

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

«УТВЕРЖДАЮ»

Проректор по учебной работе

Г.Ю. Нагорная

« 30 » 03



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 Прикладной математики и информационных технологий

Кафедра разработчик РПД Математика

Выпускающая кафедра Математика

Начальник
учебно-методического управления

Семенова Л.У.

Директор института ПМ и ИТ

Тебуев Д.Б.

Заведующий выпускающей кафедрой

Кочкаров А.М.

г. Черкесск, 2022 г.

Содержание

1	Цели освоения дисциплины	4
2	Место дисциплины в структуре образовательной программы	5
3	Планируемые результаты обучения по дисциплине	6
4	Структура и содержание дисциплины	7
	4.1. Объем дисциплины и виды учебной работы	7
	4.2. Содержание дисциплины	8
	4.2.1. Разделы (темы) дисциплины, виды учебной деятельности и формы контроля	8
	4.2.2. Лекционный курс	9
	4.2.3. Лабораторный практикум	10
	4.2.4. Практические занятия	11
	4.3. Самостоятельная работа обучающегося	13
5	Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	14
6	Образовательные технологии	17
7	Учебно-методическое и информационное обеспечение дисциплины	
	7.1. Перечень основной и дополнительной учебной литературы	18
	7.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»	19
	7.3 Информационные технологии	20
8	Материально-техническое обеспечение дисциплины	21
	8.1. Требования к аудиториям (помещениям, местам) для проведения занятий	21
	8.2. Требования к оборудованию рабочих мест преподавателя и обучающихся:	22
	8.3. Требования к специализированному оборудованию	22
9	Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья	23
	Приложение 1. Фонд оценочных средств	24
	Приложение 2. Аннотация рабочей программы	65
	Рецензия на рабочую программу	66
	Лист переутверждения рабочей программы дисциплины	67

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

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	Архитектура ЭВМ
2.		Операционные системы и сетевые технологии
3.		Технологии программирования в Internet
4.		Объектно – ориентированное программирование
5.		Технологии разработки Web-серверов

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

№ п/п	Номер/ индекс компетенции	Наименование компетенции (или ее части)	В результате изучения дисциплины обучающиеся должны:
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 работы

Вид учебной работы		Семестр	
		Всего часов	№ 1, часов
1		2	3
Аудиторная контактная работа (всего)		54	54
В том числе:			
Лекции (Л)		18	18
Лабораторные работы (ЛР)			
Практические занятия (ПЗ)		36	36
Контактная внеаудиторная работа, в том числе		1,5	1,5
Индивидуальные и групповые консультации			
Самостоятельная работа обучающегося (СРО) (всего)		52	52
<i>Контрольная работа</i>		20	20
<i>Подготовка к лабораторным работам (ЛР)</i>		-	-
<i>Подготовка к практическим работам (ПР)</i>		20	20
<i>Работа с книжными и электронными источниками</i>		10	10
<i>Подготовка к промежуточному тестовому контролю</i>		2	2
Промежуточная аттестация	Зачет с оценкой, в том числе	ЗаО	ЗаО
	Прием зачета с оценкой, час	0,5	0,5
	СРО, час.		
ИТОГО: Общая трудоемкость	часов	108	108
	зач. ед.	3	3

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	СРО	все го	
1	2	3	4	5	6	7	8
Семестр 1							
1.	Раздел 1. Основы алгоритмизации Семестр 1	4		6	10	20	Контроль, контрольные вопросы, контрольная работа, тестирование
2.	Раздел 2. Язык Pascal	4		10	20	34	Текущий тестовый контроль, контрольные вопросы, контрольная работа
3.	Раздел 3. Сортировка и поиск	4		10	20	34	Текущий тестовый контроль, контрольные вопросы, контрольная работа
4.	Раздел 4. Введение в БД	6		10	2	18	Текущий тестовый контроль, контрольные вопросы
5	Контактная внеаудиторная работа					1,5	индивидуальные и групповые консультации, контрольная работа, контрольная работа
	Промежуточная аттестация					0,5	ЗаО
ИТОГО часов в 1 семестре:		18		36	52	108	ЗаО
Всего:		18		36	52	108	

