алгоритмическом языке программирования. Основные понятия языка.

Тема 7. Программирование алгоритмов линейной структуры. Естественный ход выполнения программы. Структура программы. Описание данных. Простые данные. Константы. Переменные. Описание переменных. Понятие идентификатора. Описание действий. Понятие операции, функции, оператора. Пример простейшей программы. Ввод данных: процедуры ввода. Вычисления: оператор присваивания, выражения, стандартные операции и функции. Вывод данных: процедуры вывода. Стил записи и оформление программ.

Тема 8. Программирование алгоритмов с ветвлениями. Реализация в алгоритмах и программах фактора принятия решения. Оператор условного перехода. Понятие логического выражения как реализация критерия принятия решения в алгоритмах и программах. Оператор варианта – реализация в алгоритмах и программах многоальтернативного выбора. Оператор безусловного перехода – неструктурная передача управления. Приемы программирования алгоритмов с ветвлениями.

Тема 9. Программирование циклических алгоритмов. Реализация в алгоритмах и программах фактора повторяемости вычислений. Операторы цикла. Программирование циклических алгоритмов. Циклы с пред- и пост-условием, циклы с параметром. Правила организации циклов. Итерационные циклы – вычисления с заданной точностью. Вложенные циклы. Правила организации вложенных циклов. Проектирование программ сложной структуры. Примеры циклических алгоритмов и программ сложной структуры.

Тема 10. Организация ввода-вывода данных. Организация ввода-вывода данных – работа со стандартными файлами ввода/вывода. Процедуры ввода. Ввод числовых и символьных данных. Процедуры вывода. Стандартная форма вывода данных. Управление формой вывода. Таблицы, графики, рисунки.

Тема 11. Отладка программ. Причины и типы ошибок, возникающих при проектировании и разработке программ, способы и средства их отладки в интегрированной среде.

Тема 12. Концепция данных в программировании. Концепция типа. Понятие типа данных. Понятие типизированного языка программирования. Иерархия типов. Скалярные типы. Имя и значение. Вещественные типы. Порядковые типы. Целые типы. Логический тип. Символьный тип. Структурированные типы. Статические и динамические типы данных.

Тема 13. Регулярный тип. Массивы данных. Описание и использование массивов в программах. Одномерные массивы, представление векторов. Операции над векторами. Двумерные массивы, представление матриц. Операции над матрицами.

Тема 14. Комбинированный тип. Записи. Представление и обработка табличных документов.

Тема 15. Множественный тип. Множества. Представление и использование множеств в алгоритмических языках программирования

Тема 16. Символьный и строковый типы. Строки. Обработка текстов.

Тема 17. Файловый тип. Файлы. Хранение и поиск данных на внешних носителях. Файловая организация данных. Операции над файлами. Доступ к компонентам файла. Понятие файла, файловой переменной и файлового типа. Процедуры и функции для работы с файлами. Типизированные файлы. Текстовые файлы. Нетипизированные файлы.

Тема 18. Ссылочный тип. Понятие статических и динамических данных. Понятие статической и динамической организации памяти. Статические и динамические переменные. Ссылки или указатели. Динамические структуры данных: списки, стеки, очереди, деревья.

Тема 19. Блочный принцип организации программ. Подпрограммы. Процедуры и функции. Понятие блока. Блочный принцип организации программ. Подпрограммы. Процедуры и функции. Описание, использование, обращение. Параметры процедур и функций. Формальные и фактические параметры. Параметры-значения и параметры-переменные, параметры константы. Процедурный тип.

Тема 20. Понятие рекурсии. Семантика рекурсивного вызова. Типы рекурсий. Программирование рекурсивных функций.

Тема 21. Понятие модуля. Принципы модульного программирования. Понятие модуля. Создание и использование модулей. Общая структура модуля. Компиляция и использование модулей. Стандартные модули. Управление экраном в текстовом и графическом режимах. Графика. Управление клавиатурой и звуком.

Тема 22. Методология проектирования программ. Современные методы и технологии проектирования. Общие вопросы по методологии проектирования программ. Понятие жизненного цикла программы. Модели жизненного цикла программы. Фазы разработки программной системы. Средства поддержки жизненного цикла. Краткий обзор современных методов проектирования. Документирование программ. Стандарты в области производства и документирования программ.

