

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«СЕВЕРО-КАВКАЗСКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ»

«УТВЕРЖДАЮ»

Проректор по учебной работе

« 30 » 03

Г.Ю. Нагорная



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Процедурное программирование

Уровень образовательной программы бакалавриат

Направление подготовки 01.03.02 Прикладная математика и информатика

Направленность (профиль) Прикладная математика и информатика

Форма обучения очная

Срок освоения ОП 4 года

Институт Цифровых технологий

Кафедра разработчик РПД Математика

Выпускающая кафедра Математика

Начальник  
учебно-методического управления

Семенова Л.У.

Директор института ЦТ

Тебурев Д.Б.

Заведующий выпускающей кафедрой

Кочкаров А.М.

г. Черкесск, 2023 г.

## СОДЕРЖАНИЕ

<b>1. Цели освоения дисциплины</b> .....	4
<b>2. Место дисциплины в структуре образовательной программы</b> .....	5
<b>3. Планируемые результаты обучения по дисциплине</b> .....	5
<b>4. Структура и содержание дисциплины</b> .....	6
4.1. Объем дисциплины и виды учебной работы.....	6
4.2. Содержание дисциплины .....	7
4.2.1. Разделы (темы) дисциплины, виды учебной деятельности и формы контроля.....	7
4.2.2. Лекционный курс .....	8
4.2.3. Лабораторный практикум .....	10
4.2.4. Практические занятия .....	12
4.3. Самостоятельная работа обучающегося.....	14
<b>5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине</b> .....	15
<b>6. Образовательные технологии</b> .....	19
<b>7. Учебно-методическое и информационное обеспечение дисциплины</b> .....	20
7.1. Перечень основной и дополнительной учебной литературы.....	20
7.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет».....	22
7.3. Информационные технологии, лицензионное программное обеспечение .....	22
<b>8. Материально-техническое обеспечение дисциплины</b> .....	22
8.1. Требования к аудиториям (помещениям, местам) для проведения занятий .....	22
8.2. Требования к оборудованию рабочих мест преподавателя и обучающихся .....	25
8.3. Требования к специализированному оборудованию.....	25
<b>9. Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья</b> .....	25
<b>Приложение 1. Фонд оценочных средств</b> .....	26
<b>Приложение 2. Аннотация рабочей программы</b> .....	65
<b>Рецензия на рабочую программу</b> .....	66
<b>Лист переутверждения рабочей программы дисциплины</b> .....	67

**Целью** освоения дисциплины **«Процедурное программирование»** является ознакомление обучающихся с основными инструментариями разработки программных продуктов с использованием языков программирования высокого уровня.

**Задачи дисциплины:**

- освоение обучающимися основных принципов и методов системы программирования;
- технологии разработки алгоритмов и программ, методов отладки и решения задач на ЭВМ в различных режимах;
- освоение основ объектно-ориентированного подхода к программированию;
- основ процедурного подхода к программированию;
- изучение средств программирования предоставляющие сервисные возможности для разработки их собственных компьютерных программ;
- изучение программно-аппаратных методов и средств системы программирования;
- изучение машинно-ориентированных языков программирования.

**2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ**

2.1. Дисциплина **«Процедурное программирование»** относится к вариативной части Блока 1 Дисциплины (модули), имеет тесную связь с другими дисциплинами.

2.2. Ниже приведены предшествующие и последующие дисциплины, направленные на формирование компетенций дисциплины в соответствии с матрицей компетенций ОП.

**Предшествующие и последующие дисциплины, направленные на формирование компетенций**

№ п/п	Предшествующие дисциплины	Последующие дисциплины
1	Знания, полученные на предыдущем уровне образования	Алгоритмы на графах Языки и методы программирования Проект 2 курса Объектно – ориентированное программирование Цифровые технологии от интернета до блокчейна Управление личными финансами Мобильные сети и технологии

### 3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Планируемые результаты освоения образовательной программы (ОП) – компетенции обучающихся определяются требованиями стандарта по направлению подготовки 01.03.02 Прикладная математика и информатика и формируются в соответствии с матрицей компетенций ОП

<b>№ п/п</b>	<b>Номер/индекс компетенции</b>	<b>Наименование компетенции (или ее части)</b>	<b>В результате изучения дисциплины обучающиеся должны:</b>
1	2	3	4
1.	ПК-3	Способен осуществлять программную реализацию алгоритмов при разработке прикладных информационных систем	ПК-3.1 Обладает знаниями в области программирования и современных компьютерных технологий при разработке прикладных информационных систем. Способен осуществлять оптимизацию алгоритмов при создании прикладных программ или прикладных комплексов ПК-3.2 Анализирует стандартные технические требования к проектированию программного обеспечения. ПК-3.3 Способен осуществлять разработку технических требований к программному обеспечению, при его создании

#### 4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

##### 4.1. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

Вид учебной работы	Всего часов	Семестры		
		№ 1	№2	
		часов	часов	
1	2	3	4	
<b>Аудиторная контактная работа (всего)</b>	<b>108</b>	<b>72</b>	<b>72</b>	
В том числе:				
Лекции (Л)	36	18	18	
Практические занятия (ПЗ), В том числе, практическая подготовка				
Лабораторные работы (ЛР)	72	36	36	
Контактная внеаудиторная работа, в том числе: индивидуальные и групповые консультации	1,5	1,5		
<b>Самостоятельная работа обучающегося (СРО) (всего)</b>	<b>115</b>	<b>52</b>	<b>63</b>	
<i>Контрольная работа (Кон)</i>	20	10	10	
<i>Подготовка к лабораторным занятиям</i>	20	10	10	
<i>Подготовка к занятиям (ПЗ)</i>	30	10	20	
<i>Подготовка к текущему контролю (ПТК)</i>	20	10	10	
<i>Самоподготовка</i>	25	12	13	
<b>Промежуточ ная аттестация</b>	Зачет с оценкой, в том числе	ЗаО	ЗаО	
	Прием зачета с оценкой, час	0,5	0,5	
	экзамен (Э) <b>в том числе:</b>	Э	-	Э
	Прием экз., час.	0,5	-	0,5
	Консультация, час.	2	-	2
	СРО, час.	24,5	-	24,5
<b>ИТОГО: Общая трудоемкост ь</b>	<b>часов</b>	<b>252</b>	<b>108</b>	<b>144</b>
	<b>зач. ед.</b>	<b>7</b>	<b>3</b>	<b>4</b>

## 4.2. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### 4.2.1. Разделы (темы) дисциплины, виды учебной деятельности и формы контроля

№ п/п	Наименование раздела (темы) дисциплины	Виды учебной деятельности, включая самостоятельную работу обучающихся (в часах)					Формы текущей и промежуточной аттестации
		Л	ЛР	ПЗ (ПП)	СР О	все го	
1	2	3	4	5	6	7	8
<b>Семестр 1</b>							
1.	Раздел 1. Основы алгоритмизации	4			20	24	контрольные вопросы, контрольная работа, индивидуальные задания к практическим работам, тестирование
4.	Раздел 2. Язык Pascal	14	36		32	82	контрольные вопросы, контрольная работа, индивидуальные задания к практическим работам, тестирование
5.	<b>Контактная внеаудиторная работа</b>					<b>1,5</b>	<b>индивидуальные и групповые консультации</b>
7.	<b>Промежуточная аттестация</b>					<b>0,5</b>	<b>Зачет с оценкой</b>
<b>Итого в 1 семестре</b>		<b>18</b>	<b>36</b>		<b>52</b>	<b>108</b>	
<b>Семестр 2</b>							

8.	Раздел 3. Язык Python и его особенности	4	12		20	38	контрольные вопросы, контрольная работа, индивидуальные задания к практическим работам, тестирование
9.	Раздел 4. Основные алгоритмы и их реализация на Python	8	12		23	41	контрольные вопросы, контрольная работа, индивидуальные задания к практическим работам, тестирование
10.	Раздел 5. Графика в Python и задачи моделирования	6	12		20	38	контрольные вопросы, контрольная работа, индивидуальные задания к практическим работам, тестирование
11.	<b>Контактная внеаудиторная работа</b>					<b>2</b>	<b>индивидуальные и групповые консультации</b>
13.	<b>Промежуточная аттестация</b>					<b>25</b>	<b>Экзамен</b>
<b>Итого во 2 семестре</b>		<b>18</b>	<b>36</b>		<b>63</b>	<b>144</b>	
<b>ВСЕГО:</b>		<b>36</b>	<b>72</b>	<b>36</b>	<b>115</b>	<b>252</b>	

#### 4.2.2. Лекционный курс

№ п/п	Наименование раздела дисциплины	Наименование темы лекции	Содержание лекции	Всего часов
1	2	3	4	5
<b>Семестр 1</b>				
1	Раздел 1. Основы	Тема 1.1 Краткая история	История развития ВТ	<b>2</b>

	алгоритмизации	развития ВТ и ПО.	и ПО. Архитектура ЭВМ.	
2	Раздел 1. Основы алгоритмизации	Тема 1.2 Этапы решения задач на ЭВМ	В т.ч. Алгоритмы. Свойства. Способы описания. Оценка эффективности.	2
3	Раздел 2. Язык Pascal	Тема 2.1 Структура программы. Основные операторы и типы данных.	В т.ч. операторы присваивания, ветвления, циклы.	2
4	Раздел 2. Язык Pascal	Тема 2.2 Типы данных, определяемые пользователем	Интервальные, перечисляемые типы	2
5	Раздел 2. Язык Pascal	Тема 2.3 Подпрограммы.	Процедуры, функции, рекурсия, модули	2
6	Раздел 2. Язык Pascal	Тема 2.4 Графика	Системы координат. Область рисования. Построение фигур из отрезков.	2
7	Раздел 2. Язык Pascal	Тема 2.5 Графика	Стандартные функции построения фигур. Работа с цветом.	2
8	Раздел 2. Язык Pascal	Тема 2.6 Файлы. Текстовые файлы	Текстовые файлы. Ввод/вывод строк. Режимы доступа к файлу.	2
9	Раздел 2. Язык Pascal	Тема 2.7 Структурированные типы данных	Множество, запись, сравнение с массив, строка	2
<b>ИТОГО часов в 1 семестре:</b>				<b>18</b>
<b>Семестр 2</b>				
1	Раздел 3. Язык Python и его особенности	Тема 3.1 Общие сведения о языке Python	Типы данных, числа. Логические выражения. Структуры данных. Неизменяемые последовательности – строки. Чтение из файл и запись в файл.	2
2	Раздел 3. Язык Python и его особенности	Тема 3.2 Средства программирования на Python. Структура т программы	Обзор существующих средств программирования на Python . Технические характеристики	2
3	Раздел 4. Основные алгоритмы и их реализация на Python	Тема 4.1 Линейный алгоритм. Операции с числами и строками. Ветвление и алгоритм выбора	Операции с числами и строками. Линейная структура программы. Программа ветвления и выбора.	2
4	Раздел 4. Основные алгоритмы и их реализация на Python	Тема 4.2 Циклические алгоритмы	Циклические алгоритмы. Обработка последовательностей и одномерных массивов. Сортировка массива.	2