4.2.2. Лекционный курс

№ п/п	Наименование раздела дисциплины	Наименование темы лекции	Содержание лекции	Всего часов (ОФО)
1	2	3	4	5
Семестр 1				
1	Раздел 1. Основы алгоритмизации	Тема 1.1 Краткая история развития ВТ и ПО. Тема 1.2 Этапы решения задач на ЭВМ	История развития ВТ и ПО. Архитектура ЭВМ. В т.ч. Алгоритмы. Свойства. Способы описания. Оценка эффективности. В т.ч. операторы присваивания, ветвления, циклы. Интервальные, перечисляемые типы.	4
2	Раздел 2. Язык Pascal	Тема 2.1 Структура программы. Основные операторы и типы данных. Тема 2.2 Типы данных, определяемые пользователем Подпрограммы. Графика Файлы. Текстовые файлы Структурированные типы данных Файлы. Типизированные и не типизированные файлы.	Процедуры, функции, рекурсия, модули Системы координат. Область рисования. Построение фигур из отрезков. Стандартные функции построения фигур. Работа с цветом. Текстовые файлы. Ввод/вывод строк. Режимы доступа к файлу. Множество, запись, сравнение с массив, строка Типизированные файлы. Использование записей в типизированных файлах	4
	Раздел 3. Сортировка и поиск	Тема 3.1 Поиск. Тема 3.2 Сортировка. Простые методы сортировки. Метод вставки. Тема 3.3 Сортировка. Улучшенные методы сортировки. Быстрая сортировка. Указатели.	Последовательный поиск. Поиск с барьером. Двоичный поиск Прикладное значение сортировки. Обзор простых методов сортировки. Метод вставки. Улучшенные методы	4

		Динамические структуры данных Использование стандартных компонент Lazarus	сортировки. Быстрая сортировка. Отличие адреса и содержимого. Указатели. Динамические структуры данных Edit, Label, Button, Memo. Общие свойства и методы компонент. Визуальные и не визуальные компоненты.	
8	Раздел 4. Введение в БД	Тема 4.2 Работа с БД через стандартные компоненты	Работа с базами типа dbf. Базы с поддержкой SQL. Компоненты типа Connection, Transaction, SQLQuery. Использование DBGrid, DataSource.	6
ИТОГО часов в 1 семестре:				18
Всего:				18

4.2.3. Лабораторный практикум *Не предусмотрен*

4.2.4. Практические занятия

№ п/п	Наименование раздела дисциплины	Наименование практического занятия	Содержание практического занятия	Всего часов (ОФО)
1	2	3	4	5
Семестр 1				
1.	Раздел 1. Основы алгоритмизации	1.1. Представление информации в ЭВМ. Основные операции. 1.2. Этапы решения задач на ЭВМ 1.3 Структура программы. Числовые типы данных. Ветвления	Единицы информации. Системы счисления. Логические и числовые операции. Формализация задач. Выбор математической модели. Построение и анализ алгоритма. Структура программы. Целые и вещественные типы данных. Оператор присваивания. Операции с числовыми типами данных.	6
2.	Раздел 2. Язык Pascal	2.1 Циклы. Массивы.	Операторы for, while,	

		Строки Условный и составной операторы. Оператор Case	repeat. Признак EOF. Описание. Индексы. Обращение к элементу массива. Сравнение строк, конкатенация. Стандартные строковые функции.	2
3	Раздел 2. Язык Pascal	2.2. Графика	Системы координат и их преобразование. Пиксель. Цвет.	2
4	Раздел 2. Язык Pascal	2.3 Стандартные графические подпрограммы. Текстовые файлы Множества	Открытие, чтение, запись, закрытие В т.ч. использование для управления логикой программы.	2
5	Раздел 2. Язык Pascal	2.4 Типизированные файлы, нетипизированные файлы.	Записи. Открытие, чтение, запись, закрытие типизированных файлов Нетипизированные файлы. Произвольный доступ к данным. Эффективность нетипизированных обращений.	4
3 4	Раздел 3. Сортировка и поиск	Поиск Сортировка методом вставки, пузырька	Двоичный поиск. Сортировка методом вставки, пузырька	4
5 6	Раздел 3. Сортировка и поиск	Быстрая сортировка Указатели. Динамические структуры данных.	Быстрая сортировка Список, очередь, стек	4
7 8	Раздел 3. Сортировка и поиск	Использование стандартных компонент Lazarus Сравнение алг. сортировки	Edit, Label, Button, Memo. Общие свойства и методы компонент. Визуальные и не визуальные компоненты. Анализ и сравнение методов сортировки.	2
9	Раздел 4. Введение в БД	Базы данных	Таблицы. Чтение, изменение, добавление, удаление записей	10
ИТОГО часов в 1 семестре:				36
Всего:				36

