


МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ С.А. ЕСЕНИНА»

Утверждаю:  
Декан  
физико-математического  
факультета  
 Н.Б. Федорова  
«31» августа 2020 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**  
**ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ**

Уровень основной профессиональной образовательной программы:  
**бакалавриат**

Направление подготовки: **02.03.03 Математическое обеспечение и администрирование информационных систем**

Направленность (профиль) подготовки: **Администрирование информационных систем**

Форма обучения: очная

Срок освоения ОПОП: нормативный **срок освоения 4 года**

Факультет: **физико-математический**

Кафедра: **Информатики, вычислительной техники и методики преподавания информатики**

Рязань, 2020

## **Вводная часть**

### **1. Цели освоения дисциплины**

Целью освоения дисциплины «Информатика и программирование» является формирование у обучающихся общекультурных и общепрофессиональных компетенций в процессе изучения информатики и программирования для последующего применения в учебной и практической деятельности.

Задачи дисциплины:

- систематизация, формализация и расширение знаний по основам информатики, приобретенные в школе;
- привитие навыков алгоритмического мышления, культуры алгоритмизации и нисходящего структурного программирования;
- формирование теоретической базы и практических умений и навыков для решения задач на компьютере в императивных системах программирования,
- формирование основ современной культуры программирования.

### **2. Место дисциплины в структуре ОПОП ВУЗА**

**2.1.** Дисциплина Б1.Б.9.«Информатика и программирование» относится к базовой части Блока 1.

**2.2.** Для изучения дисциплины «Информатика и программирование» необходимы дисциплины предшествующего уровня образования:

- «Информатика и информационные технологии».

**2.3.** Перечень последующих дисциплин, для которых необходимы знания, умения, навыки, формируемые данной дисциплиной:

- «Объектно-ориентированные языки и системы»;
- «Архитектура вычислительных систем и компьютерных сетей»;
- «Проектирование реляционных БД»;
- «Параллельное программирование»;
- «Кроссплатформенное программирование»;
- «Функциональное программирование»;
- «Компьютерная графика».

**2.4. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения основной профессиональной образовательной программы**

Изучение данной дисциплины направлено на формирование у обучающихся следующих общепрофессиональных (ОПК) и профессиональных (ПК) компетенций:

№ п/п	Номер/индекс компетенции	Содержание компетенции (или ее части)	Перечень планируемых результатов обучения по дисциплине В результате изучения дисциплины обучающиеся должны:		
			Знать:	Уметь:	Владеть (навыками):
1	2	3	4	5	6
1	ОПК-3	готовность анализировать проблему и направления развития технологий программирования	<ul style="list-style-type: none"> <li>• предмет и задачи исследования информатики как науки;</li> <li>• сущность и значение информации в развитии современного информационного общества;</li> <li>• этапы решения задачи на компьютере на основе парадигмы императивного программирования;</li> <li>• понятие алгоритма и его основные свойства;</li> <li>• основные понятия алгоритмической системы: представление информации в виде данных, система команд исполнителя, алгоритмический язык исполнителя;</li> <li>• основные средства записи и типы алгоритмов;</li> <li>• цели и составные части технологии нисходящего структурного проектирования алгоритмов и программ;</li> <li>• алгоритмические структуры, их основные свойства и приемы использования;</li> <li>• понятие алгоритмического модуля и его свойства;</li> <li>• принципы выделения и правила записи алгоритмических модулей;</li> <li>• семантику вызова алгоритмического модуля, в том числе, рекурсивного вызова, и механизмы реализации межмодульного интерфейса;</li> <li>• основные парадигмы программирования;</li> <li>• номенклатуру языков программирования различного уровня и их возможности.</li> </ul>	<ul style="list-style-type: none"> <li>• при решении задач практической информатики на основе парадигмы императивного программирования конкретизировать и описать основные понятия, результаты других научных дисциплин, данные и их типы средствами записи алгоритмов;</li> <li>• разрабатывать и записывать алгоритмы и программы в соответствии с технологией нисходящего структурного проектирования;</li> <li>• записывать алгоритмы с помощью схем;</li> <li>• анализировать структуру алгоритмов;</li> <li>• описывать алфавит и основные синтаксические конструкции языков с помощью нормальных форм Бэкуса-Наура.</li> </ul>	<ul style="list-style-type: none"> <li>• основные методами, способами и средствами переработки информации на основе парадигмы императивного программирования в соответствии с технологией нисходящего структурного проектирования.</li> </ul>
2	ОПК-8	способность использовать знания методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментом	<ul style="list-style-type: none"> <li>• составные части алгоритмического языка программирования (алфавит, синтаксис, семантика) и способы их описания;</li> <li>• концепцию типов данных;</li> <li>• структуры программ на языках программирования Бейсик, Паскаль и Си;</li> <li>• основные понятия алгоритмических языков программирования и их классификацию: идентификаторы, данные и их типы, операции, стандартные функции, выражения, операторы;</li> <li>• синтаксис и семантику основных операторов языков программирования Бейсик, Паскаль и Си (присваивания,</li> </ul>	<ul style="list-style-type: none"> <li>• разрабатывать и записывать алгоритмы и программы в соответствии с принципом структурности;</li> <li>• при решении учебной задачи конкретизировать и описать основные понятия, данные и их типы, организовать ввод и вывод, реализовать алгоритмические структуры средствами, по крайней мере, трех императивных языков программирования;</li> <li>• разрабатывать алгоритмы и программы обработки массивов;</li> </ul>	<ul style="list-style-type: none"> <li>• навыкам и проектирования, ввода, отладки и тестирования программ в средах, по крайней мере, трех императивных систем программирования.</li> </ul>

		<p>альными средствами, поддерживающими создание программно-го обеспечения</p>	<p>ввода-вывода, ветвления, циклов), способы реализации алгоритмических структур средствами языков программирования;</p> <ul style="list-style-type: none"> <li>• основные синтаксические конструкции, используемые в языках программирования Бейсик, Паскаль и Си для описания и обработки данных типа массив;</li> <li>• прием флаг;</li> <li>• синтаксис и семантику реализации алгоритмического модуля средствами языков Бейсик, Паскаль и Си;</li> <li>• способы передачи параметров при реализации межмодульного интерфейса;</li> <li>• синтаксис типов запись с фиксированными полями, указатель и типизированный файл в языке Паскаль;</li> <li>• синтаксис типов структура, указатель и файл в языке Си;</li> <li>• понятия статического и динамического объекта программы на языках Паскаль и Си;</li> <li>• синтаксис и семантику основных средств управления использованием динамической памятью в языках Паскаль и Си;</li> <li>• структуру данных файл и основные действия с файлами последовательного доступа;</li> <li>• синтаксис и семантику основных средств обработки файлов в СП Turbo-Pascal и Turbo-C++ и последовательность их использования для организации ввода и вывода.</li> </ul>	<ul style="list-style-type: none"> <li>• разрабатывать алгоритмы и программы обработки строк;</li> <li>• решать учебные задачи в соответствии с принципом модульности;</li> <li>• разрабатывать и использовать библиотеку пользователя в СП Turbo-Pascal и Turbo-C++;</li> <li>• разрабатывать простейшие алгоритмы и программы моделирования движения графического изображения;</li> <li>• использовать тип указатель для организации односвязного списка, очереди и стека;</li> <li>• решать учебные задачи по обработке данных типа запись, структура, файл и линейных динамических структур данных средствами систем программирования (СП) Turbo-Pascal и Turbo-C++.</li> </ul>	
--	--	---	---	---	--

## 2.5. Карта компетенций

КАРТА КОМПЕТЕНЦИЙ ДИСЦИПЛИНЫ					
НАИМЕНОВАНИЕ ДИСЦИПЛИНЫ: ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ					
<b>Цель дисциплины</b>		Целью освоения дисциплины «Информатика и программирование» является формирование у обучающихся общекультурных и общепрофессиональных компетенций в процессе изучения информатики и программирования для последующего применения в учебной и практической деятельности			
В процессе освоения данной дисциплины студент формирует и демонстрирует следующие					
Общепрофессиональные компетенции:					
КОМПЕТЕНЦИИ		Перечень компонентов	Технологии формирования	Форма оценочного средства	Уровни освоения компетенций
ИНДЕКС	ФОРМУЛИРОВКА				
Общепрофессиональные компетенции:					
ОПК-3	готовность анализировать проблемы и направления развития технологий программирования	<p><b>Знать:</b></p> <ul style="list-style-type: none"> <li>предмет и задачи исследования информатики как науки;</li> <li>сущность и значение информации в развитии современного информационного общества;</li> <li>этапы решения задачи на компьютере на основе парадигмы императивного программирования;</li> <li>понятие алгоритма и его основные свойства;</li> <li>основные понятия алгоритмической системы: представление информации в виде данных, система команд исполнителя, алгоритмический язык исполнителя;</li> <li>основные средства записи и типы алгоритмов;</li> <li>цели и составные части технологии нисходящего структурного проектирования алгоритмов и программ;</li> <li>алгоритмические структуры, их основные свойства и приемы использования;</li> <li>понятие алгоритмического модуля и его свойства;</li> <li>принципы выделения и правила записи алгоритмических модулей;</li> <li>семантику вызова алгоритмического модуля, в том числе, рекурсивного вызова, и механизмы реализации межмодульного интерфейса;</li> <li>основные парадигмы программирования;</li> <li>номенклатуру языков программирования различного уровня и их возможности.</li> </ul> <p><b>Уметь:</b></p> <ul style="list-style-type: none"> <li>при решении задач практической информатики на основе парадигмы императивного программирования конкретизировать и описать основные понятия, результаты других научных дисциплин, данные и их типы средствами записи алгоритмов;</li> </ul>	Путем проведения лекционных, лабораторных занятий, применения новых образовательных технологий, организации самостоятельной работы студентов	Лабораторные работы, зачет, экзамен	<p><b>Пороговый</b> Способен решать стандартные задачи получения, хранения, переработки информации в соответствии с технологией нисходящего структурного проектирования.</p> <p><b>Повышенный</b> Способен решать задачи получения, хранения, переработки информации повышенной сложности в соответствии с технологией нисходящего структурного проектирования.</p>