5	Раздел 4. Основные алгоритмы и их реализация на Python	Тема 4.3 Обработка двумерных массивов.	Обработка двумерных массивов (матриц).	2
6	Раздел 4. Основные алгоритмы и их реализация на Python	Тема 4.4 Работа с ассоциативными массивами	Таблицы данных. Работа с ассоциативными массивами	2
7, 8	Раздел 5. Графика в Python и задачи моделирования	Тема 5.1 Управление исполнителем «Черепашка»	Графический исполнитель «Черепашка», работа с ним.	2
9	Раздел 5. Графика в Python и задачи моделирования	Тема 5.2 Пользовательские подпрограммы и моделирование.	Пользовательские программы и подпрограммы. Моделирование	2
10	Раздел 5. Графика в Python и задачи моделирования	Тема 5.3 Модуль Tkinter	Работа с модулем Tkinter	2
<b>ИТОГО часов во 2 семестре:</b>				<b>18</b>
<b>ВСЕГО часов</b>				<b>36</b>

#### 4.2.3. Лабораторный практикум

№ п/п	Наименование раздела дисциплины	Наименование лабораторной работы	Содержание лабораторной работы	Всего часов
1	2	3	4	5
<b>Семестр 1</b>				
1.	Раздел 2. Язык Pascal	Лабораторная работа №1 Компиляция и исполнение программ	Создание простейших консольных и оконных приложений с использованием FreePascal и Lazarus.	2
2.	Раздел 2. Язык Pascal	Лабораторная работа № 2-3. Программы линейной структуры	Создание приложений линейной структуры. Базовый ввод/вывод.	4
3.	Раздел 2. Язык Pascal	Лабораторная работа №4-5. Условный и составной операторы. Оператор Case	Создание приложений, требующих ветвления.	4
4.	Раздел 2. Язык Pascal	Лабораторная работа №6-7 Циклы	Создание приложений, использующих циклы, признак EOF.	4
5.	Раздел 2. Язык Pascal	Лабораторная работа №8-9. Массивы	Создание приложений, использующих массивы.	4
6.	Раздел 2. Язык	Лабораторная работа №10.	Создание приложений,	2

	Pascal	Строки	обрабатывающих текстовую информацию.	
7.	Раздел 2. Язык Pascal	Лабораторная работа №1 1-12 Подпрограммы	Создание приложений, использующих процедуры и функции.	<b>4</b>
8.	Раздел 2. Язык Pascal	Лабораторная работа №13. Пользовательские типы данных	Создание приложений, использующих пользовательские типы данных.	<b>2</b>
9.	Раздел 2. Язык Pascal	Лабораторная работа №14-16 Графика	Создание приложений, формирующих графическую информацию.	<b>6</b>
10.	Раздел 2. Язык Pascal	Лабораторная работа № 17-18. Текстовые файлы	Создание приложений, обрабатывающих текстовые файлы.	<b>4</b>
<b>ИТОГО часов в 1 семестре:</b>				<b>36</b>
<b>Семестр 2</b>				
11.	Раздел 3. Язык Python и его особенности	Лабораторная работа №1 Средства программирования на Python	Изучение технических возможностей средств программирования на Python.	<b>2</b>
13.	Раздел 4. Основные алгоритмы и их реализация на Python	Лабораторная работа №2 Линейный алгоритм. Операции с числами и строками. Ветвление и алгоритм выбора	Операции с числами и строками. Линейная структура программы.	<b>2</b>
	Раздел 4. Основные алгоритмы и их реализация на Python	Лабораторная работа №3 Л Ветвление и алгоритм выбора	Разработка программ ветвления и выбора.	<b>4</b>
14.	Раздел 4. Основные алгоритмы и их реализация на Python	Лабораторная работа №3 Циклические алгоритмы	Циклические алгоритмы. Обработка последовательностей и одномерных массивов. Сортировка массива.	<b>4</b>
15.	Раздел 4. Основные алгоритмы и их реализация на Python	Лабораторная работа №4 Обработка двумерных массивов.	Обработка двумерных массивов (матриц).	<b>4</b>
16.	Раздел 4. Основные алгоритмы и их реализация на Python	Лабораторная работа №5 Работа с ассоциативными массивами	Таблицы данных. Работа с ассоциативными массивами	<b>6</b>
17.	Раздел 5. Графика в Python и задачи моделирования	Лабораторная работа №6 Управление исполнителем «Черепашка»	Графический исполнитель «Черепашка», работа с ним.	<b>4</b>
18.	Раздел 5. Графика в Python и задачи	Лабораторная работа №7 Пользовательские программы и	Пользовательские программы и	<b>6</b>

	моделирования	подпрограммы и моделирование.	подпрограммы. Моделирование	
19.	Раздел 5. Графика в Python и задачи моделирования	Лабораторная работа №8 Модуль Tkinter	Работа с модулем Tkinter	<b>4</b>
<b>ИТОГО часов во 2 семестре:</b>				<b>36</b>
<b>ВСЕГО:</b>				<b>72</b>

### 4.3. САМОСТОЯТЕЛЬНАЯ РАБОТА ОБУЧАЮЩЕГОСЯ

№ п/п	Наименование раздела дисциплины	№ п/п	Виды СРО	Всего часов
1	2	3	4	5
<b>Семестр 1</b>				
1	Раздел 1. Основы алгоритмизации	1.1.	Подготовка к практическим занятиям, в т.ч. чтение конспекта лекций, формулирование вопросов.	5
		1.2	Самоподготовка, в т.ч. чтение доп. Литературы и электронных ресурсов.	10
		1.3	Подготовка к текущему контролю	5
2	Раздел 2. Язык Pascal	2.1.	Выполнение контрольной работы.	10
		2.2.	Подготовка к практическим и лабораторным занятиям, в т.ч. чтение конспекта лекций, формулирование вопросов.	10
		2.3	Самоподготовка, в т.ч. чтение доп. Литературы и электронных ресурсов, посещение форумов по программированию	6
		2.4	Подготовка к текущему контролю	6
<b>ИТОГО часов в 1 семестре:</b>				<b>52</b>
<b>Семестр 2</b>				
4	Раздел 3. Язык Python и его особенности	3.1	Подготовка к практическим и лабораторным занятиям, в т.ч. чтение конспекта лекций, формулирование вопросов.	10
		3.2	Самоподготовка, в т.ч. чтение доп. Литературы и электронных ресурсов, посещение форумов по программированию	10
5	Раздел 4. Основные алгоритмы и их реализация на Python	4.1	Подготовка к практическим и лабораторным занятиям, в т.ч. чтение конспекта лекций, формулирование вопросов.	5
		4.2	Выполнение контрольной работы.	5
		4.3	Самоподготовка, в т.ч. чтение доп. Литературы и электронных ресурсов, посещение форумов по программированию	5
		4.4	Подготовка к текущему контролю	8
6	Раздел 5. Графика в Python и задачи моделирования	5.1	Подготовка к практическим и лабораторным занятиям, в т.ч. чтение конспекта лекций, формулирование вопросов. .	10
		5.2	Самоподготовка, в т.ч. чтение доп. Литературы и электронных ресурсов, посещение форумов по программированию и базам данных	5

	5.3	Подготовка к текущему контролю	5
<b>ИТОГО часов во 2 семестре:</b>			<b>63</b>
<b>ВСЕГО часов</b>			<b>115</b>

## **5. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ**

### **5.1. Методические указания для подготовки обучающихся к лекционным занятиям**

Основными формами обучения процедурному программированию являются лекции, практические и лабораторные занятия и консультации, а также самостоятельная работа.

Лекции составляют основу теоретического обучения и дают систематизированные основы научных знаний по дисциплине, раскрывают состояние и перспективы развития средств программирования, концентрируют внимание обучающихся на наиболее важных вопросах, стимулируют их активную познавательную деятельность и способствуют формированию творческого мышления. Ведущим методом в лекции выступает устное изложение учебного материала, которое, как правило, сопровождается демонстрацией визуального материала на доске или в виде презентации с использованием мультимедиа аппаратуры.

Обучающийся во время лекции ведет конспект. Какими бы замечательными качествами ни обладал лектор, какое бы большое внимание ни уделял лекции слушатель, глубокое понимание материала достигается только путем самостоятельной работы над ним. Работа над конспектом не должна заканчиваться с прослушивания лекции. После лекции, в процессе самостоятельной работы, перед тем, как открыть тетрадь с конспектом, полезно мысленно восстановить в памяти содержание лекции, вспомнив ее структуру, основные положения и выводы. Рекомендуется:

- повторить изученный материал по конспекту;
- непонятные положения отметить на полях и уточнить;
- неоконченные фразы, пропущенные слова и другие недочеты в записях устранить, пользуясь материалами из учебника и других источников;
- завершить техническое оформление конспекта (подчеркивания, выделение главного, выделение разделов, подразделов и т.п.).

Самостоятельную работу над конспектом желательно проводить в тот же день, пока время не стерло содержание лекции из памяти (через 10 ч после лекции в памяти остается не более 30-40 % материала). С целью доработки необходимо прочитать записи, восстановить текст в памяти, а также исправить описки, расшифровать не принятые ранее сокращения, заполнить пропущенные места, понять текст, вникнуть в его смысл. Далее прочитать материал по рекомендуемой литературе, разрешая в ходе чтения возникшие вопросы, а также дополнения и исправляя свои записи. Для наглядности можно использовать выделение заголовков и особо важных мест маркером или подчеркиванием. Следует отдельно записать вопросы, требующие уточнения у преподавателя.

Подготовленный конспект используется и при подготовке к практическим и

лабораторным занятиям.

Для уточнения деталей и более подробного изучения материала следует дополнительно ознакомиться с материалом по теме лекций по рекомендованной литературе.

В конце лекции преподаватель оставляет несколько минут для ответов на вопросы. Чтобы не тратить время, постарайтесь четко сформулировать вопрос, прежде чем задавать его преподавателю.

## **5.2. Методические указания для подготовки обучающихся к лабораторным занятиям**

Ведущей дидактической целью лабораторных занятий является систематизация и обобщение знаний по изучаемой теме, приобретение практических навыков по тому или другому разделу курса, закрепление практически полученных теоретических знаний.