5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ 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 «Алгоритмические языки и программирование»).

6. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Основная литература

1. Москвитина О.А., Пылькин А.Н. Примеры и задачи по программированию на Паскале и Питоне. Фонд оценочных средств для промежуточной аттестации: учеб. пособие Ч.1 / Москвитина О.А. – М.: КУРС, 2018. – 191 с.

2. Парфилова Н.И., Пылькин А.Н., Трусов Б.Г. Программирование. Основы алгоритмизации и программирования: учебник для вузов / Трусов Б.Г. – М.: Академия, 2012. – 240 с.

3. Парфилова Н.И., Пылькин А.Н., Трусов Б.Г. Программирование. Структурирование программ и данных: учебник для вузов / Трусов Б.Г. – М.: Академия, 2012, 2016 2-е изд., стер. – 240 с.

4. Парфилова Н.И., Пылькин А.Н., Трусов Б.Г. Информатика и программирование. Основы информатики: учебник для вузов / Трусов Б.Г. – М.: Академия, 2012, 2016 2-е изд., стер. – 256 с.

5. Парфилова Н.И., Пылькин А.Н., Трусов Б.Г. Информатика и программирование. Алгоритмизация и программирование: учебник для вузов / Трусов Б.Г. – М.: Академия, 2012. – 336 с.

6. Антипов О.В., Москвитина О.А. Программирование на Паскале: метод. указ. к курс. работе / Антипов О.В. – Рязань: РГРТУ, 2019. – 48 с.

7. Москвитина О.А., Новичков В.С., Пылькин А.Н. Сборник примеров и задач по программированию: учеб. пособие / Москвитина О.А. – М.: Горячая линия-Телеком, 2007. – 244 с.

8. Новичков В.С., Парфилова Н.И., Пылькин А.Н. Основы информатики: учеб. пособие / Новичков В.С. – Рязань: РГРТА, 2006. – 311 с.

9. Новичков В.С., Парфилова Н.И., Пылькин А.Н. Алгоритмизация и программирование на Турбо Паскале: учеб. пособие / Новичков В.С. – М.: Горячая линия-Телеком, 2005. – 438 с.

6.2. Дополнительная литература

10. Москвитина О.А. Новичков В.С., Пылькин А.Н. Программирование на языке высокого уровня: метод. указ. к курс. работе / Москвитина О.А. – Рязань: РГРТУ, 2009. – 64 с.

11. Москвитина О.А., Новичков В.С., Пылькин А.Н. Алгоритмические языки и программирование: сб. примеров и заданий к практ. и лаб. работам / Москвитина О.А. – Рязань: РГРТУ, 2007. – 40 с.

12. Москвитина О.А., Новичков В.С., Пылькин А.Н. Алгоритмические языки и программирование: сб. примеров и заданий к практ. и лаб. работам / Москвитина О.А. – Рязань: РГРТУ, 2007. – 48 с.

13. Москвитина О.А., Новичков В.С., Пылькин А.Н. Алгоритмические языки и программирование: сб. примеров и заданий к практ. и лаб. работам / Москвитина О.А. – Рязань: РГРТУ, 2006. – 56 с.

14. Симонович С.В. Информатика. Базовый курс: учеб. пособие для вузов / Симонович С.В. – СПб.: Питер, 2009. – 639 с.

15. Назаренко П.А. Алгоритмы и структуры данных [Электронный ресурс]: учебное пособие / П.А. Назаренко. – Электрон. текстовые данные. – Самара: Поволжский государственный университет телекоммуникаций и информатики, 2015. – 130 с. – 2227-8397. – Режим доступа: <http://www.iprbookshop.ru/71819.html>.

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 показались 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.

4. Рекомендуется по каждой теме выразить свое мнение, комментарий, вывод.