		<ul style="list-style-type: none"> <li>• разрабатывать и записывать алгоритмы и программы в соответствие с технологией нисходящего структурного проектирования;</li> <li>• записывать алгоритмы с помощью схем;</li> <li>• анализировать структуру алгоритмов;</li> <li>• описывать алфавит и основные синтаксические конструкции языков с помощью нормальных форм Бэкуса-Наура.</li> </ul> <p><b>Владеть:</b></p> <ul style="list-style-type: none"> <li>• основными методами, способами и средствами переработки информации на основе парадигмы императивного программирования в соответствии с технологией нисходящего структурного проектирования.</li> </ul>			
ОПК-8	<p>способность использовать знания методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание программного обеспечения</p>	<p><b>Знать:</b></p> <ul style="list-style-type: none"> <li>• составные части алгоритмического языка программирования (алфавит, синтаксис, семантика) и способы их описания;</li> <li>• концепцию типов данных;</li> <li>• структуры программ на языках программирования Бейсик, Паскаль и Си;</li> <li>• основные понятия алгоритмических языков программирования и их классификацию: идентификаторы, данные и их типы, операции, стандартные функции, выражения, операторы;</li> <li>• синтаксис и семантику основных операторов языков программирования Бейсик, Паскаль и Си (присваивания, ввода-вывода, ветвления, циклов), способы реализации алгоритмических структур средствами языков программирования;</li> <li>• основные синтаксические конструкции, используемые в языках программирования Бейсик, Паскаль и Си для описания и обработки данных типа массив;</li> <li>• прием флаг;</li> <li>• синтаксис и семантику реализации алгоритмического модуля средствами языков Бейсик, Паскаль и Си;</li> <li>• способы передачи параметров при реализации межмодульного интерфейса;</li> <li>• синтаксис типов запись с фиксированными полями, указатель и типизированный файл в языке Паскаль;</li> <li>• синтаксис типов структура, указатель и файл в языке Си;</li> <li>• понятия статического и динамического объекта программы на языках Паскаль и Си;</li> <li>• синтаксис и семантику основных средств управления использованием динамической памятью в языках Паскаль и Си;</li> <li>• структуру данных файл и основные действия с файлами последовательного доступа;</li> <li>• синтаксис и семантику основных средств обработки файлов в СП Turbo-</li> </ul>	<p>Путем проведения лекционных, лабораторных занятий, применения новых образовательных технологий, организации самостоятельной работы студентов</p>	<p>Лабораторные работы, зачет, экзамен</p>	<p><b>Пороговый</b> Способен решать стандартные задачи средствами императивных систем программирования</p> <p><b>Повышенный</b> Способен решать задачи повышенной сложности средствами императивных систем программирования</p>

		<p>Pascal и Turbo-C++ и последовательность их использования для организации ввода и вывода.</p> <p><b>Уметь:</b></p> <ul style="list-style-type: none"> <li>• разрабатывать и записывать алгоритмы и программы в соответствие с принципом структурности;</li> <li>• при решении учебной задачи конкретизировать и описать основные понятия, данные и их типы, организовать ввод и вывод, реализовать алгоритмические структуры средствами, по крайней мере, трех императивных языков программирования;</li> <li>• разрабатывать алгоритмы и программы обработки массивов;</li> <li>• разрабатывать алгоритмы и программы обработки строк;</li> <li>• решать учебные задачи в соответствие с принципом модульности;</li> <li>• разрабатывать и использовать библиотеку пользователя в СП Turbo-Pascal и Turbo-C++;</li> <li>• разрабатывать простейшие алгоритмы и программы моделирования движения графического изображения;</li> <li>• использовать тип указатель для организации односвязного списка, очереди и стека;</li> <li>• решать учебные задачи по обработке данных типа запись, структура, файл и линейных динамических структур данных средствами систем программирования (СП) Turbo-Pascal и Turbo-C++.</li> </ul> <p><b>Владеть:</b></p> <ul style="list-style-type: none"> <li>• навыками проектирования, ввода, отладки и тестирования программ в средах, по крайней мере, трех императивных систем программирования.</li> </ul>			
--	--	--	--	--	--

## ОСНОВНАЯ ЧАСТЬ

### 1. Объем дисциплины и виды учебной работы

Вид учебной работы	Всего часов	Семестры			
		№ 1 часов	№ 2 часов	№ 3 часов	
1	2	3	4	5	
<b>1. Контактная работа обучающихся с преподавателем (по видам учебных занятий) (всего)</b>	<b>288</b>	<b>90</b>	<b>108</b>	<b>90</b>	
В том числе:					
Лекции (Л)	90	36	36	18	
Лабораторные работы (ЛР)	198	54	72	72	
<b>2. Самостоятельная работа студента (всего)</b>	<b>360</b>	<b>90</b>	<b>144</b>	<b>126</b>	
В том числе:					
<i>СРС в семестре</i>	<b>288</b>	<b>90</b>	<b>108</b>	<b>90</b>	
Изучение литературы и других источников	77	28	26	23	
Подготовка к выполнению лабораторных работ	125	32	51	42	
Подготовка к защите лабораторных работ	86	30	31	25	
<i>СРС в период сессии</i>	<b>72</b>	-	<b>36</b>	<b>36</b>	
Вид промежуточной аттестации	зачет (З)	+			
	экзамен (Э)		+	+	
<b>ИТОГО: общая трудоемкость</b>	<b>часов</b>	<b>648</b>	<b>180</b>	<b>252</b>	<b>216</b>
	зач. ед.	<b>18</b>	<b>5</b>	<b>7</b>	<b>6</b>

Дисциплина частично реализуется с применением дистанционных образовательных технологий с использованием платформы Microsoft Teams, ЭИОС Moodle, корпоративной электронной почты.

### 2. Содержание дисциплины

#### 2.1. Содержание разделов дисциплины

№ семестра	№ раздела	Наименование раздела дисциплины	Содержание раздела в дидактических единицах
1	2	3	4
1	1	Информатика как наука и как вид практической деятельности	История информатики, место информатики в системе наук. Теоретическая и прикладная информатика. Принципы фон Неймана как основа структуры персонального компьютера. Позиционные системы счисления. Представление в компьютере целых чисел. Представление в компьютере действительных чисел. Алгоритмизация и программирование как составные части информатики. Основные этапы компьютерного решения задач на основе парадигмы императивного программирования
1	2	Алгоритмическая система и ее составные части	Алгоритмическая система как совокупность средств и понятий. Понятие алгоритма и исполнителя алгоритма. Представление информации в виде данных: понятие о вводе/выводе; входные, выходные и промежуточные данные. Система команд исполнителя. Константы, переменные, операции, функции и выражения. Понятие типа данных. Алгоритмический язык исполнителя. Средства записи алгоритмов. Схемы алгоритмов



1	2	3	4
1	3	Основные понятия и простейшие средства алгоритмических языков программирования	Программирование как этап решения задачи на компьютере. Понятие языка программирования высокого уровня. Составные части языка высокого уровня: алфавит, синтаксис, семантика. Метаязыки описания конструкций языков программирования. Расширенные Бэкуса-Наура формы, примеры метаформул. Структуры программ и концепции типов данных в алгоритмических языках на примере языков Qbasic и Turbo-Pascal. Оператор как логически завершенная конструкция алгоритмического языка программирования. Различные классификации операторов: исполняемые и неисполняемые, простые и составные. Различные классификации типов данных: простые и структурированные, стандартные и определяемые пользователем. Основные простые типы данных в языках Qbasic и Pascal (кроме логического): идентификаторы стандартных типов, диапазоны, операции и функции, синтаксис типа диапазон. Особенности целочисленной и действительной арифметики в системах программирования (СП) Qbasic и Turbo-Pascal. Синтаксис разделов программы на языке Turbo-Pascal, понятие порядкового типа в языке Pascal. Синтаксис и семантика оператора присваивания, знакомство с операторами ввода/вывода в языках Qbasic и Pascal. Общая характеристика императивных СП Qbasic и Turbo-Pascal: основное меню, окна редактирования и результатов исполнения программы, диалоговые окна, встроенная система помощи. Написание, ввод, отладка и тестирование программ линейной структуры в СП Qbasic и Turbo-Pascal
1	4	Технология нисходящего структурного проектирования. Разработка разветвляющихся алгоритмов и программ в соответствии с принципом структурности	Цели и составные части технологии нисходящего структурного проектирования. Взаимосвязь принципов нисходящего проектирования, модульности и структурности. Алгоритмические структуры как основа структурного кодирования. Базовые алгоритмические структуры: следование и развилка. Вложение (суперпозиция) алгоритмических структур как основной прием принципа структурности. Присоединение алгоритмических структур. Правила пунктуации в программах на алгоритмическом языке. Составной оператор в языке Pascal. Логический тип данных в алгоритмических языках Qbasic и Pascal. Операции отношения и логические операции. Алгоритмические структуры, реализующие ветвления. Организация ветвлений средствами алгоритмических языков: синтаксис и семантика строчного и блочного условных операторов в Qbasic, условного оператора в Pascal, операторов выбора в языках Qbasic и Pascal. Понятие о приеме программирования "флаг". Примеры алгоритмов и программ разветвляющейся структуры. Разработка схем алгоритмов разветвляющейся структуры. Написание, ввод, отладка и тестирование программ разветвляющейся структуры в СП Qbasic и Turbo-Pascal
1	5	Разработка циклических алгоритмов и программ в соответствии с принципом структурности	Базовая алгоритмическая структура цикл с предусловием. Цикл с постусловием. Проблема заикливания. Цикл с параметром. Синтаксис и семантика операторов циклов в языках Qbasic и Pascal. Особенности системной организации циклов с параметром в СП Qbasic и Turbo-Pascal. Рекомендации по программированию циклов. Примеры циклических алгоритмов и программ. Разработка схем алгоритмов циклической структуры. Написание, ввод, отладка и тестирование программ циклической структуры в СП Qbasic и Turbo-Pascal
1	6	Ввод с клавиатуры и вывод на монитор средствами алгоритмических языков	Понятие о текстовом режиме монитора, текущее положение курсора. Синтаксис и семантика операторов ввода с клавиатуры в СП Qbasic и Turbo-Pascal. Особенности ввода с клавиатуры символьной и строковой информации в СП Turbo-Pascal. Общая характеристика операторов вывода на монитор в императивных алгоритмических СП. Синтаксис оператора PRINT в Qbasic. Организация слитного и зонального вывода средствами оператора PRINT, примеры. Особенности вывода значений числового и строкового типа. Организация форматированного вывода средствами оператора PRINT USING в Qbasic: синтаксис, символы форматирования, основные форматы для вывода значений строкового и числового типа, семантика. Синтаксис и семантика оператора вывода на монитор в СП Turbo-Pascal.