Обязательным условием успешного выполнения любой лабораторной работы является умение откомпилировать программу и выполнить ее, обеспечив вывод полученных результатов. Чтобы быстро выполнять эти действия, обучающийся при подготовке к лабораторным работам должен как минимум еженедельно проводить их самостоятельно - дома или в аудитории для самостоятельной работы. Нужно создавать самостоятельно как приложения с графическим интерфейсом (GUI), так и консольные. В каждой подготовке нужно использовать как процедуры ввода ( `Read / Readln` ) и вывода ( `Write / Writeln` ) для консольных приложений, так и преобразования типов (например `StrToInt`, `IntToStr`, `FloatToStr` и т.п. ). Требуется научиться понимать сообщения компилятора. Для этого нужно ознакомиться хотя бы с минимумом соответствующей английской лексики. В случае затруднений, запишите текст сообщения, переведите его и попробуйте понять при подготовке к следующим занятиям. В конце работы, как правило, проводится обсуждение с участием всей подгруппы различных вариантов реализации заданий. Следует использовать это обсуждение для подготовки к следующим лабораторным работам, записав новые для себя способы создания и выполнения программ.

## **5.4 Методические указания к выполнению контрольной работы**

Контрольной работе как одной из форм самостоятельной учебно-исследовательской работы отводится особая роль при формировании компетенции будущего специалиста и бакалавра. Здесь обучающийся демонстрирует применение полученных знаний для создания приложений, решающих конкретные поставленные перед ним задачи. Обучающийся предъявляет преподавателю несколько версий программ, как правило, в электронном виде и получает от преподавателя положительное заключение о результате, либо замечания и предложения по корректировке программы. Программа должна предъявляться в виде, допускающем быстрый переход к ее компиляции, т.е. не допускается передача в виде изображения. Принимаются любые варианты программы, решающие исходную задачу. Преподаватель при приеме приводящей к верному результату программы зачитывает ее как исполненную, но может дать рекомендации по ее улучшению.

## **5.5 Методические указания по самостоятельной работе обучающихся**

Самостоятельная работа обучающегося предполагает различные формы индивидуальной учебной деятельности: конспектирование научной литературы, сбор и анализ практического материала в СМИ, проектирование, выполнение тематических и творческих заданий и пр. Выбор форм и видов самостоятельной работы определяется индивидуально-личностным подходом к обучению совместно преподавателем и обучающимся. Формы текущего контроля успеваемости обучающихся- устный опрос, проверка программ, выполненных в рамках контрольных и лабораторных работ.

Содержание внеаудиторной самостоятельной работы обучающихся по дисциплине «Процедурное программирование» включает в себя различные виды деятельности:

- чтение текста (учебника, первоисточника, дополнительной литературы);
- конспектирование текста;
- изучение конспекта лекций и формулирование вопросов к преподавателю;
- работа со словарями и справочниками;
- ознакомление с доступным кодом программ разных разработчиков;
- исследовательская работа;
- работа с электронными информационными ресурсами, в т.ч. использование аудио- и видеозаписей ;
- чтение материалов форумов по программированию, в т.ч. на иностранных языках;
- ответы на контрольные вопросы;
- решение задач и упражнений.

## **5.6 Методические рекомендации прохождения тестирования**

Подготовку к итоговому тестированию необходимо осуществлять поэтапно.

На первом этапе необходимо повторить основные положения всех тем, детально разбирая наиболее сложные моменты. Непонятные вопросы необходимо выписывать, чтобы по ним можно было проконсультироваться с преподавателем перед прохождением итогового тестирования. Подготовку по темам каждой дидактической единицы целесообразно производить отдельно. На этом этапе необходимо использовать материалы лекционного курса, материалы семинарских занятий, тестовые задания для текущего контроля знаний, а также презентации лекционного курса.

На втором этапе подготовки предлагается без повторения теоретического материала дать ответы тестовые задания для рубежного контроля знаний. Если ответы на какие-то вопросы вызвали затруднение, необходимо еще раз повторить соответствующий теоретический материал.

Наконец, третий этап подготовки необходимо осуществить непосредственно накануне теста. На данном этапе необходимо аккуратно просмотреть весь лекционный курс.

В случае, если результаты выполнения тестового задания оказались неудовлетворительными, необходимо зафиксировать темы, на вопросы по которым были даны неверные ответы, и еще раз углубленно повторить соответствующие темы в соответствии с указанными выше тремя этапами подготовки к тестированию

## 6. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

№ п/п	Виды учебной работы	Образовательные технологии	Всего часов
1	2	3	4
<b>Семестр 1</b>			
1	<i>Лекция 1.1</i> Краткая история развития ВТ и ПО	Обзорная лекция. Визуализация	<b>2</b>
2	<i>Лекция 1.2</i> Этапы решения задач на ЭВМ	Лекция–беседа. Визуализация	<b>2</b>
3	<i>Лекция 1.3</i> Структура программы. Основные операторы и типы данных	Лекция – информация. Презентация	<b>2</b>
4	<i>Лабораторная работа №1, 10,14-16</i>	Самостоятельная работа обучающихся по индивидуальным заданиям лабораторного практикума с ПЭВМ. Коллективное обсуждение реализованных вариантов.	<b>8</b>
5	<i>Практическое занятие № 3,8</i>	Коллективный разбор конкретных ситуаций.	<b>4</b>
<b>Итого часов в 1 семестре:</b>			<b>18</b>
<b>Семестр 2</b>			
6	<i>Лекция 2.5</i> Сортировка. Улучшенные методы сортировки. Быстрая сортировка	Лекция – беседа. Визуализация	<b>2</b>
7	<i>Лекция 2.7-2.8</i> Использование стандартных компонент Lazarus	Лекция – беседа. Визуализация	<b>4</b>
8	<i>Лекция 2.9</i> Работа с БД через стандартные компоненты	Лекция – информация. Визуализация	<b>2</b>
9	<i>Лабораторная работа № 22,24,26</i>	Самостоятельная работа обучающихся по индивидуальным заданиям лабораторного практикума с ПЭВМ. Коллективное обсуждение реализованных вариантов.	<b>6</b>
10	<i>Практическое занятие №10-12,18</i>	Коллективный разбор конкретных ситуаций.	<b>6</b>
11	<i>Практическое занятие №17</i> Сравнение алгоритмов сортировки	Учебно-исследовательская работа.	<b>2</b>
<b>Итого часов во 2 семестре:</b>			<b>22</b>
<b>Всего:</b>			<b>40</b>



## 7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### 7.1. Перечень основной и дополнительной учебной литературы

#### Основная литература

1. Гумерова Л.З. Программирование в Delphi 7 : учебное пособие / Гумерова Л.З., Аглямзянова Г.Н.. — Красноярск : Научно-инновационный центр, 2019. — 246 с. — ISBN 978-5-6042232-5-3. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/97107.html> — Режим доступа: для авторизир. пользователей
2. Основы программирования на языке Паскаль. Основные понятия алгоритмического языка Паскаль : учебное пособие для самостоятельной работы по дисциплине «Информатика» студентов 2-го курса всех направлений подготовки / . — Воронеж : Воронежский государственный технический университет, ЭБС АСВ, 2017. — 53 с. — ISBN 978-5-7731-0504-6. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/72929.html> — Режим доступа: для авторизир. пользователей
3. Савельева Н.В. Основы программирования на PHP : учебное пособие / Савельева Н.В.. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 260 с. — ISBN 978-5-4497-0681-2. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/97567.html> — Режим доступа: для авторизир. пользователей
4. Информатика. Теория, вычисления, программирование : учебное пособие для практических и лабораторных работ для студентов вузов / Т.П. Крюкова [и др.].. — Кемерово : Кемеровский технологический институт пищевой промышленности, 2014. — 226 с. — ISBN 978-5-89289-836-2. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/61264.html> — Режим доступа: для авторизир. пользователей
5. Куляс О.Л. Программирование на языке ASSEMBLER. Часть 2 : лабораторный практикум по дисциплине «ЭВМ и периферийные устройства» / Куляс О.Л., Никитин К.А.. — Самара : Поволжский государственный университет телекоммуникаций и информатики, 2016. — 79 с. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/71870.html> — Режим доступа: для авторизир. пользователей
6. Куляс О.Л. Программирование на языке ASSEMBLER. Часть 1 : лабораторный практикум по дисциплине «ЭВМ и периферийные устройства» / Куляс О.Л., Никитин К.А.. — Самара : Поволжский государственный университет телекоммуникаций и информатики, 2016. — 89 с. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/71869.html> — Режим доступа: для авторизир. пользователей
7. Хвощев С.В. Основы программирования в Delphi для ОС Android : учебное пособие / Хвощев С.В.. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 85 с. — ISBN 978-5-4497-0891-5. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/102032.html> — Режим доступа: для авторизир. пользователей
10. Смоленцев Н.К. MATLAB: программирование на Visual C#, Borland C#, JBuilder, VBA : учебный курс / Смоленцев Н.К.. — Саратов : Профобразование, 2019. — 456 с. — ISBN 978-5-4488-0066-5. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/89868.html> — Режим доступа: для авторизир. пользователей

#### Дополнительная литература

1. Сузи Р.А. Язык программирования Python : учебное пособие / Сузи Р.А.. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 350 с. — ISBN 978-5-4497-0705-5. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/97589.html> . — Режим доступа: для авторизир. пользователей
2. Туральчук К.А. Параллельное программирование с помощью языка C# / Туральчук К.А.. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр

Медиа, 2019. — 189 с. — ISBN 978-5-4486-0506-2. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/79714.html> — Режим доступа: для авторизир. пользователей

3. Левин М.П. Параллельное программирование с использованием OpenMP : учебное пособие / Левин М.П.. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 133 с. — ISBN 978-5-4497-0685-0. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/97572.html> — Режим доступа: для авторизир. пользователей
4. Борисенко В.В. Основы программирования : учебное пособие / Борисенко В.В.. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 322 с. — ISBN 978-5-4497-0678-2. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/97568.html> — Режим доступа: для авторизир. пользователей

## 7.2 Интернет-ресурсы, справочные системы

1. Электронно-библиотечная система IPRbooks URL: [http:// www.iprbooks.ru/](http://www.iprbooks.ru/) ООО «Ай Пи Эр Медиа»

## 7.3. Информационные технологии, лицензионное программное обеспечение

Лицензионное программное обеспечение	Реквизиты лицензий/ договоров
Microsoft Azure Dev Tools for Teaching 1. Windows 7, 8, 8.1, 10 2. Visual Studio 2008, 2010, 2013, 2019 5. Visio 2007, 2010, 2013 6. Project 2008, 2010, 2013 7. Access 2007, 2010, 2013 и т. д.	Идентификатор подписчика: 1203743421 Срок действия: 30.06.2022  (продление подписки)
MS Office 2003, 2007, 2010, 2013	Сведения об Open Office: 63143487, 63321452, 64026734, 6416302, 64344172, 64394739, 64468661, 64489816, 64537893, 64563149, 64990070, 65615073 Лицензия бессрочная
Антивирус Dr.Web Desktop Security Suite	Лицензионный сертификат Серийный № 8DVG-V96F-H8S7-NRBC Срок действия: с 20.10.2022 до 22.10.2023
Цифровой образовательный ресурс IPRsmart	Лицензионный договор № 10423/23П от 30.06.2023 г. Срок действия: с 01.07.2023 г. до 01.07.2024г.