4.3. САМОСТОЯТЕЛЬНАЯ РАБОТА ОБУЧАЮЩЕГОСЯ

№ п/п	Наименование раздела дисциплины	№ п/п	Виды СРО	Всего часов (ОФО)
1	2	3	4	5
Семестр 1				
1	Раздел 1. Основы алгоритмизации	1.1.	Подготовка к практическим занятиям., в т.ч. чтение конспекта лекций, формулирование вопросов.	6
		1.2	Самоподготовка, в т.ч. чтение доп. Литературы и электронных ресурсов.	2
		1.3	Подготовка к текущему контролю	2
2	Раздел 2. Язык Pascal	2.1.	Выполнение контрольной работы.	6
		2.2.	Подготовка к практическим и лабораторным занятиям, в т.ч. чтение конспекта лекций, формулирование вопросов.	6
		2.3	Самоподготовка, в т.ч. чтение доп. Литературы и электронных ресурсов, посещение форумов по программированию	6
		2.4	Подготовка к текущему контролю	2
5	Раздел 3. Сортировка и поиск	5.1	Подготовка к практическим и лабораторным занятиям, в т.ч. чтение конспекта лекций, формулирование вопросов.	6
		5.2	Выполнение контрольной работы.	4
		5.3	Самоподготовка, в т.ч. чтение доп. Литературы и электронных ресурсов, посещение форумов по программированию	6
		5.4	Подготовка к текущему контролю	4
6	Раздел 4. Введение в БД	6.1	Подготовка к практическим и лабораторным занятиям, в т.ч. чтение конспекта лекций, формулирование вопросов. посещение форумов по программированию и базам данных.	2
ИТОГО часов в1 семестре:				52
Всего:				52

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

5.1. Методические указания для подготовки обучающихся к лекционным занятиям

Основными формами обучения программированию для ЭВМ 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.

5.3. Методические указания для подготовки обучающихся к практическим занятиям

В процессе подготовки и проведения практических занятий обучающиеся закрепляют полученные ранее теоретические знания, приобретают навыки их практического применения, опыт рациональной организации учебной работы, готовятся к сдаче зачета, экзамена.

Для подготовки к практическому занятию следует повторить конспект по данной теме, а также ознакомиться с данной темой по рекомендованным литературным источникам. Это поможет активно участвовать в обсуждениях, предлагать свои варианты создания программы, проверить свои знания при ответах на вопросы преподавателя. Особо важной темой является «Подпрограммы». Обучающийся изучения данной темы должен научиться точно формулировать и понимать основные понятия подпрограмм. Например, отличия функций и процедур, определения формальных и фактических параметров, особенности использования локальных и глобальных переменных и т.п. Также нужно пробовать выделять различные части программы в отдельные подпрограммы. Без понимания подпрограмм невозможно будет освоить работу с графикой, сортировкой и поиском, базами данных. Без подпрограмм не обходится практически ни одно приложение для решения реальных задач.

Для точного представления об особенностях реализации языка программирования необходимо обращаться к документации на компилятор. Оригинальная документация представлена на английском языке. Поэтому, хотя существуют переводы ее в том числе и на русский язык, желательно научиться понимать документацию на языке оригинала. На занятиях по английскому языку постарайтесь уделять этой теме особое внимание. Навыки чтения документации на английском нужны в наше время практически любому специалисту и по окончании ВУЗа.

Чтобы решать предлагаемые задачи, необходимо также знание школьного курса математики. Для первых занятий требуется повторить понятия логарифма, тригонометрических функций, свойства треугольника, способы решения квадратного уравнения. При подготовке к применению массивов — понятие матрицы, определителя, решение системы линейных уравнений. Перед изучением графики требуется вспомнить геометрию и уметь активно применять тригонометрические функции, чтобы делать

построение линий и преобразования координат. Для сравнения различных алгоритмов поиска и сортировки требуется понимание пределов.

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.