1	2	3	4
			Форматированный вывод текстовой и числовой информации на монитор в СП Turbo-Pascal. Диалоговые программы. Дружественность интерфейса с пользователем. Примеры организации форматированного вывода на монитор. Разработка схем алгоритмов, написание, ввод, отладка и тестирование программ с форматированным выводом на монитор в СП Qbasic и Turbo-Pascal
1	7	Структурированный тип данных массив	Синтаксис и семантика оператора резервирования памяти в Qbasic, примеры. Особенности распределения памяти в СП Qbasic. Синтаксис определения типа массив в языке Pascal. Рекомендации по описанию переменной типа массив и понятие полной совместимости типов в языке Pascal, примеры. Использование переменной типа массив в операторе присваивания. Элемент массива (переменная с индексами). Основные алгоритмы и программы обработки числовых массивов: поиск максимума (минимума), поиск элементов с заданными свойствами, сортировка методом пузырька, модифицированный алгоритм сортировки методом пузырька. Примеры алгоритмов и программ обработки массивов, в том числе с использованием приема "флаг". Разработка схем алгоритмов обработки массивов. Написание, ввод, отладка и тестирование программ обработки массивов в СП Qbasic и Turbo-Pascal
2	8	Символьный и строковый типы данных	Стандартный идентификатор строкового типа в языках Qbasic и Pascal. Строковый тип в языке Pascal как массив символов. Операция конкатенации в Qbasic и Turbo-Pascal. Синтаксис и семантика функции MID\$ и оператора MID\$. Функции и процедуры для работы со строками в языке Turbo-Pascal. Примеры алгоритмов и программ обработки данных строкового типа. Написание, ввод, отладка и тестирование программ обработки строк в СП Qbasic и Turbo-Pascal
2	9	Разработка алгоритмов в соответствии с принципом модульности	Понятие алгоритмического модуля, его свойства и правила выделения. Концепция "черного ящика". Правила записи алгоритмического модуля: спецификация и тело модуля. Входные и выходные формальные параметры. Блок вызова алгоритмического модуля. Фактические параметры. Семантика вызова алгоритмического модуля и механизм реализации межмодульного интерфейса. Понятие рекурсии. Семантика рекурсивного вызова. Примеры простейших алгоритмов, составленных в соответствии с принципом модульности. Разработка схем алгоритмов в соответствии с принципом модульности
2	10	Реализация принципа модульности средствами алгоритмических языков программирования	Способы записи алгоритмического модуля средствами алгоритмических языков программирования: процедуры и функции. Синтаксис процедуры, синтаксис и семантика оператора процедуры в языке Qbasic. Способы передачи параметров значением и ссылкой. Синтаксис процедуры, синтаксис и семантика оператора процедуры в языке Pascal. Параметры значения и параметры переменные. Рекомендации по выбору способа передачи параметров. Примеры простейших программ, составленных в соответствии с принципом модульности. Синтаксис функции в языках Qbasic и Pascal. Рекомендации по использованию функций в императивных СП Qbasic и Turbo-Pascal. Примеры простейших программ с использованием функций пользователя. Синтаксис процедурного типа в языке Pascal. Рекомендации по использованию процедурного типа для решения задач: параметры процедуры и параметры функции. Пример алгоритма и программы решения задачи с использованием процедурного типа. Примеры алгоритма и программ решения задачи рекурсивной природы. Понятие о библиотеке СП Turbo-Pascal. Обзор стандартных модулей. Синтаксис раздела модулей программы и особенности трансляции процедур и функций модуля СП Turbo-Pascal. Синтаксис модуля пользователя и рекомендации по организации библиотеки пользователя в СП Turbo-Pascal. Длинная целочисленная арифметика. Примеры программ сложения и вычитания натуральных чисел с числом разрядов больше десяти. Написание, ввод, отладка и тестирование программ в соответствии с принципом модульности в СП Qbasic и Turbo-Pascal

1	2	3	4
2	11	Структурированный тип данных запись	Синтаксис типа запись с фиксированными полями в языке Pascal. Рекомендации по описанию переменной типа запись, примеры. Использование переменной типа запись в операторе присваивания. Селектор записи (переменная с точкой). Синтаксис оператора присоединения. Примеры алгоритмов и программ обработки данных типа запись. Разработка схем алгоритмов обработки данных типа запись. Написание, ввод, отладка и тестирование программ обработки данных типа запись в соответствии с принципом модульности в СП Turbo-Pascal
2	12	Тип данных указатель	Понятия статических и динамических объектов программы на языке Паскаль. Синтаксис типа указатель (ссылочный тип) в языке Pascal. Рекомендации по описанию переменной типа указатель, примеры. Синтаксис и семантика оператора резервирования (захвата) памяти под динамическую переменную в языке Pascal, примеры. Использование переменной типа указатель в операторе присваивания и в выражениях отношения с операциями = и <. Динамическая переменная (переменная с крышкой). Синтаксис и семантика освобождения области памяти динамической переменной в языке Pascal. Понятие динамических структур данных. Использование типа указатель для организации динамического односвязного списка по принципам ссылки вперед и ссылки назад. Примеры алгоритмов и программ создания динамических односвязных списков. Алгоритмы вставки (добавления), удаления и поиска элемента динамического односвязного списка. Структуры данных на основе односвязных списков: очередь и стек. Рекомендации по созданию и обработке динамических структур данных очередь и стек с примерами фрагментов алгоритмов и программ. Разработка схем алгоритмов создания и обработки динамических структур данных. Написание, ввод, отладка и тестирование программ создания и обработки динамических структур данных в соответствии с принципом модульности в СП Turbo-Pascal
2	13	Структурированный тип данных файл	Понятие файла, файлы данных различного доступа. Основные действия с файлами данных последовательного доступа: запись (вывод в файл) и чтение (ввод из файла). Обзор файловых типов данных в языке Pascal, синтаксис типизированного файлового типа. Рекомендации по описанию переменной типа файл, примеры. Синтаксис и семантика основных операторов для работы с типизированными файлами данных в СП Turbo-Pascal: связь файловой переменной с файлом на внешнем устройстве, открытие файла для записи и чтения, вывод в файл и ввод из файла, закрытие файла. Функция конец типизированного файла. Рекомендации по корректной обработке типизированных файлов с примерами фрагментов алгоритмов и программ. Разработка схем алгоритмов создания и обработки типизированных файлов. Написание, ввод, отладка и тестирование программ создания и обработки типизированных файлов в соответствии с принципом модульности в СП Turbo-Pascal
2	14	Обзор графических возможностей систем программирования (СП) Qbasic и Turbo-Pascal	Понятие о графическом режиме монитора, текущее положение графического курсора. Операторы установки параметров окна результатов и задания текущего цвета изображения в языке Qbasic. Обзор стандартных средств языка Qbasic для изображения графических примитивов: точек, отрезков, прямоугольников, окружностей и эллипсов, дуг окружностей и эллипсов. Операторы стандартных процедур модуля Graph библиотеки СП Turbo-Pascal для определения подходящего графического драйвера, инициализации и прекращения графического режима и задания текущего цвета изображения. Обзор стандартных процедур модуля Graph для изображения графических примитивов: точек, отрезков, прямоугольников, окружностей и эллипсов, дуг окружностей и эллипсов. Простейший алгоритм моделирования движения графического изображения (анимация). Написание, ввод, отладка и тестирование программ моделирования движения графического изображения в соответствии с принципом модульности в СП Qbasic и Turbo-Pascal