Бесплатное ПО: Lazarus, Firebird, IBE Expert, Pascal ABC, Python, VBA, Virtual box, Sumatra PDF, 7-Zip

## 8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### 8.1. Требования к аудиториям (помещениям, местам) для проведения занятий

Учебная аудитория для проведения занятий лекционного типа

Специализированная мебель:

Доска меловая - 1 шт., стол компьютерный угловой преподавательский - 1 шт., стул мягкий - 1 шт., кафедра напольная - 1 шт., парты – 12 шт., компьютерные столы -8 шт., стулья – 28 шт.

**Лабораторное оборудование, технические средства обучения, служащие для предоставления учебной информации большой аудитории:**

Экран настенный рулонный – 1 шт.

Проектор – 1 шт.

Компьютер в сборе - 10шт.

**Учебная аудитория для проведения занятий лекционного типа**

Специализированная мебель:

Кафедра напольная - 1 шт., стул преподавательский мягкий - 1 шт., парты - 18шт., стулья мягкие -32шт., стулья ученические-11 шт., доска меловая - 1 шт.

Набор демонстрационного оборудования и учебно-наглядных пособий, обеспечивающих тематические иллюстрации:

Интерактивная система – 1 шт.

Системный блок -1 шт.

Проектор – 1шт.

**Учебная аудитория для проведения занятий семинарского типа, курсового проектирования (выполнение курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации**

Специализированная мебель:

Доска меловая - 1 шт., стол компьютерный угловой преподавательский - 1 шт., стул мягкий - 1 шт., кафедра напольная - 1 шт., парты – 12 шт., компьютерные столы -8 шт., стулья – 28 шт.

**Лабораторное оборудование, технические средства обучения, служащие для предоставления учебной информации большой аудитории:**

Экран настенный рулонный – 1 шт.

Проектор – 1 шт.

Компьютер в сборе - 10шт.

**Учебная аудитория для проведения занятий семинарского типа, курсового проектирования (выполнение курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации**

Специализированная мебель:

Доска меловая - 1 шт., парты - 10шт., стулья - 30шт., стул мягкий-1шт., стол одностумбовый преподавательский – 1шт., компьютерные столы - 10шт.

Технические средства обучения, служащие для предоставления учебной информации большой аудитории:

ПК-8 шт.

Экран настенный рулонный – 1 шт.

Проектор – 1 шт.

#### **Лаборатория современных вычислительных систем. Лаборатория новых компьютерных технологий**

Специализированная мебель:

Доска меловая - 1шт., стол компьютерный угловой преподавательский - 1шт., стул мягкий - 1шт., кафедра напольная - 1шт., парты – 12 шт., компьютерные столы -8 шт., стулья – 28 шт.

**Лабораторное оборудование, технические средства обучения, служащие для предоставления учебной информации большой аудитории:**

Экран настенный рулонный – 1 шт.

Проектор – 1 шт.

Компьютер в сборе - 10шт.

#### **Лаборатория прикладного статистического анализа**

Специализированная мебель:

Парты - 8шт., стулья - 22шт., стол преподавательский -1шт., доска меловая - 1шт., компьютерные столы - 8 шт.

**Технические средства обучения, служащие для предоставления учебной информации большой аудитории:**

Компьютер в сборе -7 шт.

Настенный экран – 1 шт.

Проектор -1 шт.

Ноутбук – 1шт.

#### **Лаборатория метрологии, стандартизации и подтверждения соответствия.**

Специализированная мебель:

Доска меловая - 1шт., парты - 10шт., стулья - 30шт., стул мягкий-1шт., стол одностумбовый преподавательский – 1шт., компьютерные столы - 10шт.

**Технические средства обучения, служащие для предоставления учебной информации большой аудитории:**

ПК-8 шт.

Экран настенный рулонный – 1 шт.

Проектор – 1 шт.

## **8.2. Требования к оборудованию рабочих мест преподавателя и обучающихся**

1. Рабочее место преподавателя, оснащенное компьютером с доступом в Интернет.
2. Рабочие места обучающихся, оснащенные компьютерами с доступом в Интернет, предназначенные для работы в цифровой образовательной среде.

## **8.3. Требования к специализированному оборудованию**

- нет

## **9. ОСОБЕННОСТИ РЕАЛИЗАЦИИ ДИСЦИПЛИНЫ ДЛЯ ИНВАЛИДОВ И ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ**

Для обеспечения образования инвалидов и обучающихся с ограниченными возможностями здоровья разрабатывается (в случае необходимости) адаптированная образовательная программа, индивидуальный учебный план с учетом особенностей их психофизического развития и состояния здоровья, в частности применяется индивидуальный подход к освоению дисциплины, индивидуальные задания: рефераты, письменные работы и, наоборот, только устные ответы и диалоги, индивидуальные консультации, использование диктофона и других записывающих средств для воспроизведения лекционного и семинарского материала.

В целях обеспечения обучающихся инвалидов и лиц с ограниченными возможностями здоровья комплектуется фонд основной учебной литературой, адаптированной к ограничению электронных образовательных ресурсов, доступ к которым организован в БИЦ Академии. В библиотеке проводятся индивидуальные консультации для данной категории пользователей, оказывается помощь в регистрации и использовании сетевых и локальных электронных образовательных ресурсов, предоставляются места в читальном зале.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**  
**ПО ДИСЦИПЛИНЕ «Процедурное программирование»**

# 1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

## Процедурное программирование

### Компетенции, формируемые в процессе изучения дисциплины

Индекс	Формулировка компетенции
ПК-3	Способен осуществлять программную реализацию алгоритмов при разработке прикладных информационных систем

## 2. Этапы формирования компетенции в процессе освоения дисциплины

Основными этапами формирования указанных компетенций при изучении обучающимися дисциплины являются последовательное изучение содержательно связанных между собой разделов (тем) учебных занятий. Изучение каждого раздела (темы) предполагает овладение обучающимися необходимыми компетенциями. Результат аттестации обучающихся на различных этапах формирования компетенций показывает уровень освоения компетенций обучающимися.

Этапность формирования компетенций прямо связана с местом дисциплины в образовательной программе.

Разделы (темы) дисциплины	Темы	Формируемые компетенции (коды)
		ПК -3
Раздел 1. Основы алгоритмизации	1.1 Краткая история развития ВТ и ПО	+
	1.2 Этапы решения задач на ЭВМ	+
Раздел 2. Язык Pасca	2.1 Структура программы. Основные операторы и типы данных	+
	2.2 Типы данных, определяемые пользователем	+
	2.3 Подпрограммы	+
	2.4 Графика	+
	2.6 Файлы. Текстовые файлы	+
	2.7 Структурированные типы данных	+
Раздел 3. Язык Python и его особенности	Тема 3.1 Общие сведения о языке Python	+
	Тема 3.2 Средства программирования на Python. Структура т программы	+
Раздел 4.	Тема 4.2 Циклические	+

Основные алгоритмы и их реализация на Python	алгоритмы	
	Тема 4.3 Обработка двумерных массивов.	+
	Тема 4.4 Работа с ассоциативными массивами	+
Раздел 5. Графика в Python и задачи моделирования	Тема 5.1 Управление исполнителем «Черепашка»	+
	Тема 5.2 Пользовательские подпрограммы и моделирование.	+
	Тема 5.3 Модуль Tkinter	+



### 3. Показатели, критерии и средства оценивания компетенций, формируемых в процессе изучения дисциплины

#### ПК-3 Способен осуществлять программную реализацию алгоритмов при разработке прикладных информационных систем

Индикаторы достижения компетенций	Критерии оценивания результатов обучения				Средства оценивания результатов обучения	
	неудовлетворительно	удовлетворительно	хорошо	отлично	Текущий контроль	Промежуточная аттестация
ПК-3.1 Обладает знаниями в области программирования и современных компьютерных технологий при разработке прикладных информационных систем. Способен осуществлять оптимизацию алгоритмов при создании прикладных программ или прикладных комплексов	Не обладает знаниями в области программирования и современных компьютерных технологий при разработке прикладных информационных систем. Способен осуществлять оптимизацию алгоритмов при создании прикладных программ или прикладных комплексов	Обладает частичными знаниями в области программирования и современных компьютерных технологий при разработке прикладных информационных систем. Способен осуществлять оптимизацию алгоритмов при создании прикладных программ или прикладных комплексов	Знает методы и способы программирования и современных компьютерных технологий при разработке прикладных информационных систем. Способен осуществлять оптимизацию алгоритмов при создании прикладных программ или прикладных комплексов	Знает на отлично методы и способы программирования и современных компьютерных технологий при разработке прикладных информационных систем. Способен осуществлять оптимизацию алгоритмов при создании прикладных программ или прикладных комплексов	контрольные вопросы, контрольная работа, тестирование, задания для лабораторной работы	Зачет с оценкой Экзамен
ПК-3.2 Анализирует стандартные технические требования к проектированию программного обеспечения.	Не умеет анализировать стандартные технические требования к проектированию программного обеспечения.	Умеет частично анализировать стандартные технические требования к проектированию программного обеспечения.	Умеет с неточностями анализировать стандартные технические требования к проектированию программного обеспечения	Может на отлично анализировать стандартные технические требования к проектированию программного обеспечения	контрольные вопросы, контрольная работа, тестирование, задания для лабораторной работы	Зачет с оценкой Экзамен
ПК-3.3 Способен осуществлять разработку технических требований к программному обеспечению, при его создании	Не может осуществлять разработку технических требований к программному обеспечению, при его создании	Способен частично осуществлять разработку технических требований к программному обеспечению, при его создании	Способен осуществлять разработку технических требований к программному обеспечению, при его создании	Может на отлично осуществлять разработку технических требований к программному обеспечению, при его создании	контрольные вопросы, контрольная работа, тестирование, задания для лабораторной работы	Зачет с оценкой Экзамен