6. Образовательные технологии

№ п/п	Виды учебной работы	Образовательные технологии	Всего часов
1	3	4	
1	<i>Лекция.</i> Краткая история развития ВТ и ПО	Обзорная лекция. Презентация	2
2	<i>Лекция.</i> Этапы решения задач на ЭВМ	Лекция–беседа. Презентация	2
3	<i>Лекция.</i> Структура программы. Основные операторы и типы данных	Лекция – информация. Презентация	2
4	<i>Лекция.</i> Сортировка. Улучшенные методы сортировки. Быстрая сортировка	Лекция – беседа. Презентация	2
5	<i>Лекция.</i> Использование стандартных компонент Lazarus	Лекция – беседа. Презентация	2
6	<i>Лекция.</i> Работа с БД через стандартные компоненты	Лекция – информация. Презентация	2
7	<i>Практическое занятие</i>	Коллективный разбор конкретных ситуаций.	2
8	<i>Практическое занятие</i>	Коллективный разбор конкретных ситуаций.	2
9	<i>Практическое занятие</i> Сравнение алгоритмов сортировки	Учебно-исследовательская работа.	2
ИТОГО часов в 1 семестре:			18
Всего:			18

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

7.1. Перечень основной и дополнительной учебной литературы

Основная литература

- 1 Кучунова Е.В. Программирование. Процедурное программирование : учебное пособие / Кучунова Е.В., Олейников Б.В., Чередниченко О.М.. — Красноярск : Сибирский федеральный университет, 2016. — 92 с. — ISBN 978-5-7638-3555-7. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/84094.html>
- 2 Павловская Т.А. Программирование на языке высокого уровня Паскаль : учебное пособие / Павловская Т.А.. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 153 с. — ISBN 978-5-4497-0864-9. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/102052.html>
- 3 Маркин А.В. Web-программирование : учебник / Маркин А.В.. — Москва : Ай Пи Ар Медиа, 2021. — 286 с. — ISBN 978-5-4497-1002-4. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/104883.html>
- 4 Ачкасов В.Ю. Программирование баз данных в Delphi : учебное пособие / Ачкасов В.Ю.. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 431 с. — ISBN 978-5-4497-0942-4. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/102047.html>
- 5 Гунько А.В. Программирование (в среде Windows) : учебное пособие / Гунько А.В.. — Новосибирск : Новосибирский государственный технический университет, 2019. — 155 с. — ISBN 978-5-7782-3890-9. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/99209.html>

7.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

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

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

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
Консультант Плюс	Договор № 272-186/С-23-01 от 20.12.2022 г.
ЭБС IPR SMART	Лицензионный договор № 9368/22П от 11.06.2022 г. Срок действия: с 01.07.2022 до 01.07.2023

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

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

1. Учебная аудитория для проведения занятий лекционного типа

Ауд.243. Специализированная мебель:

Кафедра настольная - 1 шт., доска меловая - 1 шт., стулья – 65 шт., парты - 34шт.

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

Экран на штативе – 1 шт.

Проектор – 1 шт.

Ноутбук – 1 шт.

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

Ауд.251. Специализированная мебель:

Стол преподавательский - 1 шт., компьютерные столы - 10шт., парты -7шт., стулья - 24шт., доска меловая - 1шт.

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

Персональный компьютер – 10 шт.

Экран настенный рулонный – 1 шт.

3. Лаборатория компьютерной графики

Ауд.251. Специализированная мебель:

Стол преподавательский - 1 шт., стул мягкий - 1 шт., доска меловая - 1 шт., парты - 10шт., компьютерные столы - 11шт., стулья - 21 шт.,

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

Персональный компьютер – 11шт.

Экран рулонный настенный – 1 шт.

Проектор – 1 шт.

8.2. Требования к оборудованию рабочих мест преподавателя и обучающихся

Рабочее место преподавателя, оснащенное компьютером с ОС MSWindows

Рабочие места обучающихся, оснащенные компьютерами с ОС MSWindows

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	Формируемые компетенции (коды)
	ПК-3
Раздел 1. Основы алгоритмизации	+
Раздел 2. Язык Pascal	+
Раздел 3. Сортировка и поиск	+
Раздел 4. Введение в БД	+