6.3.2. Методические указания к практическим занятиям/лабораторным занятиям

Подготовка к лабораторным работам. Перед началом лабораторной работы необходимо ознакомиться с методическими указаниями к данной лабораторной работе, внимательно ознакомиться с заданием и желательнее 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 курсовому проектированию (курсовой работе)

Выполнение курсового проектирования. Выполнение курсовой работы начинается со знакомства с литературой по предложенной тематике, с дополнительной методической литературой для обеспечения вспомогательных сервисных возможностей. При выполнении курсовой работы необходимо использовать материал лекций и рекомендуемую учебно-методическую литературу. После знакомства с литературой анализируется задание, по словесному описанию задачи разрабатывается общий план решения и обобщенные, укрупненные схемы алгоритмов. В соответствии с принципами структурного программирования схемы алгоритмов проектируются из отдельных базовых структур по нисходящей методике, от более крупных блоков – к мелким, к деталям.

В схемах определяются участки, логически законченные и неоднократно повторяющиеся, которые целесообразно оформлять в виде отдельных процедур и функций. Предполагаемые наборы данных оформляются отдельно и хранятся в соответствующих файлах. Определяются имена основных структур программы, приводятся в соответствие формальные и фактические параметры этих структур. Намечаются места вводных и пояснительных комментариев, выбирается способ внешнего оформления разрабатываемой программы для наибольшей наглядности и повышения удобства использования. Далее разрабатывается соответствующий программный продукт и производится его отладка и тестирование. Отлаженная и протестированная программа и полученные результаты анализируются. Если полученные результаты соответствуют ожидаемому, целесообразно перейти к оформлению пояснительной записки по выполненной работе.

На заключительном этапе работы студент должен предоставить *пояснительную записку* и полную инструкцию по использованию разработанной программы, рассчитанную на неподготовленного к работе пользователя, выводы и особенности, отмеченные при выполне-

нии данного конкретного задания, а также демонстрационный пример, где были бы отражены возможности, достоинства, перспективы произведенной разработки.

Также необходимо вести учет трудоемкости каждого этапа для сравнения теоретических затрат и реально затрачиваемого времени. В выводах, касающихся особенностей конструирования задач данного типа, необходимо провести анализ причин, вызвавших либо резкое увеличение, либо резкое уменьшение трудозатрат на определенный этап разработки.

Студент обязан периодически по установленному графику предъявлять материалы по выполненным этапам работы закрепленному преподавателю в часы его консультаций.

Для контроля проводимых работ необходимо вести календарь выполнения работ:

№ n/n	Этапы решения задачи	Ориентировочная трудоемкость, (ч)	Реально затрачи- ваемое время, (ч)
1.	Анализ предметной области и формализация поставленной задачи. Создание плана проекта. Формирование списка задач проекта.	2 недели	
2.	Планирование ресурсов. Анализ и выравнивание загрузки ресурсов. Разработка структуры организации данных.	2 недели	
3.	Определение длительности задач и взаимосвязей между ними. Составление алгоритмов.	2 недели	
4.	Написание программного продукта.	2 недели	
5.	Устранение синтаксических ошибок.	2 недели	
6.	Подготовка тестов. Тестирование и отладка.	2 недели	
7.	Подготовка документации и отчета по проекту	2 недели	

Если имели место дополнительные этапы, необходимо внести их в календарный план. Календарь заполняется в течение всего времени работы над заданием, окончание каждого из этапов фиксируется преподавателем.

Оформление курсовой работы. Документами курсовой работы являются текст программы и пояснительная записка. Пояснительная записка имеет объем *не менее 25 страниц текста* (без листинга самой программы) на бумаге форматом 297x210 мм. Распечатки (листинги) программного кода оформляются в виде приложений.

Содержательная сторона пояснительной записки должна включать в себя описание всех этапов проделанной работы. Материал пояснительной записки рекомендуется располагать в следующем порядке:

- титульный лист;
- оглавление;
- задание на курсовую работу;
- введение;
- анализ и постановка задачи;
- разработка схем алгоритмов и их описание;
- разработка программного продукта и его описание;
- инструкция по использованию разработанного программного продукта;
- результаты отладки (в этом разделе приводятся все полученные в процессе отладки листинги программы с расшифровкой сообщений об ошибках);
- решение контрольных примеров и проверка правильности функционирования программы (верификация программы) по результатам контрольных просчетов;
- краткая аннотация программы с подробным описанием возможностей, достоинств, особенностей разработки;
- заключение;
- библиографический список.
- приложения: текст разработанного программного продукта.