## 4. Комплект контрольно-оценочных средств по дисциплине

### Вопросы к зачету с оценкой

#### по дисциплине Процедурное программирование

##### **Вопрос №1:**

- 1- Фон-неймановская архитектура.
- 2- Этапы решения задач на ЭВМ.
- 3- Свойства алгоритмов.
- 4- Алфавит языка Паскаль. Комментарий.
- 5- Общая структура программы на языке Паскаль.
- 6- Данные целого типа и операции над ними.
- 7- Данные вещественного типа и операции над ними.
- 8- Данные булевого типа и операции над ними.
- 9- Данные строкового типа и операции над ними.
- 10- Определение ограниченного и перечисляемого типов.
- 11- Массивы. Описание, обращение к элементу массива.
- 12- Логические операции над целыми операндами.
- 13- Определение типов.
- 14- Оператор присваивания.
- 15- Порядок выполнения операций в выражениях.
- 16- Условный и составной операторы.
- 17- Оператор цикла while.
- 18- Оператор цикла until.
- 19- Оператор цикла for.
- 20- Процедуры ввода данных.
- 21- Оператор case.
- 22- Данные символьного типа и операции над ними.
- 23- различия операций / и Div
- 24- Процедуры вывода данных.
- 25- Использование функции EOF.
- 26- Тип данных Record. Описание, использование.
- 27- Множества. Описание и использование.
- 28- Процедуры break и continue.
- 29- Чтение и запись текстовых файлов.
- 30- Виды файлов в языке Pascal.
- 31- Чтение и запись типизированных файлов.
- 32- Чтение и запись нетипизированных файлов.
- 33- Оператор With.
- 34- Метки. оператор GOTO.
- 35- Последовательный поиск. Двоичный поиск.
- 36- Простые методы внутренней сортировки.
- 37- Графика. Переход к экранной системе координат.
- 38- Сортировка методом выбора.
- Подпрограммы. Формальные и фактические параметры.
- 39- Сортировка методом вставки.
- 40- Сортировка методом пузырька.

##### **Вопрос №2:**

- 1- Подпрограммы. Отличительная особенность функций.
- 2- Подпрограммы. Параметры, передаваемые по ссылке и по значению.

- 3- Подпрограммы. Формальные и фактические параметры.
- 4- Подпрограммы. Вызов.
- 5- Подпрограммы. Описание.
- 6- Подпрограммы. Рекурсия.
- 7- Подпрограммы. Глобальные и локальные переменные.

## Вопросы к экзамену

### по дисциплине Процедурное программирование

1. Python - интерпретируемый язык или компилируемый
2. . Какие есть меняющиеся и постоянные типы данных
3. Что такое область видимости переменных
4. Что такое introspection
5. Разница между is и ==
6. Разница между \_\_init\_\_ () и \_\_new\_\_ ()
7. В чем разница между потоками и процессами
8. Какие есть виды импорта
9. Что такое класс, итератор, генератор?
10. Что такое метакласс, переменная цикла?
11. В чем разница между итераторами и генераторами?
12. В чем разница между staticmethod и classmethod?
13. Как работают декораторы, контекстные менеджеры?
14. Как работают dict comprehension, list comprehension и set comprehension?
15. Можно ли использовать несколько декораторов для одной функции?
16. Можно ли создать декоратор из класса?
17. Какие есть основные популярные пакеты (requests, pytest, etc)?
18. Что такое lambda-функции?
19. Что означает \*args, \*\*kwargs и как они используются?
20. Что такое exceptions, <try-except>?
21. Что такое PEP (Python Enhancement Proposal), какие из них знаете (PEP 8, PEP 484)?
22. Напишите hello-world сервис, используя один из фреймворков.
23. Какие есть типы данных и какая разница между list и tuple, зачем они?
24. Как использовать встроенные коллекции (list, set, dictionary)?
25. В чем заключается сложность доступа к элементам dict?
26. Как создается объект в Python, для чего \_\_new\_\_, зачем \_\_init\_\_ ?

#### **Варианты практических заданий:**

- 01- Найти минимальное из 3-х введенных чисел
- 02- Найти минимальное из 7 введенных чисел
- 03- Найти минимальное из введенных чисел  
Признак окончания ввода — неположительное число
- 04- Найти максимальное из 3-х введенных чисел
- 05- Найти максимальное из 8 введенных чисел
- 06- Найти минимальное из введенных чисел.  
Признак окончания ввода — положительное число
- 07- Подсчитать количество введенных чисел.  
Задать критерий конца ввода на свое усмотрение.
- 08- Подсчитать количество чисел больших 5.  
Задать критерий конца ввода на свое усмотрение.
- 09- Заполнить массив размерностью 5x5 случайными целыми числами. Выдать на экран сумму элементов

- главной диагонали.
- 10- Заполнить массив размерностью 5x5 случайными целыми числами. Выдать на экран сумму всех элементов
  - 11- Заполнить массив размерностью 5x5 случайными целыми числами. Выдать на экран содержимое массива
  - 12- Подсчитать количество строк текстового файла c:\a\abc.txt с использованием пользовательской функции.
  - 13- Подсчитать количество строк текстового файла c:\a\abc.txt с длиной больше 5 с использованием пользовательской функции.
  - 14- Подсчитать количество строк текстового файла c:\a\abc.txt с длиной меньше 10 с использованием пользовательской функции.
  - 15- Подсчитать количество строк текстового файла c:\a\abc.txt с длиной от 10 до 25 с использованием пользовательской функции.
  - 16- Выдать на экран последнюю строку текстового файла c:\BQ\qq.txt с использованием пользовательской функции.
  - 17- Выдать на экран первую строку текстового файла c:\BQ\qq.txt с использованием пользовательской функции.
  - 18- Выдать на экран предпоследнюю строку текстового файла c:\BQ\qq.txt с использованием пользовательской функции.
  - 19- Выдать на экран три первых строки текстового файла c:\BQ\qq.txt
  - 20- Записать в файл c:\out.txt первую строку файла c:\bb.txt
  - 21- Записать в файл c:\out.txt три первых строки файла [c:\bb.txt](#)
  - 22- Записать в файл c:\out.txt три последних строки файла [c:\bb.txt](#)
  - 23- Записать в файл c:\out.txt последнюю строку файла c:\bb.txt
  - 24- Записать в файл c:\out.txt предпоследнюю строку файла c:\bb.txt
  - 25- Дописать к файлу c:\out.txt первую строку файла c:\bb.txt
  - 26- Дописать к файлу c:\out.txt три первых строки файла [c:\bb.txt](#)
  - 27- Дописать к файлу c:\out.txt три последних строки файла [c:\bb.txt](#)
  - 28- Дописать к файлу c:\out.txt последнюю строку файла c:\bb.txt
  - 29- Дописать к файлу c:\out.txt предпоследнюю строку файла c:\bb.txt
  - 30- Записать в файл [c:\out.txt](#) длины строк файла c:\qq.txt
  - 31- Дописать к файлу c:\out.txt длины строк файла [c:\qq.txt](#)
  - 32- Записать в файл [c:\out.txt](#) 120 упорядоченных по возрастанию вещественных чисел
  - 33- Дописать к файлу [c:\out.txt](#) 120 упорядоченных по возрастанию вещественных чисел
  - 34- Записать в файл [c:\out.txt](#) 10 упорядоченных по убыванию вещественных чисел
  - 35- Дописать к файлу [c:\out.txt](#) 10 упорядоченных по убыванию вещественных чисел
  - 36- Записать в файл [c:\out.txt](#) 120 упорядоченных по возрастанию целых чисел
  - 37- Дописать к файлу [c:\out.txt](#) 120 упорядоченных по возрастанию целых чисел
  - 38- Записать в файл [c:\out.txt](#) 10 упорядоченных по убыванию целых чисел
  - 39- Дописать к файлу [c:\out.txt](#) 10 упорядоченных по убыванию целых чисел
  - 40- Записать в файл [c:\out.txt](#) 15 упорядоченных по возрастанию

- строк, не являющихся числами
- 41- Дописать к файлу [c:\out.txt](#) 15 упорядоченных по возрастанию строк, не являющихся числами
  - 42- Записать в файл [c:\out.txt](#) 10 упорядоченных по убыванию строк, не являющихся числами
  - 43- Дописать к файлу [c:\out.txt](#) 10 упорядоченных по убыванию строк, не являющихся числами
  44. Описать нерекурсивную функцию вычисления факториала.

# СЕВЕРО-КАВКАЗСКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ

Кафедра Математика

20 - 20 учебный год

Экзаменационный билет № 1

по дисциплине Процедурное программирование

для обучающихся направления подготовки 01.03.02 Прикладная математика и информатика, 1 курс

1. Фон-неймановская архитектура.
2. Подпрограммы. Отличительная особенность функций.
3. Практическое задание.

Зав. кафедрой

Кочкаров АМ.

## Контрольные вопросы

по дисциплине Процедурное программирование

### Вопросы к разделу 1.

1. Понятие алгоритма. Свойства алгоритмов.
2. Формы представления алгоритмов.
3. Эффективность алгоритмов.
4. Понятия  $O(f)$  и  $o(f)$
5. Можно ли считать алгоритмом `while true do x:=2;`
6. Единицы измерения информации.
7. Этапы решения задач на ЭВМ.
8. Можно ли считать программу правильной (неправильной) при успешном(неуспешном) выполнении тестов?
9. Формы представления числовой информации.
10. Позиционные системы счисления.
11. Перевести число из одной системы счисления в другую.(с основаниями 10,2,8,16) например представить в10-й системе 16-чное число  $2F$  и т.д.
12. Булевские операции: конъюнкция, дизъюнкция, инверсия.
13. Таблицы истинности.
14. Выполнение логических операций над целыми операндами, например вычислить  $z:= 5 \text{ or } 3;$
15. Эволюция языков программирования.
16. Разница между компиляторами и интерпретаторами.
17. Исходный, объектный и загрузочный модули.