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 компетенции	Критерии оценивания результатов обучения				Средства оценивания результатов обучения	
	неудовлетв	удовлетв	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	Допускает существенные ошибки в знаниях стандартных пакетов прикладных программ для решения практических задач защиты информации, отлаживать и тестировать программы по алгоритмам шифрования; пробелы в знаниях элементов теории чисел.	Демонстрирует некоторые способности в знаниях стандартных пакетов прикладных программ для решения практических задач защиты информации, отлаживать и тестировать программы по алгоритмам шифрования, а также криптографическим методам и средств обеспечения информационной безопасности.	Демонстрирует способность в знаниях стандартных пакетов прикладных программ для решения практических задач защиты информации, отлаживать и тестировать программы по алгоритмам шифрования, а также криптографическим методам и средств обеспечения информационной безопасности	Демонстрирует профессиональную способность к языкам программирования, для целей автоматизации шифрования информации; методов кодирования; знания о государственной политике в области безопасности компьютерных систем; знания элементов теории чисел, составляющие математическую основу криптографических методов.	Текущий тестовый контроль, контрольные вопросы, контрольная работа	Зачет с оценкой
ПК-3.2 Анализирует стандартные технические требования к проектированию программно-обеспечения.	Допускает частично освоенное умение пользоваться стандартными и программами для решения практических задач защиты информации; отлаживать и тестировать программы по	Демонстрирует в целом удовлетворительные, но не систематизированные умения пользоваться стандартными программами для решения практических задач защиты информации; отлаживать и тестировать программы по	Демонстрирует в целом хорошие, но содержащие отдельные пробелы в использовании стандартных программ для решения практических задач защиты информации; отлаживать и тестировать	Демонстрирует умения в использовании стандартных программ для решения практических задач защиты информации; отлаживать и тестировать программы по алгоритмам шифрования; применять	Текущий тестовый контроль, контрольные вопросы, контрольная работа	Зачет с оценкой

	алгоритмам шифрования; применять криптографические методы и средства.	алгоритмам шифрования; применять криптографические методы чисел для моделирования задач защиты информации.	программы по алгоритмам шифрования; применять элементы теории чисел для моделирования задач криптографии	элементы теории чисел для задач криптографических методов и средств обеспечения информационной безопасности.		
П К - 3 . 3 Способен осуществлять разработку технических требований к программному обеспечению, при его создании	Фрагментарно владеет некоторыми пакетами стандартных программ для решения задач защиты информации; не владеет интернет технологиям и для организации защиты информации, не владеет возможностями математической основы криптографии и теории чисел.	Владеет отдельными навыками работы со стандартными программам для решения задач защиты информации; технологиями отладки и тестирования программ по алгоритмам шифрования; элементов теории чисел для моделирования задач защиты информации.	Демонстрирует в целом успешные знания по стандартным программам для решения практических задач защиты информации; технологии отладки и тестирования программ по алгоритмам шифрования; по криптографическим методам и средствам обеспечения информационной безопасности.	Демонстрирует профессиональные навыки работы в специализированных стандартных пакетах программ для решения практических задач защиты информации; владеет технологиями отладки и тестирования программ по алгоритмам шифрования; владеет способностью применять элементы теории чисел для решения задач криптографии.	Текущий тестовый контроль, контрольные вопросы, контрольная работа	Зачет с оценкой

4. Комплект контрольно-оценочных средств по дисциплине

Вопросы к зачету

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 булевского 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.

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 рекурсивную функцию вычисления факториала.

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 `whiletruedox:=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 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 S1
- 02 найти радиус окружности с площадью S и сравнить его с R1
- 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 $x^2 + x \cdot \sin(R) + Q = 0$.
- 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 C1

- 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 x 5 заполнить случайными числами от -10 до 10
и найти количество положительных элементов
- 18 массив размером 5 x 4 заполнить случайными числами от -10 до 10
и найти количество неположительных элементов
- 19 массив размером 3 x 5 заполнить случайными числами от -10 до 10
и найти наибольший элемент и его индексы
- 20 массив размером 4 x 3 заполнить случайными числами от 0 до 30
и найти количество неотрицательных элементов
- 21 массив размером 3 x 5 заполнить случайными числами от -20 до 0
и найти наибольший по абсолютной величине элемент и его индексы
- 22 сравнить количество введенных строк с N1
- 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 4x4,
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\1.txt' к файлу 'c:\1\d.txt'

- 23 Дописать последние 3 строки файла 'с:\1\а1.txt' к файлу 'с:\1\d.txt'
- 24 Записать первые 3 строки файла 'с:\1\а1.txt' в файл 'с:\1\d.txt'
- 25 Записать последние 3 строки файла 'с:\1\а1.txt' в файл 'с:\1\d.txt'
- 26 Дописать с 4 по 6 строки файла 'с:\1\а1.txt' к файлу 'с:\1\d.txt'
- 27 Дописать с 4 по 6 строки файла 'с:\1\а1.txt' к файлу 'с:\1\d.txt'
- 28 Записать с 4 по 6 строки файла 'с:\1\а1.txt' в файл 'с:\1\d.txt'
- 29 Записать с 4 по 6 строки файла 'с:\1\а1.txt' в файл 'с:\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.

NB 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.

Оценка «**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.