1	2	3	4
3	15	Парадигмы программирования	Основные парадигмы программирования: императивная, событийно-управляемая (объектно-ориентированная и параллельная), декларативная (функциональная и логическая). Архитектура и возможности семейства языков высокого уровня
3	16	Общая характеристика языков программирования семейства Си	Особенности языков программирования семейства Си. Структура простейшей программы на языке Си: раздел препроцессора и раздел функций. Синтаксис разделов программы, составной оператор. Понятие о библиотеке языка Си. Трансляция программ в СП (СП) семейства Си. Основные простые типы данных в языке Си: идентификаторы типов, модификаторы типов, диапазоны значений, синтаксис и семантика оператора резервирования памяти, инициализация переменных, операции и функции, преобразование типов. Некоторые особенности операций языка Си: перегрузка операции деления, операция присваивания и возможные варианты записи выражений с использованием операции присваивания, операции инкремента и декремента в префиксной и постфиксной форме, операции взятия адреса и содержимое по указанному адресу. Знакомство с операторами потокового ввода/вывода. Общая характеристика императивной СП Turbo-C++: основное меню, окна редактирования и результатов исполнения программы, диалоговые окна, встроенная система помощи
3	17	Разработка разветвляющихся алгоритмов и программ в СП Turbo-C++	Интерпретация данных целого типа как данных логического типа в языке Си. Операции отношения и логические операции. Организация развилки средствами языка Си: синтаксис и семантика условных оператора и операции. Операция запятая. Примеры организации развилки средствами Си. Синтаксис и семантика оператора переключателя. Организация посредством переключателя структуры выбор, пример. Разработка схем алгоритмов разветвляющейся структуры. Написание, ввод, отладка и тестирование программ разветвляющейся структуры в СП Turbo-C++
3	18	Разработка циклических алгоритмов и программ в СП Turbo-C++	Синтаксис и семантика операторов циклов в языке Си: циклы с предусловием и постусловием, цикл for. Особенности системной организации цикла for в СП Turbo-C++. Рекомендации по программированию циклов. Примеры циклических алгоритмов и программ. Разработка схем алгоритмов циклической структуры. Написание, ввод, отладка и тестирование программ циклической структуры в СП Turbo-C++
3	19	Ввод с клавиатуры и вывод на монитор средствами C++	Синтаксис и семантика операторов потокового ввода/вывода. Манипуляторы. Примеры организации потокового ввода/вывода. Синтаксис и семантика функций форматированного ввода/вывода. Синтаксис формата ввода, символы преобразования и модификаторы. Примеры организации форматированного ввода. Синтаксис формата вывода, управляющие символы. Примеры организации форматированного вывода. Написание, ввод, отладка и тестирование программ с форматированным выводом на монитор в СП Turbo-C++
3	20	Массивы в языке Си	Синтаксис и семантика оператора резервирования памяти под массив, примеры. Идентификатор массива как указатель на область памяти, захваченной под массив. Использование переменной типа массив в выражении с операцией присваивания. Особенности распределения памяти под элементы массива и инициализация массивов в СП семейства Си. Традиционная индексная и альтернативная, с помощью операции "*", адресация элементов массива, примеры. Примеры обработки массивов средствами языка Си. Разработка схем алгоритмов обработки массивов. Написание, ввод, отладка и тестирование программ обработки массивов в СП Turbo-C++
3	21	Строки в языке Си	Строки как массивы символов. Особенности инициализации и организации ввода строк в языке Си. Стандартные функции обработки строк в языке Си. Алгоритмы копирования, удаления и замены части строки средствами языка Си, примеры. Разработка схем алгоритмов обработки строк. Написание, ввод, отладка и тестирование программ обработки строковых данных в СП Turbo-C++

1	2	3	4
3	22	Реализация принципа модульности средствами языка Си	Синтаксис функции в языке Си. Формальные и фактические параметры. Способы передачи параметра простого типа. Рекомендации по использованию функций в программах на C++, примеры. Параметры массивы и параметры функции, примеры. Функции с аргументами по умолчанию, примеры. Перегрузка функций, пример. Классы памяти в языке Си. Рекомендации по организации простейшей библиотеки пользователя средствами СП Turbo-C++. Разработка схем алгоритмов в соответствии с принципом модульности. Написание, ввод, отладка и тестирование программ в соответствии с принципом модульности в СП Turbo-C++
3	23	Тип данных структура	Синтаксис типа структура в языках семейства Си. Оператор определения типа пользователя, пример. Определение типа структура с помощью тэга. Рекомендации по описанию переменной типа структура, примеры. Использование переменной типа структура в выражении с операцией присваивания. Член структуры (идентификатор с точкой). Разработка схем алгоритмов обработки данных типа структура. Написание, ввод, отладка и тестирование программ обработки данных типа структура в соответствии с принципом модульности в СП Turbo-C++
3	24	Динамические переменные в языках семейства Си	Синтаксис типа указатель (ссылочный тип) в языках семейства Си. Рекомендации по описанию переменной типа указатель. Синтаксис и семантика оператора захвата памяти под динамическую переменную. Идентификатор динамической переменной в языках семейства Си (идентификатор со звездочкой). Синтаксис и семантика освобождения области памяти динамической переменной. Синтаксис и семантика операторов захвата и освобождения области памяти для динамического массива. Использование типа указатель для организации односвязного списка, определение типа элемента односвязного списка с помощью тэга. Адресация члена динамической переменной типа структура с помощью литеры алфавита “->” (идентификатор со стрелкой). Рекомендации по организации вставки (добавления), удаления и поиска элементов линейных динамических структур данных. Примеры алгоритмов создания и обработки линейных динамических структур данных и соответствующих программ на языке C++. Разработка схем алгоритмов создания и обработки линейных динамических структур данных. Написание, ввод, отладка и тестирование программ создания и обработки линейных динамических структур данных в соответствии с принципом модульности в СП Turbo-C++
3	25	Тип файл в СП Turbo-C++	Организация файлов данных в языках семейства Си. Синтаксис и семантика захвата памяти под указатель на буфер файла. Синтаксис и семантика функции открытия файла. Атрибуты режима по умолчанию для записи (вывода), чтения (ввода) и добавления данных. Атрибуты режимов текстового и бинарного файла данных. Синтаксис и семантика функций форматированной записи данных в файл и форматированного чтения данных из файла. Функция конец файла в СП Turbo-C++. Рекомендации для СП Turbo-C++ по корректному открытию файла и чтению данных из файла, примеры. Синтаксис и семантика функции закрытия файла. Примеры алгоритма и программы обработки файла данных, в которых для хранения промежуточных результатов используется линейная динамическая структура. Разработка схем алгоритмов обработки файлов данных. Написание, ввод, отладка и тестирование программ создания и текстовых файлов в соответствии с принципом модульности в СП Turbo-C++

## 2.2. Разделы дисциплины, виды учебной деятельности и формы контроля

№ семестра	№ раздела	Наименование раздела дисциплины	Виды учебной деятельности, включая самостоятельную работу студентов (в часах)				Формы текущего контроля успеваемости (по неделям семестра)
			Л	ЛР	СРС	всего	
1	2	3	4	5	6	7	8
1	1	Информатика как наука и как вид практической деятельности	2		4	6	1 неделя: лабораторная работа №1
	2	Алгоритмическая система и ее составные части			2	2	
	3	Основные понятия и простейшие средства алгоритмических языков программирования	3	3	6	12	
	4	Технология нисходящего структурного проектирования. Разработка разветвляющихся алгоритмов и программ в соответствии с принципом структурности	8	12	20	40	2-5 неделя: лабораторная работа №2
	5	Разработка циклических алгоритмов и программ в соответствии с принципом структурности	9	18	22	47	6-7 неделя: лабораторная работа №3 8-9 неделя: лабораторная работа №4 10-11 неделя: лабораторная работа №5
	6	Ввод с клавиатуры и вывод на монитор средствами алгоритмических языков	4	9	12	25	12-14 неделя: лабораторная работа №6
	7	Структурированный тип данных массив	10	12	24	36	15-18 неделя: лабораторная работа №7
<b>ИТОГО 1 семестр</b>			<b>36</b>	<b>54</b>	<b>90</b>	<b>180</b>	<b>Зачет</b>
2	8	Символьный и строковый типы данных	6	20	26	52	1-2 неделя: лабораторная работа №8 3-5 неделя: лабораторная работа №9
	9	Разработка алгоритмов в соответствии с принципом модульности	2	8	10	20	6-7 неделя: лабораторная работа №10
	10	Реализация принципа модульности средствами алгоритмических языков программирования	10	20	30	60	8-12 неделя: лабораторная работа №11
	11	Структурированный тип данных запись	4	4	8	16	13 неделя: лабораторная работа №12
	12	Тип данных указатель	8	8	16	32	15-16 неделя: лабораторная работа №13
	13	Структурированный тип данных файл	4	4	8	16	

1	2	3	4	5	6	7	8
	14	Обзор графических возможностей систем программирования (СП) Qbasic и Turbo-Pascal	2	8	10	20	17-18 неделя: лабораторная работа №14
		Разделы дисциплины 1 -14			36	36	Экзамен
		<b>ИТОГО 2 семестр</b>	<b>36</b>	<b>72</b>	<b>144</b>	<b>252</b>	
3	15	Парадигмы программирования	1		1	2	1 неделя: лабораторная работа №15
	16	Общая характеристика языков программирования семейства Си	1	2	3	6	
	17	Разработка разветвляющихся алгоритмов и программ в СП Turbo-C++	2	10	12	24	2-3 неделя: лабораторная работа №16
	18	Разработка циклических алгоритмов и программ в СП Turbo-C++	2	8	10	20	4 неделя: лабораторная работа №17 5 неделя: лабораторная работа №18
	19	Ввод с клавиатуры и вывод на монитор средствами C++	1	8	9	18	6-7 неделя: лабораторная работа №19
	20	Массивы в языке Си	2	8	10	20	8-9 неделя: лабораторная работа №20
	21	Строки в языке Си	2	8	10	20	10-11 неделя: лабораторная работа №21
	22	Реализация принципа модульности средствами языка Си	3	16	19	38	12-16 неделя: лабораторная работа №22
	23	Тип данных структура	1	4	5	10	
	24	Динамические переменные в языках семейства Си	2	6	8	16	17 неделя: лабораторная работа №23 18 неделя: лабораторная работа №24
	25	Тип файл в СП Turbo-C++	1	2	3	6	
		Разделы дисциплины 15 -25			36	36	Экзамен
		<b>ИТОГО 3 семестр</b>	<b>18</b>	<b>72</b>	<b>126</b>	<b>216</b>	
		<b>ИТОГО</b>	<b>90</b>	<b>198</b>	<b>360</b>	<b>648</b>	