### Вопросы к разделу 2.

1. Консольные приложения (CLI) и приложения с графическим интерфейсом (GUI).
2. Интегрированная среда программирования.
3. Создание приложений CLI в Lazarus.
4. Создание приложений GUI в Lazarus.
5. Форма и размещение на ней управляющих элементов. Палитра компонент.
6. Инспектор объектов.
7. Компоненты Edit, Label и их основные свойства.
8. Компонент Button и его основные свойства и события.
9. Разновидности операции деления в Pascal.
10. Различия между операторами цикла repeat и while.
11. Выразить while через repeat.
12. Выразить repeat через while.
13. Выразить for через while.
14. Какие операторы цикла являются универсальными, т. е. позволяющими обойтись без других операторов цикла?
15. Как округлить результат вычисления (вещественное число) до 3 значащих десятичных цифр?
16. Какой тип удобен для представления 4-х сторон света (север, восток, юг, запад)?
17. Как можно представить условие — в будний(выходной) день?
18. Разновидности файлов в Pascal.
19. Как дописать строку к текстовому файлу?
20. Как записать информацию в текстовый файл, очистив его предыдущее содержимое.
21. Что может привести к неполной записи информации в файл?
22. Выполнение логических операций над целыми операндами, например вычислить  $z := 5 \text{ or } 3;$  или  $z := 7 \text{ shl } 2;$
23. Попробуйте упростить выражение  $(abc > 3) \text{ and } (bc > abc) \text{ and } (3 \geq abc)$
24. Определить значение выражения  $(A \text{ div } 10) * 10$  при целом  $a=25$
25. Определить значение выражения  $(A / 10) * 10$  при целом  $a=25$
26. Записать в элементы массива  $a[1..100]$  с четными индексами их номера, увеличенные на 1.
27. Понятие компонент.
28. Основные свойства компонент.
29. Как сделать компонент недоступным / доступным?
30. Как сделать компонент невидимым / видимым?
31. Компонент Memo.
32. Основные свойства и методы для Tstrings.

### Вопросы к разделу 3.

1. Сравните максимальное количество сравнений при двоичном и последовательном поиске.
2. Сравните минимальное количество сравнений при двоичном и последовательном поиске.
3. Оцените экономию от упорядоченности массива при поиске, если его размер —  $8 * 10 ** 6$  элементов.
4. Приведите нижнюю оценку сложности алгоритмов сортировки, основанных на сравнениях..
5. В чем схожесть и различие алгоритмов пузырька и quick-sort.
6. Какие минусы имеет использование отсортированных файлов.

### Вопросы к разделу 4.

1. Какие основные типы БД поддерживаются при работе с Lazarus?
2. Назовите и кратко охарактеризуйте методы для работы с таблицами dbf.
3. Как получить значение поля таблицы dbf?

4. Назовите основные компоненты для работы с БД, поддерживающими SQL/
5. Каково основное назначение компонент типа Tconnection.
6. Каково основное назначение компонент типа TSQL.
7. Что такое транзакция?
8. Как стартовать транзакцию?
9. Как подтвердить транзакцию?
10. Как откатить транзакцию?
11. Как выполнить запрос на удаление записей?
12. Как выполнить запрос на чтение записей?
13. Как перейти к следующей записи?
14. Использование компонента DBGrid.

## **Комплект заданий для лабораторной работы**

по дисциплине «Процедурное программирование»

**Лабораторная работа №1.** Компиляция и исполнение программ

**Цель:** Создание простейших консольных и оконных приложений с использованием Freepascal и Lazarus.

**Варианты заданий:** Вывод текстового сообщения или результата вычисления по формуле, например  $2*3$ .

**Лабораторная работа № 2-3.** Программы линейной структуры

**Цель:** Создание приложений линейной структуры. Базовый ввод/вывод.

**Варианты заданий:** Вывод исходных данных и результата решения задачи для задач, решение которых сводится к вычислению по формуле, например определение площади плоских фигур, площади поверхности и объема тел, времени падения тел, преобразованию единиц измерения, сумм договоров в разных валютах и т.д.

**Лабораторная работа №4-5.** Условный и составной операторы. Оператор Case

**Цель:** Создание приложений, требующих ветвления.

**Варианты заданий:** Вывод 2-х введенных чисел в порядке возрастания/ убывания.

Вывод 3-х введенных чисел в порядке возрастания/ убывания. Нахождение величины или порядкового номера максимального / минимального целого / вещественного числа из 2 (3,6) введенных. Решение квадратного уравнения. Определение площади треугольника по сторонам.

**Лабораторная работа №6-7. Циклы**

**Цель:** Создание приложений, использующих циклы, признак EOF.

**Варианты заданий:** Определение количества, среднего значения, величины или порядкового номера максимального / минимального целого / вещественного числа, когда их количество заранее неизвестно, а определяется по признаку окончания, в т.ч. EOF.

Перенаправление ввода/ вывода для чтения из файла, вывода в файл.

**Лабораторная работа №8-9. Массивы**

**Цель:** Создание приложений, использующих массивы.

**Варианты заданий:** Определение среднего значения, дисперсии, максимального отклонения от среднего для последовательности чисел. Решение системы 2 линейных уравнений с 2 неизвестными. Транспонирование матриц, вычисление следа и др. характеристик матриц. Вывод таблиц значений разных функций, например соответствия градусов Цельсия и Фаренгейта, фунтов и килограммов, величины дохода за разные



месяцы по вкладу при разных ставках годового процента и заданной сумме вклада.

#### **Лабораторная работа №10. Строки**

**Цель:** Создание приложений, обрабатывающих текстовую информацию.

**Варианты заданий:** Определение длины, количества слов, количества пробелов в строке / в нескольких введенных строках. Определение количества всех / с длиной больше / меньше заданной/ содержащих / не содержащих заданную подстроку строк. Ввод с клавиатуры / из стандартного ввода, перенаправленного в файл.

#### **Лабораторная работа №11-12. Подпрограммы**

**Цель:** Создание приложений, использующих процедуры и функции.

**Варианты заданий:** Вывод вычисленных значений в виде простых отчетов, например, расчетного листка по зарплате, платежной ведомости. Пользовательские функции для расчета дохода по вкладу, НДФЛ (с учетом необлагаемого минимума), расчета объема / массы тел, приближенных вычислений, например сумм рядов и др. Рекурсивные функции, например вычисление факториала.

#### **Лабораторная работа №13. Пользовательские типы данных**

**Цель:** Создание приложений, использующих пользовательские типы данных.

**Варианты заданий:** Использование в программе понятий день недели, месяц, будни, выходные, время года: север, восток, юг, запад и т.п.

#### **Лабораторная работа №14-16. Графика**

**Цель:** Создание приложений, формирующих графическую информацию.

**Варианты заданий:** Работа в экранной системе координат. Формирование изображения из отрезков прямых. Переход от традиционной декартовой к экранной системе координат. Переход между произвольными системами координат. Переход от полярной к декартовой системе координат. Сдвиг, поворот. Повторение фрагментов. Работа с цветом. Градиенты цвета. Правильные многоугольники. Забор, волна, солнышко. Спираль Архимеда, астроида, фигуры Лиссажу. Динамические изображения.

#### **Лабораторная работа № 17-18. Текстовые файлы**

**Цель:** Создание приложений, обрабатывающих текстовые файлы.

**Варианты заданий:** Определение длины, количества слов, количества пробелов в строке / в нескольких введенных строках. Определение количества всех / с длиной больше / меньше заданной/ содержащих / не содержащих заданную подстроку строк. Строки вводятся из заданного пользователем файла. Перебор имен файлов с определенными именами в заданном каталоге. Определение выше приведенных параметров для нескольких файлов. Запись / добавление отвечающих заданным критериям строк из нескольких файлов в заданный, например с целью поиска в журналах активности пользователей или ошибок оборудования.

#### **Лабораторная работа №19. Множества**

**Цель:** Создание приложений, использующих множества.

**Варианты заданий:** Анализ исходных кодов программ Lazarus, содержащих множества. Создание программ, содержащих условия, сформулированные с использованием in, not in, and, or или операции объединения / разности множеств.

#### **Лабораторная работа №20. Записи**

**Цель:** Создание приложений, использующих записи.

**Варианты заданий:** Записи о сотруднике для отдела кадров, бухгалтерии, Запись об успеваемости обучающегося, Запись о характеристиках рыб, животных, деревьев. Запись о химических элементах, химических соединениях. О строительных блоках. ит.д.

#### **Лабораторная работа №21. Типизированные файлы**

**Цель:** Создание приложений, обрабатывающих типизированные файлы.

**Варианты заданий:** Чтение / запись типизированных файлов с разными типами элементов:

целые и вещественные разных типов, строки, записи, массивы. Обратить особое внимание

на переход к заданному элементу.

#### **Лабораторная работа №22-23. Нетипизированные файлы**

**Цель:** Создание приложений, обрабатывающих нетипизированные файлы.

**Варианты заданий:** Чтение информации из файлов с разными расширениями и вывод в 16-ом / текстовом виде. Обратит внимание на представление последовательных целых чисел. Высказать соображения о порядке следования байт. Изменение одного/ нескольких байт в файле с произвольным расширением. Обратит внимание на опасность такой операции.

#### **Лабораторная работа №24. Последовательный поиск**

**Цель:** Создание приложений, использующих последовательный поиск.

**Варианты заданий:** Последовательный поиск среди целых/ вещественных чисел; строк. Определять количество сравнений. Сравнить количество сравнений при успешном и неуспешном поиске. Как реализуется в Pascal поиск с барьером?

#### **Лабораторная работа №25. Двоичный поиск.**

**Цель:** Создание приложений, использующих двоичный поиск.

**Варианты заданий:** Двоичный поиск среди целых/ вещественных чисел; строк.

Определять количество сравнений. Сравнить количество сравнений при успешном и неуспешном поиске. Сравнить эти показатели с последовательным поиском.

#### **Лабораторная работа № 26. Сортировка методом выбора**

**Цель:** Создание приложений, использующих сортировку методом выбора.

**Варианты заданий:** Сортировать файл со случайными целыми / вещественными числами / строками методом выбора. Отсортированный массив выдать на экран /записать в файл.

#### **Лабораторная работа №27. Сортировка методами вставки и пузырька**

**Цель:** Создание приложений, использующих сортировку методами вставки и пузырька.

**Варианты заданий:** Сортировать файл со случайными целыми / вещественными числами / строками методами вставки / пузырька. Отсортированный массив выдать на экран / записать в файл.

#### **Лабораторная работа №28. Быстрая сортировка**

**Цель:** Создание приложений, использующих сортировку методом quick-sort.

**Варианты заданий:** Сортировать файл со случайными целыми / вещественными числами / строками методом quick-sort. Отсортированный массив выдать на экран /записать в файл.

#### **Лабораторная работа №29. Исследование методов сортировки**

**Цель:** Анализ и сравнение методов сортировки.

**Варианты заданий:** Определять количество сравнений. Заполнить таблицу при разных количествах элементов для всех изучавшихся методов сортировки для файлов упорядоченных по возрастанию / убыванию/ со случайными числами.

Анализ полученных результатов будет проведен на практическом занятии.