Инструкция по использованию разработанной программы должна включать сведения об особенностях функционирования программы, используемых подпрограммах, форме и порядке подготовки исходных данных для решения, о виде и смысле получаемых результатов.

Порядок сдачи курсовой работы. Законченная пояснительная записка к курсовой работе и диск с исходным кодом программы, скомпилированным *EXE* файлом, и всеми необходимыми файлами данных в запланированный срок представляется преподавателю для проверки, обычно *за 2 недели до начала экзаменационной сессии*.

После сдачи на проверку студенту назначается время защиты работы. Сдача курсовой работы (защита) производится в форме демонстрации разработанной программы и собеседования, направленных на выявление степени самостоятельности в выполнении курсовой работы, уровня знаний студента и понимания по вопросам выполненной работы. При определении оценки учитываются показатели работы студента в течение семестра над курсовой работой, материалы пояснительной записки и ответы студента в ходе защиты.

Рекомендуется публичная защита курсовой работы с участием студентов группы.

Если курсовая работа сдаётся на проверку не в срок, то *оценка снижается на 1 балл*.

6.3.4. Методические указания к подготовке и сдаче экзамена и теоретического зачета

Экзамен и теоретический зачет – форма промежуточной проверки знаний, умений, навыков, степени освоения дисциплины. Главная задача экзамена и 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 экзамену и зачету, студент приводит в систему знания, полученные на лекциях, консультациях 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.

Экзамены и теоретические 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 экзаменов и 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 экзамену и 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 экзаменом назначается *консультация*, 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 экзаменах ответы были неудовлетворительными, а также фиксирует внимание на наиболее трудных разделах или темах курса.

На непосредственную подготовку к экзамену обычно дается 3 – 5 дней. Этого времени достаточно для углубления, расширения и систематизации знаний, полученных в ходе обучения, на устранение пробелов в знании отдельных вопросов, для определения объема ответов на каждый из вопросов рабочей программы дисциплины.

Планируя подготовку, обучаемый должен учитывать сразу несколько факторов: сложность учебного материала и степень его проработки в ходе обучения, свои индивидуальные способности. Рекомендуется делать перерывы в занятиях через каждые 50-60 минут на 10 минут. После 3-4 часов занятий следует сделать часовой перерыв. Целесообразно разделять весь рабочий день на три рабочих периода – с утра до обеда, с обеда до ужина и с ужина до сна. Каждый рабочий период дня должен заканчиваться отдыхом не менее 1 часа.

Работая в сессионном режиме, студент имеет возможность увеличить время занятий с 8 (как требовалось в семестре) до 10 часов в сутки. Подготовка к экзаменам следует начинать с общего планирования деятельности и определения 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.

6.3.5. Рекомендации по работе с литературой

Теоретический материал курса становится более понятным, когда дополнительно к прослушиванию лекции и изучению конспекта изучаются и учебная литература 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 учебных 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 (консультации), тогда занятия будут гораздо понятнее. В течение недели рекомендуется выбрать время (1 час) для работы с литературой.

6.4. 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 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 на первом году обучения 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 система 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, 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 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>.
5. Библиотека и форум по программированию. – URL: <http://www.cyberforum.ru>
6. Национальный открытый университет ИНТУИТ. – URL: <http://www.intuit.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. PascalABC.NET – свободно распространяемое ПО.

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для освоения дисциплины необходимы следующие материально-технические ресурсы:

- 1) аудитория для проведения лекционных занятий, групповых и индивидуальных консультаций, текущего контроля, промежуточной аттестации, оборудованная проектором и маркерной (меловой) доской;
- 2) классы, оснащенные персональными компьютерами для проведения лабораторных занятий и курсового проектирования;
- 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 ОЗУ: 1 Гб	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>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 работы и 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 работы и 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).</p>

		<p>5. Microsoft SQL Server (Microsoft Imagine: Номер подписки 700102019, бессрочно).</p> <p>6. Microsoft Project (Microsoft Imagine: Номер подписки 700102019, бессрочно).</p> <p>7. Свободно распространяемое программное обеспечение под лицензиями GNU, Apache, Oracle, Mozilla, CeCILL.</p>
--	--	---