### 2.3. Лабораторный практикум

№ семестра	№ раздела	Наименование раздела дисциплины	Наименование лабораторных работ	Всего часов
1	2	3	4	5
1	1	Информатика как наука и как вид практической деятельности	ЛР вводная. Знакомство с императивными системами программирования QBasic и Turbo-Pascal. Ввод, отладка и тестирование простейших программ	1

1	2	3	4	5
1	2	Алгоритмическая система и ее составные части	ЛР №1. <i>Написание, ввод, отладка и тестирование программ линейной структуры</i>	3
	3	Основные понятия и простейшие средства алгоритмических языков программирования		
1	4	Технология нисходящего структурного проектирования. Разработка разветвляющихся алгоритмов и программ в соответствии с принципом структурности	ЛР №2. <i>Написание, ввод, отладка и тестирование алгоритмов и программ разветвляющейся структуры</i>	6
			ЛР №3. <i>Написание, ввод, отладка и тестирование разветвляющихся алгоритмов и программ с использованием структуры выбор</i>	6
1	5	Разработка циклических алгоритмов и программ в соответствии с принципом структурности	ЛР №4. <i>Написание, ввод, отладка и тестирование циклических алгоритмов и программ</i>	9
			ЛР №5. <i>Написание, ввод, отладка и тестирование циклических алгоритмов и программ целочисленной арифметики</i>	9
1	6	Ввод с клавиатуры и вывод на монитор средствами алгоритмических языков	ЛР №6. <i>Написание, ввод, отладка и тестирование циклических алгоритмов и программ с форматированным выводом на монитор</i>	9
1	7	Структурированный тип данных массив	ЛР №7. <i>Написание, ввод, отладка и тестирование алгоритмов и программ обработки числовых массивов</i>	12
<b>ИТОГО 1 семестр</b>				<b>54</b>
2	8	Символьный и строковый типы данных	ЛР №8. <i>Написание, ввод, отладка и тестирование алгоритмов и программ обработки строк</i>	8
			ЛР №9. <i>Написание, ввод, отладка и тестирование алгоритмов и программ обработки массивов строк</i>	12
2	9	Разработка алгоритмов в соответствии с принципом модульности	ЛР №10. <i>Написание, ввод, отладка и тестирование алгоритмов и программ в соответствии с принципом модульности</i>	8
2	10	Реализация принципа модульности средствами алгоритмических языков программирования	ЛР №11. <i>Написание, ввод, отладка и тестирование алгоритмов и программ обработки массивов в соответствии с принципом модульности</i>	12
2	11	Структурированный тип данных запись	ЛР №12. <i>Написание, ввод, отладка и тестирование алгоритмов и программ создания и обработки односвязных списков, записей и типизированных файлов в СП Turbo-Pascal</i>	16
	12	Тип данных указатель		
2	13	Структурированный тип данных файл	ЛР №13. <i>Написание, ввод, отладка и тестирование алгоритмов и программ длинной арифметики в соответствии с принципом модульности</i>	8
2	14	Обзор графических возможностей систем программирования (СП) Qbasic и Turbo-Pascal	ЛР №14. <i>Написание, ввод, отладка и тестирование алгоритмов и программ моделирования движения графического изображения в соответствии с принципом модульности</i>	8
<b>ИТОГО 2 семестр</b>				<b>72</b>
3	16	Общая характеристика языков программирования семейства Си	ЛР №15. <i>Знакомство с императивной системой программирования Turbo-C++. Написание, ввод, отладка и тестирование алгоритмов и программ разветвляющейся структуры в СП Turbo-C++</i>	4



1	2	3	4	5
3	17	Разработка разветвляющихся алгоритмов и программ в СП Turbo-C++	ЛР №16. Написание, ввод, отладка и тестирование разветвляющихся алгоритмов и программ с использованием структуры выбор в СП Turbo-C++	8
3	18	Разработка циклических алгоритмов и программ в СП Turbo-C++	ЛР №17. Написание, ввод, отладка и тестирование циклических алгоритмов и программ в СП Turbo-C++	4
			ЛР №18. Написание, ввод, отладка и тестирование циклических алгоритмов и программ целочисленной арифметики в СП Turbo-C++	4
3	19	Ввод с клавиатуры и вывод на монитор средствами C++	ЛР №19. Написание, ввод, отладка и тестирование циклических алгоритмов и программ с форматированным выводом на монитор в СП Turbo-C++	8
3	20	Массивы в языке Си	ЛР №20. Написание, ввод, отладка и тестирование алгоритмов и программ обработки массивов в СП Turbo-C++	8
3	21	Строки в языке Си	ЛР №21. Написание, ввод, отладка и тестирование алгоритмов и программ обработки строковых данных в СП Turbo-C++	8
3	22	Реализация принципа модульности средствами языка Си	ЛР №22. Написание, ввод, отладка и тестирование алгоритмов и программ в соответствие с принципом модульности в СП Turbo-C++	8
3	23	Тип данных структура	ЛР №23. Написание, ввод, отладка и тестирование алгоритмов и программ создания и обработки односвязных списков, структур и файлов в СП Turbo-C++	12
	24	Динамические переменные в языках семейства Си		
	25	Тип файл в СП Turbo-C++		
		<b>ИТОГО 3 семестр</b>		<b>72</b>
		<b>ИТОГО</b>		<b>198</b>

### 3. Самостоятельная работа студента

#### 3.1. Виды СРС

№ семестра	№ раздела	Наименование раздела дисциплины	Виды СРС	Всего часов
1	2	3	4	5
1	1	Информатика как наука и как вид практической деятельности	Изучение литературы и других источников	2
			Подготовка к выполнению лабораторной работы	2
1	2	Алгоритмическая система и ее составные части	Изучение литературы и других источников	1
			Подготовка к выполнению лабораторной работы	1
1	3	Основные понятия и простейшие средства алгоритмических языков программирования	Изучение литературы и других источников	2
			Подготовка к выполнению лабораторной работы	2
			Подготовка к защите лабораторной работы	2
1	4	Технология нисходящего структурного проектирования. Разработка разветвляющихся алгоритмов и программ в соответствии с принципом структурности	Изучение литературы и других источников (ЛР №2)	3
			Подготовка к выполнению лабораторной работы (ЛР №2)	3
			Подготовка к защите лабораторной работы (ЛР №2)	4
			Изучение литературы и других источников (ЛР №3)	3
			Подготовка к выполнению лабораторной работы (ЛР №3)	3
			Подготовка к защите лабораторной работы (ЛР №3)	4

1	5	Разработка циклических алгоритмов и программ в соответствии с принципом структурности	Изучение литературы и других источников (ЛР №4)	3
			Подготовка к выполнению лабораторной работы (ЛР №4)	4
			Подготовка к защите лабораторной работы (ЛР №4)	4
			Изучение литературы и других источников (ЛР №5)	3
			Подготовка к выполнению лабораторной работы (ЛР №5)	4
			Подготовка к защите лабораторной работы (ЛР №5)	4
1	6	Ввод с клавиатуры и вывод на монитор средствами алгоритмических языков	Изучение литературы и других источников (основная и дополнительная литература)	2
			Изучение литературы и других источников (конспекты лекций и ресурсы компьютерных сетей)	3
			Подготовка к выполнению лабораторной работы	3
			Подготовка к защите лабораторной работы (ввод и вывод в СП Qbasic)	2
			Подготовка к защите лабораторной работы (ввод и вывод в СП Turbo-Pascal)	2
1	7	Структурированный тип данных массив	Изучение литературы и других источников (основная и дополнительная литература)	3
			Изучение литературы и других источников (конспекты лекций и ресурсы компьютерных сетей)	3
			Подготовка к выполнению лабораторной работы (разработка схемы алгоритма)	4
			Подготовка к выполнению лабораторной работы (разработка программы в СП Qbasic)	3
			Подготовка к выполнению лабораторной работы (разработка программы в СП Turbo-Pascal)	3
			Подготовка к защите лабораторной работы (массивы в СП Qbasic)	4
			Подготовка к защите лабораторной работы (массивы в СП Turbo-Pascal)	4
			<b>ИТОГО 1 семестр</b>	<b>90</b>
2	8	Символьный и строковый типы данных	Изучение литературы и других источников (ЛР №8)	3
			Подготовка к выполнению лабораторной работы (разработка схемы алгоритма ЛР №8)	3
			Подготовка к выполнению лабораторной работы (разработка программ ЛР №8)	3
			Подготовка к защите лабораторной работы (ЛР №8)	4
			Изучение литературы и других источников (ЛР №9)	3
			Подготовка к выполнению лабораторной работы (разработка схемы алгоритма ЛР №9)	4
			Подготовка к выполнению лабораторной работы (разработка программ ЛР №9)	3
			Подготовка к защите лабораторной работы (ЛР №9)	3
2	9	Разработка алгоритмов в соответствии с принципом модульности	Изучение литературы и других источников	2
			Подготовка к выполнению лабораторной работы (разработка схемы алгоритм)	2
			Подготовка к выполнению лабораторной работы (разработка программ)	2
			Подготовка к защите лабораторной работы (свойства алгоритмического модуля)	2
			Подготовка к защите лабораторной работы (оформление алгоритмического модуля)	2
2	10	Реализация принципа модульности средствами алгоритмических языков программирования	Изучение литературы и других источников (ЛР №10)	3
			Подготовка к выполнению лабораторной работы (разработка схемы алгоритма ЛР №10)	3
			Подготовка к выполнению лабораторной работы (разработка программ ЛР №10)	3
			Подготовка к защите лабораторной работы (реализация принципа модульности в СП Qbasic ЛР №10)	2
			Подготовка к защите лабораторной работы (реализация принципа модульности в СП Turbo-Pascal ЛР №10)	3