#### **Лабораторная работа №30-31. Указатели. Динамические структуры данных**

**Цель:** Процедуры new, dispose. Операции над указателями. Реализация стека.

**Варианты заданий:** Опробовать адресные операции. Реализовать стек или очередь.

#### **Лабораторная работа №32-34. Стандартные компоненты Lazarus**

**Цель:** Создание приложений, использующих стандартные компоненты .

**Варианты заданий:** Обеспечение видимости / невидимости кнопки/ окна и надписи в зависимости от введенного в другое окно значения. То же для доступности / недоступности. Добавление строк / очистка Мемо по заданным кнопкам. Перемещение визуальных компонент в run-time. Убегающая кнопка. Обработка событий OnKeyDown, OnKeyUp, OnKeyPress/ Обеспечение надежного ввода данных.

#### **Лабораторная работа №35-36. Стандартные компоненты доступа к БД**

**Цель:** Создание приложений, использующих стандартные компоненты доступа к БД.

**Варианты заданий:** Прочитать, обработать и вывести на экран одно из полей первой / последней записи таблицы tdbf. Изменить одно из полей первой / последней записи таблицы tdbf. Связать tdbf с DBGrid. Прочитать, обработать и вывести на экран одно из

полей первой / последней записи таблицы БД с помощью SQL. Связать таблицу с DBGrid.

**Результат работы:**

программа в соответствии с заданием к лабораторной работе.

**Форма отчёта: произвольная.**

## **Комплект тестовых вопросов и заданий**

### по дисциплине **Процедурное программирование**

№1 Алгоритм – это

- 1) правила выполнения определенных действий;
- 2) ориентированный граф, указывающий порядок исполнения некоторого набора команд;
- 3) это понятные и точные предписания выполнения действия или решения задачи, которые обязательно приводят к правильному результату;

№2 Суть такого свойства алгоритма, как результативность, заключается в том, что

- 1) алгоритм всегда состоит из последовательности дискретных шагов;
- 2) для записи алгоритма используются команды, которые входят в систему команд исполнителя;
- 3) алгоритм обеспечивает решение не одной конкретной задачи, а некоторого класса задач;

№3 Суть такого свойства алгоритма, как формальность, заключается в том, что

- 1) при выполнении алгоритма исполнитель не должен рассуждать о целесообразности выполнения предписываемых алгоритмом действий, а тем более сомневаться в их истинности. Он должен их выполнять строго в том порядке, который предписывает алгоритм;
- 2) для записи алгоритма используются команды, которые входят в систему команд исполнителя;
- 3) алгоритм обеспечивает решение не одной конкретной задачи, а некоторого класса задач;

№4 Результат выполнения оператора Write('произведение', '=', 5\*5)

- 1) произведение=25;
- 2) произведение=5\*5;
- 3) 25;

№5 Раздел типов определяется служебным словом:

---

№6 Символьный тип данных объявляется служебным словом:

---

№7 Цикл с предусловием определяется служебным словом:

---

№8 Определите, если число повторений цикла известно и задано наибольшее допустимое значение n, то лучше использовать:

- 1) цикл с предусловием

- 2) цикл с постусловием
- 3) цикл со счетчиком

№9 Набор однотипных данных, имеющий общее для всех своих элементов имя

- 1) множество
- 2) массив
- 3) запись

№10 процесс перестановки элементов массива с целью упорядочивания их в соответствии с каким-либо критерием

---

№11 Выясните, в основе какого метода сортировки лежит обмен соседних элементов массива:

- 1) прямой обмен
- 2) прямой выбор
- 3) прямой вариант

№12 Какое утверждение относительно класса `java.lang.Object` верно:

- 1) у этого класса нет полей
- 2) нельзя явно переопределять методы этого класса
- 3) нельзя явно наследовать этот класс

№13 Дан массив `>>> c = array([[1,2], [2,3], [4,5]])`. Чему равен срез `c[:,1]`:

- 1) `array([1, 2, 4])`
- 2) `array([2, 3])`
- 3) `array([2, 3, 5])`

№14 Какие новые имена появятся в текущем модуле после выполнения следующего кода:

```
import sre as re
```

```
from re import compile
```

- 1) имена `sre`, `re` и `compile`
- 2) только имена `re` и `compile`
- 3) только имена `sre` и `compile`

№15 Чему будет равен результат выполнения:

```
urlparse.urlsplit(«http://google.com/search?q=Python#1»):
```

- 1) `('http', 'google.com', '/search', », 'q=Python', '1')`
- 2) `('http://', 'google.com/', 'search?', 'q=Python#', '1')`
- 3) `('http', 'google.com', '/search', 'q=Python', '1')`

№16 Для чего применяется метод `nextset()` объекта-курсора:

- 1) для перехода к следующему набору записей результата запроса
- 2) для перехода к следующей записи результата запроса
- 3) для получения следующего набора записей результата запроса

№17 В каком модуле нужно искать функции, помогающие тестировать программу:

---

№18 Каким образом в модуле `poplib` представлен сеанс работы с POP3-сервером:

- 1) набор функций

- 2) экземпляр класса POP3
- 3) список кортежей

№19 Какой из перечисленных обработчиков `mod_python` выполняется раньше других:

- 1) `PythonPostReadRequestHandler`
- 2) `PythonHandler`
- 3) `PythonFixupHandler`

№20 Что будет получено в результате вычисления следующего выражения:

$(0 < 5 \leq 3) \text{ and } (0 / 0)$ :

- 1) True (или 1)
- 2) False (или 0)
- 3) синтаксическая ошибка

№21 Какое из приведенных ниже регулярных выражений некорректно:

- 1) `a+b++`
- 2) `(?P(ac))`
- 3) `(a+b+)`

№22 Зачем в XML пространства имен:

- 1) пространства имен позволяют указывать опции для приложения, обрабатывающего XML
- 2) пространства имен позволяют включать однотипные XML-документы друг в друга
- 3) для сочетания в одном документе XML с различными DTD +

№23 Регулярное выражение это

- 1) шаблон, описывающий множество строк
- 2) синтаксически правильное выражение на языке Python
- 3) шаблон для поиска файлов в каталоге

№24 Одно из ключевых слов языка Java:

- 1) `false`
- 2) `null`
- 3) `default`

№25 Отметьте верное утверждение относительно языков Java и JavaScript:

- 1) JavaScript является синонимом Java
- 2) их спецификации являются закрытыми
- 3) оба языка кроссплатформенны

№26 Результат работы фрагмента программы: `for i = 1 to 5 do for j:= 1 to 2 do write (a[i,j])`

---

№27 В каком случае в теле цикла не нужно писать операторные скобки:

- 1) если в теле цикла используется 1 оператор
- 2) если в теле цикла используется пустой оператор
- 3) если тело цикла состоит из 2 операторов

№28 Структура языка Pascal, которая повторяет действие, пока сохраняет силу некоторое условие:

---

№29 На каком этапе создания программы формируется задача и выбирается метод решения:

- 1) постановка задачи
- 2) отладка задачи
- 3) тестирование программы

№30 Отладка – это ...

---

№31 Ошибки, которые выявляются при компиляции программы

---

№32 Инструмент сред программирования, который поддерживает совокупность объектных файлов с подпрограммами, типами данных:

- 1) библиотекарь
- 2) словарь
- 3) платформа

№33 Верный вызов вызов процедуры abc(k: integer)

- 1) abc(5);
- 2) z:=abc(5);
- 3) abc(3.5);

№34 Неверное описание массива

- 1) a1: array[1..2,45..70] of integer;
- 2) a2: array(1..2,1..7)of integer;
- 3) a3: array[1..2,1..25] of real;
- 4) a4 : array[2..7] of integer;

## Темы контрольных работ

### по дисциплине Процедурное программирование

Создать приложение в среде Lazarus по следующим темам:

#### **Семестр 1, тема 1.1:**

- 01 найти площадь прямоугольного треугольника с катетами  $a$  и  $b$  и указать, превышает ли она  $S_1$
- 02 найти радиус окружности с площадью  $S$  и сравнить его с  $R_1$
- 03 сравнить объемы куба с ребром  $a$  и цилиндра с диаметром  $a$  и высотой  $h$
- 04 определить какая из трех точек плоскости, заданных координатами  $x, y$  находится дальше от начала координат
- 05 определить, находится ли начало координат внутри треугольника, заданного координатами вершин  $(-x_1, y_1), (x_1, y_1), (0, y_2)$ ,  $x_1 < 0, y_2 < y_1$
- 06 определить, имеет ли действительные корни уравнение  $x^2 + x \sin(R) + Q = 0$ .
- 07 сравнить среднее арифметическое и среднее геометрическое чисел  $A$  и  $B$
- 08 определить, у какой из трех пар чисел  $X_i, Y_i$  наименьшее значение выражения  $X_i - Y_i$
- 09 найти гипотенузу прямоугольного треугольника с катетом  $A$  и площадью  $S$  и сравнить ее с  $S_1$
- 10 определить меньший из углов прямоугольного треугольника с катетами  $A$  и  $B$
- 11 определить, могут ли отрезки с длинами  $a, b, c$  быть вершинами треугольника
- 12 найти длину первого слова, начинающегося на  $Z$ , во введенной строке.  
Слова разделяются пробелами
- 13 найти количество символов, отличных от пробела, во введенной строке
- 14 найти максимальное по абсолютной величине из введенных чисел
- 15 найти минимальное по абсолютной величине из введенных чисел
- 16 найти минимальное среди введенных чисел
- 17 массив размером  $3 \times 5$  заполнить случайными числами от  $-10$  до  $10$  и найти количество положительных элементов
- 18 массив размером  $5 \times 4$  заполнить случайными числами от  $-10$  до  $10$  и найти количество неположительных элементов
- 19 массив размером  $3 \times 5$  заполнить случайными числами от  $-10$  до  $10$  и найти наибольший элемент и его индексы
- 20 массив размером  $4 \times 3$  заполнить случайными числами от  $0$  до  $30$  и найти количество неотрицательных элементов
- 21 массив размером  $3 \times 5$  заполнить случайными числами от  $-20$  до  $0$  и найти наибольший по абсолютной величине элемент и его индексы
- 22 сравнить количество введенных строк с  $N_1$
- 23 посчитать количество четных цифр во введенном числе
- 24 посчитать количество семерок во введенном числе
- 25 заполнить одномерный массив размерностью  $10$  случайными числами от  $0$  до  $90$  и элементы переставить в обратном порядке
- 26 найти след (сумму элементов главной диагонали) матрицы  $A$  размерностью  $4 \times 4$ ,