			Изучение литературы и других источников (основная и дополнительная литература ЛР №11)	2
			Изучение литературы и других источников (конспекты лекций и ресурсы компьютерных сетей ЛР №11)	3
			Подготовка к выполнению лабораторной работы (разработка схемы алгоритма ЛР №11)	3
			Подготовка к выполнению лабораторной работы (разработка программ ЛР №11)	3
			Подготовка к защите лабораторной работы (ввод из блока данных в СП Qbasic ЛР №11)	2
			Подготовка к защите лабораторной работы (типизированные константы в СП Turbo-Pascal ЛР №11)	3
2	11	Структурированный тип данных запись	Изучение литературы и других источников (основная и дополнительная литература)	2
			Изучение литературы и других источников (конспекты лекций и ресурсы компьютерных сетей)	2
			Подготовка к выполнению лабораторной работы (разработка структуры данных)	2
			Подготовка к выполнению лабораторной работы (разработка описаний)	2
2	12	Тип данных указатель	Подготовка к выполнению лабораторной работы (разработка схем алгоритмов)	2
			Подготовка к выполнению лабораторной работы (разработка описаний)	3
			Подготовка к выполнению лабораторной работы (разработка программного модуля)	3
			Подготовка к защите лабораторной работы (тип указатель)	2
			Подготовка к защите лабораторной работы (структуры данных на основе односвязных списков)	2
			Изучение литературы и других источников (основная и дополнительная литература)	2
			Изучение литературы и других источников (конспекты лекций и ресурсы компьютерных сетей)	2
2	13	Структурированный тип данных файл	Подготовка к выполнению лабораторной работы	2
			Подготовка к защите лабораторной работы (вывод в файл)	2
			Подготовка к защите лабораторной работы (ввод из файла)	2
			Изучение литературы и других источников	2
2	14	Обзор графических возможностей систем программирования (СП) Qbasic и Turbo-Pascal	Подготовка к выполнению лабораторной работы (разработка схем алгоритмов)	2
			Подготовка к выполнению лабораторной работы (разработка программы для СП Qbasic)	3
			Подготовка к выполнению лабораторной работы (разработка программы для СП Turbo-Pascal)	3
			Подготовка к защите лабораторной работы	2
2		Экзамен	Изучение литературы и других источников по теме «Информатика как наука и как вид практической деятельности»	3
			Изучение литературы и других источников по теме «Алгоритмическая система и ее составные части»	2
			Изучение литературы и других источников по теме «Основные понятия и простейшие средства алгоритмических языков программирования»	3
			Изучение литературы и других источников по теме «Разработка алгоритмов и программ в соответствии с принципом структурности»	3
			Изучение литературы и других источников по теме «Структурированный тип данных массив»	3
			Изучение литературы и других источников по теме «Символьный и строковый типы данных»	3
			Изучение литературы и других источников по теме «Разработка алгоритмов и программ в соответствии с	3

			принципом модульности»	
			Изучение литературы и других источников по теме «Структурированный тип данных запись»	3
			Изучение литературы и других источников по теме «Тип данных указатель»	3
			Изучение литературы и других источников по теме «Структурированный тип данных файл»	3
			Разбор практических заданий по теме «Принцип модульности»	3
			<b>Сдача экзамена</b>	<b>4</b>
		<b>ИТОГО 2 семестр</b>		<b>144</b>
3	15	Парадигмы программирования	Изучение литературы и других источников	1
3	16	Общая характеристика языков программирования семейства Си	Подготовка к выполнению лабораторной работы	1
			Подготовка к защите лабораторной работы	2
3	17	Разработка разветвляющихся алгоритмов и программ в СП Turbo-C++	Изучение литературы и других источников (основная и дополнительная литература)	2
			Изучение литературы и других источников (конспекты лекций и ресурсы компьютерных сетей)	2
			Подготовка к выполнению лабораторной работы (разработка схем алгоритмов)	3
			Подготовка к выполнению лабораторной работы (разработка програм)	3
			Подготовка к защите лабораторной работы	2
3	18	Разработка циклических алгоритмов и программ в СП Turbo-C++	Изучение литературы и других источников	2
			Подготовка к выполнению лабораторной работы (разработка схем алгоритмов)	2
			Подготовка к выполнению лабораторной работы (разработка програм)	2
			Подготовка к защите лабораторной работы (циклы с предусловием и постусловием)	2
			Подготовка к защите лабораторной работы (цикл for)	2
3	19	Ввод с клавиатуры и вывод на монитор средствами C++	Изучение литературы и других источников	3
			Подготовка к выполнению лабораторной работы	3
			Подготовка к защите лабораторной работы	3
3	20	Массивы в языке Си	Изучение литературы и других источников	2
			Подготовка к выполнению лабораторной работы (разработка схемы алгоритма)	3
			Подготовка к выполнению лабораторной работы (разработка програм)	3
			Подготовка к защите лабораторной работы	2
3	21	Строки в языке Си	Изучение литературы и других источников	2
			Подготовка к выполнению лабораторной работы (разработка схем алгоритмов)	3
			Подготовка к выполнению лабораторной работы (разработка програм)	3
			Подготовка к защите лабораторной работы	2
3	22	Реализация принципа модульности средствами языка Си	Изучение литературы и других источников (основная и дополнительная литература)	2
			Изучение литературы и других источников (конспекты лекций и ресурсы компьютерных сетей)	3
			Подготовка к выполнению лабораторной работы (разработка схем алгоритмов)	3
			Подготовка к выполнению лабораторной работы (разработка программных модулей)	3
			Подготовка к выполнению лабораторной работы (разработка программы)	3
			Подготовка к защите лабораторной работы (способы передачи параметров)	2

			Подготовка к защите лабораторной работы (передача параметров массивов и функций)	3		
3	23	Тип данных структура	Изучение литературы и других источников	2		
			Подготовка к выполнению лабораторной работы	3		
3	24	Динамические переменные в языках семейства Си	Подготовка к защите лабораторной работы	2		
			Изучение литературы и других источников	2		
			Подготовка к выполнению лабораторной работы (разработка схемы алгоритма)	2		
			Подготовка к выполнению лабораторной работы (разработка програм)	2		
3	25	Тип файл в СП Turbo-C++	Подготовка к защите лабораторной работы	3		
3		Экзамен	Изучение литературы и других источников по теме «Парадигмы программирования. Общая характеристика языков программирования семейства Си»	2		
			Изучение литературы и других источников по теме «Разработка разветвляющихся программ на языке Си»	3		
			Изучение литературы и других источников по теме «Разработка циклических программ на языке Си»	3		
			Изучение литературы и других источников по теме «Ввод с клавиатуры и вывод на монитор средствами языка Си++»	3		
			Изучение литературы и других источников по теме «Массивы в языке Си»	3		
			Изучение литературы и других источников по теме «Строки в языке Си»	3		
			Изучение литературы и других источников по теме «Реализация принципа модульности средствами языка Си»	3		
			Изучение литературы и других источников по теме «Тип данных структура»	3		
			Изучение литературы и других источников по теме «Динамические переменные в языках семейства Си»	3		
			Изучение литературы и других источников по теме «Тип файл в языке Си»	3		
			Разбор практических заданий по теме «Структурированные типы данных»	3		
			<b>Сдача экзамена</b>			<b>4</b>
			<b>ИТОГО 3 семестр</b>			<b>126</b>
<b>ИТОГО</b>			<b>360</b>			

### 3.2. График работы студента

#### Семестр № 1

Форма оценочного средства	Усл. обозн.	НЕДЕЛЯ																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Лабораторная работа	ЛР	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+

#### Семестр № 2

Форма оценочного средства	Усл. обозн.	НЕДЕЛЯ																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Лабораторная работа	ЛР	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+

#### Семестр № 3

Форма оценочного средства	Усл. обозн.	НЕДЕЛЯ																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Лабораторная работа	ЛР	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+

### 3.3. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

3.3.1. Контрольные работы/рефераты *не предусмотрены*

### 4. Оценочные средства для контроля успеваемости и результатов освоения учебной дисциплины (см. Фонд оценочных средств)

4.2. Рейтинговая система оценки знаний обучающихся по учебной дисциплине  
*Рейтинговая система не используется.*