- заполненной случайными числами от -5 до 35
- 27 сравнить суммы введенных чисел с четными и нечетными номерами
  - 28 вывести две введенные строки в алфавитном порядке
  - 29 вывести только наиболее короткую из введенных строк
  - 30 вывести последнюю в алфавитном порядке из введенных строк
  - 31 выяснить, больше ли 3 чем число букв 'd' во введенной строке
  - 32 выяснить, сколько раз встречается сочетание 'the' во введенной строке
  - 33 определить, сколько из введенных строк заканчиваются '.'
  - 34 определить, сколько из введенных строк начинаются на 'Z'
  - 35 определить процентное содержание пробелов во введенной строке
  - 36 определить процентное содержание буквы Q во введенной строке (без учета пробелов)
  - 37 заполнить одномерный массив [1..10] случайными числами от 0 до 90 и определить сумму элементов с четными индексами
  - 38 заполнить одномерный массив [1..10] случайными числами от -20 до 20 и определить количество положительных элементов с четными индексами
  - 39 найти количество четных чисел среди введенных
  - 40 найти количество нечетных чисел среди введенных
  - 41 сравнить количество четных и нечетных среди введенных чисел
  - 42 массив размером 3 x 5 заполнить случайными числами от -10 до 10 и сравнить количество положительных и отрицательных элементов
  - 43 массив размером 3 x 5 заполнить случайными числами от -10 до 10 и сравнить суммы положительных и отрицательных элементов
  - 44 массив размером 3 x 5 заполнить случайными числами от -10 до 10 и сравнить количество положительных элементов с N1
  - 45 массив размером 4 x 5 заполнить случайными числами от -10 до 10 и найти сумму элементов, у которых второй индекс больше первого

### **Семестр 1, тема 1.2:**

- 01 Определить количество строк файла 'f.txt'.
- 02 Как открыть файл 'qwert.1' для чтения?
- 03 Как открыть файл 'a1.txt' для записи?
- 04 Как открыть файл 'as22.t' для расширения (добавления к текущему содержимому)?
- 05 Написать функцию, определяющую количество строк текстового файла 'abc.txt', в которых не содержится 'abcd'
- 06 Написать функцию, определяющую номер строки с минимальным числом текстового файла 'd.txt', содержащего в каждой строке целое число.
- 07 Написать функцию, определяющую количество строк текстового файла 'abc.txt', в которых содержится 'abcd'
- 08 Как прочитать информацию последовательно из нескольких файлов, используя одну файловую переменную?
- 09 Дописать к файлу его пятую строку, используя одну файловую переменную.
- 10 Написать подпрограмму, которая выводит на экран имя файла и количество строк в нем по заданному имени файла.
- 11 Написать функцию, которая возвращает третью строку файла



- с именем, заданным в качестве параметра.
- 12 Написать функцию, которая возвращает последнюю строку файла с именем, заданным в качестве параметра.
  - 13 Написать функцию, которая возвращает длину третьей строки файла с именем, заданным в качестве параметра.
  - 14 Написать функцию, которая возвращает длину последней строки файла с именем, заданным в качестве параметра.
  - 15 Написать функцию, которая возвращает количество строк файла с именем, заданным в качестве параметра.
  - 16 Написать функцию, которая возвращает количество длинных (с длиной > 10 символов) строк файла 'abc.txt'.
  - 17 Написать функцию, которая возвращает количество коротких (с длиной < 10 символов) строк файла 'abc.txt'.
  - 18 Написать функцию, которая возвращает количество содержащих пробел строк файла 'abc.txt'.
  - 19 Написать функцию, которая возвращает количество не содержащих пробел строк файла 'abc.txt'.
  - 20 Написать функцию, которая возвращает True, если хотя бы одна строка файла 'abc.txt' содержит 'abc'.
  - 21 Написать функцию, определяющую номер строки с максимальным числом текстового файла 'd.txt', содержащего в каждой строке целое число.
  - 22 Дописать первые 3 строки файла 'c:\1\al.txt' к файлу 'c:\1\d.txt'
  - 23 Дописать последние 3 строки файла 'c:\1\al.txt' к файлу 'c:\1\d.txt'
  - 24 Записать первые 3 строки файла 'c:\1\al.txt' в файл 'c:\1\d.txt'
  - 25 Записать последние 3 строки файла 'c:\1\al.txt' в файл 'c:\1\d.txt'
  - 26 Дописать с 4 по 6 строки файла 'c:\1\al.txt' к файлу 'c:\1\d.txt'
  - 27 Дописать с 4 по 6 строки файла 'c:\1\al.txt' к файлу 'c:\1\d.txt'
  - 28 Записать с 4 по 6 строки файла 'c:\1\al.txt' в файл 'c:\1\d.txt'
  - 29 Записать с 4 по 6 строки файла 'c:\1\al.txt' в файл 'c:\1\d.txt'
  - 30 Определить, какая из четных строк файла 'qq.txt' имеет минимальную длину.
  - 31 Определить, какая из четных строк файла 'qq.txt' имеет максимальную длину.
  - 32 Определить, используются ли массивы в программе на языке Pascal, находящейся в файле 'test1.pas'.

## Семестр 2:

Для каждого обучающегося выбирается одна из комбинаций вариантов, например 1А/З2/5Б/6А/7В/8А — Это Приложение с графическим интерфейсом для ввода из входного типизированного файла заданного количества целых чисел, сортировки их методом пузырька и записи в выходной, отличный от входного, текстовый файл.

- 1А Приложение с графическим интерфейсом
- 1Б Консольное приложение

Z1) для записи в 2А текстовый файл  
2Б типизированный файл  
3А упорядоченных по возрастанию  
3Б упорядоченных по убыванию  
3В случайных

4А целых чисел  
4Б действительных чисел

Z2) для ввода из входного 5А текстового файла  
5Б типизированного файла  
заданного количества  
6А целых чисел  
6Б действительных чисел

сортировки их  
7А методом вставки  
7Б методом выбора  
7В методом пузырька

и вывода в выходной, отличный от входного  
8А текстовый файл  
8Б типизированный файл

Z3) для ввода из входного 9А текстового файла  
9Б типизированного файла  
заданного количества  
10А целых чисел  
10Б действительных чисел

и вывода их на экран для просмотра  
(только приложение с графическим интерфейсом)

\* \* \*

на следующем этапе эти программы будут использоваться для сравнения алгоритмов сортировки по эффективности.

NВ для сортировок продумать, как определять количество сравнений и пересылок, какие части следует оформить в виде подпрограмм.

## **5 Методические материалы, определяющие процедуры оценивания компетенции**

### **5.1 Критерии оценивания качества выполнения лабораторного практикума**

Оценка «зачтено» выставляется обучающемуся, если программа решает поставленную задачу и обучающийся ответил на все вопросы, поставленные преподавателем на защите. Оценка «незачтено» выставляется обучающемуся, если программа не решает поставленную задачу или обучающийся не проявил глубоких теоретических знаний при защите работы.

### **5.2 Критерии оценивания качества устного ответа**

Оценка «отлично» выставляется за глубокое знание предусмотренного программой материала, за умение четко, лаконично и логически последовательно отвечать на поставленные вопросы.

Оценка «хорошо» – за твердое знание основного (программного) материала, за грамотные, без существенных неточностей ответы на поставленные вопросы.

Оценка «удовлетворительно» – за общее знание только основного материала, за ответы, содержащие неточности или слабо аргументированные, с нарушением последовательности изложения материала.

Оценка «неудовлетворительно» – за незнание значительной части программного материала, за существенные ошибки в ответах на вопросы, за неумение ориентироваться в материале, за незнание основных понятий дисциплины.

### **5.3 Критерии оценивания тестирования**

При тестировании все верные ответы берутся за 100%.

90%-100% отлично

75%-90% хорошо

60%-75% удовлетворительно

менее 60% неудовлетворительно

### **5.4 Критерии оценивания качества выполнения контрольной работы**

Оценка «зачтено» выставляется, если представленные программы решают поставленную задачу и обучающийся может обосновать достоверность результата.

Оценка «незачтено» – если представленные программы не решают поставленную задачу или обучающийся не может обосновать достоверность результата.

### **5.5 Критерии оценивания результатов освоения дисциплины**

Оценка «отлично» выставляется, если обучающийся **знает** источники информации об используемом языке программирования (Pascal) и среде разработки (Lazarus); основные возможности среды разработки. **Умеет** настраивать параметры среды разработки, формализовать поставленную задачу; разработать структуры данных для представления предметной области решаемой задачи; реализовать выбранный алгоритм на языке высокого уровня (Pascal); использовать стандартные и создавать при необходимости свои

процедуры и функции.

**Владеет навыками** создания консольных приложений и приложений с графическим интерфейсом решающих задачи разного профиля, в том числе работающих с файлами; навыками тестирования и отладки создаваемых приложений.

Оценка **«хорошо»** – если обучающийся **знает** источники информации об используемом языке программирования (Pascal) и среде разработки (Lazarus); основные возможности среды разработки. **Умеет** настраивать параметры среды разработки, использовать структуры данных для представления предметной области решаемой задачи; реализовать выбранный алгоритм на языке высокого уровня (Pascal); использовать стандартные и создавать при необходимости свои процедуры и функции. **Владеет** с отдельными пробелами навыками алгоритмизации и применения языка Pascal для создания приложений, решающих задачи разного профиля, в том числе работающих с файлами;; навыками тестирования и отладки создаваемых приложений.

**«удовлетворительно»** – если обучающийся **знает** источники информации об используемом языке программирования (Pascal) и среде разработки (Lazarus); основные возможности среды разработки. **Умеет** настраивать параметры среды разработки, использовать структуры данных для представления предметной области решаемой задачи; реализовать выбранный алгоритм на языке высокого уровня (Pascal); использовать стандартные и создавать при необходимости свои процедуры и функции. Имеются недочеты при работе с подпрограммами. **Владеет** отдельными навыками алгоритмизации и применения языка Pascal для создания приложений, решающих задачи разного профиля, в том числе работающих с файлами.

Оценка **«неудовлетворительно»** – если обучающийся **не знает** источники информации об используемом языке программирования (Pascal) и среде разработки (Lazarus); основные возможности среды разработки, **не умеет** настраивать параметры среды разработки, использовать структуры данных для представления предметной области решаемой задачи; реализовать выбранный алгоритм на языке высокого уровня (Pascal); использовать стандартные и создавать при необходимости свои процедуры и функции, **не владеет** навыками алгоритмизации и применения языка Pascal для создания приложений, решающих задачи разного профиля, в том числе работающих с файлами;; навыками тестирования и отладки создаваемых приложений.