### 5. Учебно-методическое и информационное обеспечение дисциплины

#### 5.1. Основная литература

№ п/п	Автор (ы), наименование, место издания и издательство, год	Используется при изучении разделов	Семестр	Количество экземпляров	
				В библиотеке	На кафедре
1	2	3	4	5	6
1.	Информатика и программирование. Основы информатики [Текст]: учебник для студ. учреждений высш. проф. образования/ [Н.И.Парфилова, А.В.Пруцков, А.Н.Пылькин, Б.Г.Трусов]; под ред. Б.Г.Трусова.-М.: Издательский центр «Академия», 2012.-336 с.	1, 2, 15	1, 3	14	
2.	Информатика и программирование. Алгоритмизация и программирование [Текст]: учебник для студ. учреждений высш. проф. образования/ [Н.И.Парфилова, А.В.Пруцков, А.Н.Пылькин, Б.Г.Трусов]; под ред. Б.Г.Трусова.-М.: Издательский центр «Академия», 2012.-336 с.	1-14	1, 2	14	
3.	Новичков В.С., Пылькин А.Н. Начала программирование на языке QBasic [Текст]: Учебное пособие. М.: Горячая линия-Телеком, 2007	1-14	1, 2	14	
4.	Каширин И.Ю., Новичков В.С. От С к С++ [Текст]: Учебное пособие для вузов.- М.: Горячая линия – Телеком, 2005	16-25	3	10	

## 5.2. Дополнительная литература

№ п/п	Автор (ы), наименование, место издания и издательство, год	Используется при изучении разделов	Семестр	Количество экземпляров	
				В библиотеке	На кафедре
1	2	3	4	5	6
1.	Новичков, В. С. Алгоритмизация и программирование на Турбо Паскале [Текст] : учебное пособие / В. С. Новичков, Н. И. Парфилова, А. Н. Пылькин. – Москва : Горячая линия-Телеком, 2005. – 438 с.	1-14	1, 2	7	
2.	Новожилов, О. П. Информатика [Электронный ресурс] : учебник для прикладного бакалавриата / О. П. Новожилов. – 3-е изд., перераб. и доп. – Москва : Юрайт, 2017. – 619 с. Режим доступа: <a href="https://www.biblio-online.ru/book/FEE705BC-11CB-46EB-810E-2634A4DE5E46">https://www.biblio-online.ru/book/FEE705BC-11CB-46EB-810E-2634A4DE5E46</a> (дата обращения: 31.08.2020).	1, 2, 15	1, 3	ЭБС	
3.	Савич, У. Программирование на С++ [Текст] / У. Савич. – 4-е изд. – Санкт-Петербург : Питер, 2004. – 781 с.	16-25	3	5	

## 5.3. Базы данных, информационно-справочные и поисковые системы

1. VOOR.ru [Электронный ресурс] : электронная библиотека. – Доступ к полным текстам по паролю. – Режим доступа: <http://www.book.ru> (дата обращения: 31.08.2020).

2. East View [Электронный ресурс] : [база данных]. – Доступ к полным текстам статей научных журналов из сети РГУ имени С.А. Есенина. – Режим доступа: <http://dlib.eastview.com> (дата обращения: 31.08.2020).

3. Moodle [Электронный ресурс] : среда дистанционного обучения / Ряз. гос. ун-т. – Рязань, [Б.г.]. – Доступ, после регистрации из сети РГУ имени С.А. Есенина, из любой точки, имеющей доступ к Интернету. – Режим доступа: <http://e-learn2.rsu.edu.ru/moodle2> (дата обращения: 31.08.2020).

4. Znanium.com [Электронный ресурс] : [база данных]. – Доступ к полным текстам по паролю. – Режим доступа: <http://znanium.com> (дата обращения: 31.08.2020).

5. «Издательство «Лань» [Электронный ресурс] : электронная библиотека. – Доступ к полным текстам по паролю. – Режим доступа: <http://e-lanbook.com> (дата обращения: 31.08.2020).

6. Университетская библиотека ONLINE [Электронный ресурс] : электронная библиотека. – Доступ к полным текстам по паролю. – Режим доступа: <http://www.biblioclub.ru> (дата обращения: 31.08.2020).

7. Юрайт [Электронный ресурс] : электронная библиотека. – Доступ к полным текстам по паролю. – Режим доступа: <http://www.biblio-online.ru> (дата обращения: 31.08.2020).

8. Труды преподавателей [Электронный ресурс] : коллекция // Электронная библиотека Научной библиотеки РГУ имени С.А. Есенина. – Доступ к полным текстам по паролю. – Режим доступа: <http://dspace.rsu.edu.ru/xmlui/handle/123456789/3> (дата обращения: 31.08.2020).

## 5.4. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. eLIBRARY.RU [Электронный ресурс] : научная электронная библиотека. – Режим доступа: <http://elibrary.ru/defaultx.asp>, свободный (дата обращения: 31.08.2020).

2. Единая коллекция цифровых образовательных ресурсов [Электронный ресурс] : федеральный портал. – Режим доступа: <http://school-collection.edu.ru/>, свободный (дата обращения: 31.08.2020).

3. Единое окно доступа к образовательным ресурсам [Электронный ресурс] : федеральный портал. – Режим доступа: <http://window.edu.ru/>, свободный (дата обращения: 31.08.2020).

4. Интернет Университет Информационных технологий. [Электронный ресурс] : сайт. – Режим доступа: <http://www.intuit.ru/>, свободный (дата обращения 31.08.2020).
5. Петров Д.Н. Парадигмы программирования. [Электронный ресурс] : сайт. – Режим доступа: <http://dnpetrov.narod.ru/>, свободный (дата обращения 31.08.2020).
6. Портал естественных наук. [Электронный ресурс] : сайт. – Режим доступа: <http://e-science11.ru>, свободный (дата обращения 31.08.2020).
7. Портал для программистов и администраторов информационных систем. [Электронный ресурс] : сайт. – Режим доступа: <http://www.coderpost.net/>, свободный (дата обращения 31.08.2020).
8. Российский общеобразовательный портал [Электронный ресурс] : образовательный портал. – Режим доступа: <http://www.school.edu.ru/>, свободный (дата обращения: 31.08.2020).
9. Сервер Информационных Технологий [Электронный ресурс] : сайт. – Режим доступа: <http://citforum.ru/>, свободный (дата обращения 31.08.2020).
10. Сайт программирования. [Электронный ресурс] : сайт. – Режим доступа: <http://www.cyberguru.ru/>, свободный (дата обращения 10.06.2018).
11. Сайт программирования в среде Delphi. [Электронный ресурс]: сайт. – Режим доступа: <http://www.delphisources.ru/>, свободный (дата обращения 10.06.2018).
12. Федеральный центр информационно-образовательных ресурсов [Электронный ресурс] : Единое окно доступа к образовательным ресурсам. – Режим доступа: <http://fcior.edu.ru>, свободный (дата обращения: 31.08.2020).

## **6. Материально-техническое обеспечение дисциплины**

### **6.1. Требования к аудиториям для проведения занятий:**

Специализированные лекционные аудитории, оснащенные видеопроекционным оборудованием, подключенным к компьютеру.

Компьютерные классы для проведения лабораторных занятий и организации самостоятельной работы студентов, имеющие рабочие места, оснащенные компьютером с доступом к серверам кафедры ИВТ и МПИ, сети Интернет и видеопроекционному оборудованию.

### **6.2. Требования к оборудованию рабочих мест преподавателя и обучающихся:**

Персональный компьютер под управлением MS Windows, Microsoft Office или аналогичное.

### **6.3. Требования к специализированному оборудованию: отсутствует**

## **7. Методические указания для обучающихся по освоению дисциплины**

Вид учебных занятий	Организация деятельности студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначить вопросы, термины, материал, который вызывает трудности, пометить и попытаться найти ответ в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на консультации, на практическом занятии. Уделить внимание следующим понятиям: <i>этапы решения задачи на компьютере; основные понятия алгоритмической системы; принципы пошаговой детализации структурности и модульности; алгоритмические структуры; вложение; структурный анализ; алфавит, синтаксис и семантика языка программирования; программа для компьютера; идентификатор; переменная; данные и их типы, операция, стандартная функция, выражение; простой и структурированный оператор.</i>



Лабораторная работа	<p>В соответствии с запланированным на самостоятельную работу временем (раздел 3.1) изучить соответствующий теоретический материал и практические рекомендации.</p> <p>В соответствии с запланированным на самостоятельную работу временем составить схемы алгоритмов и программы решения соответствующего варианта учебной задачи.</p> <p>Согласовать заранее составленные схемы и программы с преподавателем, ведущим занятие. Тексты программ должны содержать короткие комментарии, отражающие тему и номер лабораторной работы, номер варианта, фамилию студента, связь тех или иных переменных с условием задачи, а также комментарии, отражающие этапы решения задачи. Ввод с клавиатуры и вывод на монитор числовых данных должны сопровождаться краткими текстовыми сообщениями.</p> <p>Ввести тексты программ в компьютер, осуществить отладку и тестирование программ, при наличии нескольких программ решения одной и той же задачи добиться с учетом возможных погрешностей одинаковых результатов и продемонстрировать работоспособность программ преподавателю.</p> <p>Оформить отчет о лабораторной работе с указанием фамилии студента, номера лабораторной работы и номера варианта. Оформленная работа также должна содержать полный текст задания, схему алгоритма (кроме л/р № 1), тексты отлаженных на компьютере программ с комментариями по всем переменным. После текста программы на конкретном языке программирования должны быть отражены результаты ее выполнения из окна результатов в том виде, в котором они были выведены системой программирования. В случае ветвящихся алгоритмов и программ необходимо отразить несколько вариантов выполнения программы с различными наборами исходных данных, соответствующих различным ветвям исполнения алгоритма.</p> <p>Защитить оформленную лабораторную работу, продемонстрировав теоретические и практические знания, умения и навыки по соответствующей теме, возможные варианты схем решения задачи, структуры алгоритмов которых отличны от структур оформленных схем. Перечень примерных контрольных заданий и вопросов для защиты лабораторных работ приведен в разделе 11.2.</p>
Подготовка к экзамену (зачету)	При подготовке к экзамену (зачету) необходимо ориентироваться на конспекты лекций, рекомендуемую литературу, типовые практические задания и др.

## **8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине**

Для организации учебной и самостоятельной работы обучаемых используется технология удаленного доступа. Для каждой из учебных групп на сервере кафедры ИВТ и МПИ созданы каталоги с соответствующими правами доступа. В каталоге группы создан подкаталог для данной дисциплины, в котором по мере необходимости преподавателем размещаются рабочая программа дисциплины, электронные варианты лекций, электронные обучающие ресурсы, задания к лабораторным работам, графики выполнения лабораторных работ, материалы для самостоятельной работы, контрольные материалы, оценки текущих результатов учебной деятельности обучающихся и др. материалы для организации учебного процесса по данной дисциплине. Материалы, размещенные

в каталоге группы доступны любому обучающемуся соответствующей группы посредством локальной компьютерной сети университета с любого рабочего места компьютерных классов кафедры ИВТ и МПИ.


В каталоге группы также для каждого обучающегося создан личный подкаталог, к которому разрешен доступ только обучающемуся и преподавателям кафедры. В личном подкаталоге обучающийся размещает результаты своей учебной деятельности: выполненные лабораторные работы, отчеты и другие результаты.

Для организации учебной работы может использоваться набор веб-сервисов MS office365, вебинарная платформа РГУ имени С.А. Есенина, университетская информационно-образовательная среда Moodle, облачные технологии. Координация учебной работы осуществляется через университетскую электронную почту.

## **9. Требования к программному обеспечению учебного процесса**

1. Операционная система Windows Pro (договор №65/2019 от 02.10.2019);
2. Антивирус Kaspersky Endpoint Security (договор №14-ЗК-2020 от 06.07.2020г.);
3. Офисное приложение LibreOffice (свободно распространяемое ПО);
4. Система программирования Qbasic (свободно распространяемое ПО);
5. Система программирования PascalABC (свободно распространяемое ПО);
6. Среда разработки приложений RAD Studio 10.1 Berlin Professional Concurrent ELC (договор № 11\05\2016-9774 11.05.16г.);
7. Архиватор 7-zip (свободно распространяемое ПО);
8. Браузер изображений Fast Stone Image Viewer (свободно распространяемое ПО);
9. PDFридер Foxit Reader (свободно распространяемое ПО);
10. Медиа проигрыватель VLC mediaplayer (свободно распространяемое ПО);
11. Запись дисков ImageBurn (свободно распространяемое ПО);
12. DJVU браузер DjVu Browser Plug-in (свободно распространяемое ПО);
13. Набор веб-сервисов MS office365 (бесплатное ПО для учебных заведений <https://www.microsoft.com/ru-ru/education/products/office>);
14. Система электронного обучения Moodle (свободно распространяемое ПО).

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ С.А. ЕСЕНИНА»

Утверждаю:  
Декан физико-математического  
факультета  
 Н.Б. Федорова  
«31» августа 2020 г.

## **АННОТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ**

### **ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ**

Направление подготовки  
**02.03.03 Математическое обеспечение и администрирование  
информационных систем**

Направленность (профиль) подготовки  
**Администрирование информационных систем**

Квалификация  
**Бакалавриат**

Форма обучения  
**Очная**

Рязань, 2020

### **1. Цель освоения дисциплины**

Целью освоения учебной дисциплины «Информатика и программирование» является формирование у обучающихся общекультурных и общепрофессиональных компетенций в процессе изучения информатики и программирования для последующего применения в учебной и практической деятельности.

### **2. Место дисциплины в структуре ОПОП**

Дисциплина Б1.Б.9.«Информатика и программирование» относится к базовой части Блока 1.

Дисциплина изучается на 1 и 2 курсе (1 - 3 семестры)

**3. Трудоемкость дисциплины:** 18 зачетных единиц, 648 академических часов.

**4. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы.**

№ п/п	Номер/индекс компетенции	Содержание компетенции (или ее части)	Перечень планируемых результатов обучения по дисциплине В результате изучения дисциплины обучающиеся должны:		
			Знать:	Уметь:	Владеть (навыками):
1	2	3	4	5	6
1	ОПК-3	готовность анализировать проблемы и направления развития технологий программирования	<ul style="list-style-type: none"> <li>• предмет и задачи исследования информатики как науки;</li> <li>• сущность и значение информации в развитии современного информационного общества;</li> <li>• этапы решения задачи на компьютере на основе парадигмы императивного программирования;</li> <li>• понятие алгоритма и его основные свойства;</li> <li>• основные понятия алгоритмической системы: представление информации в виде данных, система команд исполнителя, алгоритмический язык исполнителя;</li> <li>• основные средства записи и типы алгоритмов;</li> <li>• цели и составные части технологии нисходящего структурного проектирования алгоритмов и программ;</li> <li>• алгоритмические структуры, их основные свойства и приемы использования;</li> <li>• понятие алгоритмического модуля и его свойства;</li> <li>• принципы выделения и правила записи алгоритмических модулей;</li> <li>• семантику вызова алгоритмического модуля, в том числе, рекурсивного вызова, и механизмы реализации межмодульного интерфейса;</li> <li>• основные парадигмы программирования;</li> <li>• номенклатуру языков программирования различного уровня и их возможности.</li> </ul>	<ul style="list-style-type: none"> <li>• при решении задач практической информатики на основе парадигмы императивного программирования конкретизировать и описать основные понятия, результаты других научных дисциплин, данные и их типы средствами записи алгоритмов;</li> <li>• разрабатывать и записывать алгоритмы и программы в соответствии с технологией нисходящего структурного проектирования;</li> <li>• записывать алгоритмы с помощью схем;</li> <li>• анализировать структуру алгоритмов;</li> <li>• описывать алфавит и основные синтаксические конструкции языков с помощью нормальных форм Бэкуса-Наура.</li> </ul>	<ul style="list-style-type: none"> <li>• основными методами, способами и средствами обработки информации на основе парадигмы императивного программирования в соответствии с технологией нисходящего структурного проектирования.</li> </ul>

1	2	3	4	5	6
2	ОПК-8	<p>способность использовать знания методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание программного обеспечения</p>	<ul style="list-style-type: none"> <li>• составные части алгоритмического языка программирования (алфавит, синтаксис, семантика) и способы их описания;</li> <li>• концепцию типов данных;</li> <li>• структуры программ на языках программирования Бейсик, Паскаль и Си;</li> <li>• основные понятия алгоритмических языков программирования и их классификацию: идентификаторы, данные и их типы, операции, стандартные функции, выражения, операторы;</li> <li>• синтаксис и семантику основных операторов языков программирования Бейсик, Паскаль и Си (присваивания, ввода-вывода, ветвления, циклов), способы реализации алгоритмических структур средствами языков программирования;</li> <li>• основные синтаксические конструкции, используемые в языках программирования Бейсик, Паскаль и Си для описания и обработки данных типа массив;</li> <li>• прием флаг;</li> <li>• синтаксис и семантику реализации алгоритмического модуля средствами языков Бейсик, Паскаль и Си;</li> <li>• способы передачи параметров при реализации межмодульного интерфейса;</li> <li>• синтаксис типов запись с фиксированными полями, указатель и типизированный файл в языке Паскаль;</li> <li>• синтаксис типов структура, указатель и файл в языке Си;</li> <li>• понятия статического и динамического объекта программы на языках Паскаль и Си;</li> <li>• синтаксис и семантику основных средств управления использованием динамической памятью в языках Паскаль и</li> </ul>	<ul style="list-style-type: none"> <li>• разрабатывать и записывать алгоритмы и программы в соответствие с принципом структурности;</li> <li>• при решении учебной задачи конкретизировать и описать основные понятия, данные и их типы, организовать ввод и вывод, реализовать алгоритмические структуры средствами, по крайней мере, трех императивных языков программирования;</li> <li>• разрабатывать алгоритмы и программы обработки массивов;</li> <li>• разрабатывать алгоритмы и программы обработки строк;</li> <li>• решать учебные задачи в соответствие с принципом модульности;</li> <li>• разрабатывать и использовать библиотеку пользователя в СП Turbo-Pascal и Turbo-C++;</li> <li>• разрабатывать простейшие алгоритмы и программы моделирования движения гра-</li> </ul>	<ul style="list-style-type: none"> <li>• навыками проектирования, ввода, отладки и тестирования программ в средах, по крайней мере, трех императивных систем программирования.</li> </ul>

			<p>Си;</p> <ul style="list-style-type: none"> <li>• структуру данных файл и основные действия с файлами последовательного доступа;</li> <li>• синтаксис и семантику основных средств обработки файлов в СП Turbo-Pascal и Turbo-C++ и последовательность их использования для организации ввода и вывода.</li> </ul>	<p>фического изображения;</p> <ul style="list-style-type: none"> <li>• использовать тип указатель для организации односвязного списка, очереди и стека;</li> <li>• решать учебные задачи по обработке данных типа запись, структура, файл и линейных динамических структур данных средствами систем программирования (СП) Turbo-Pascal и Turbo-C++.</li> </ul>	
--	--	--	--	--	--

### **5. Форма промежуточной аттестации и семестр прохождения**

Зачет (1 семестр).

Экзамен (2, 3 семестры).

Дисциплина реализуется частично с применением дистанционных образовательных технологий.