

Министерство просвещения РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Дагестанский государственный педагогический университет им. Р. Гамзатова»
Факультет профессионально-педагогического образования
Кафедра интеллектуальных систем и цифровой экономики


УТВЕРЖДАЮ
И.И. начальника УМУ
И.И. Сагиреев И.И.
« » 2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.О.06 Модуль «Предметно-методический»

Б1.О.06.16 Языки и системы программирования

Направление подготовки 44.03.04 – *Профессиональное обучение*

Профиль подготовки *Информационные технологии*

Квалификация *Бакалавр*

Формы обучения: *очная; заочная*

Год приёма – 2025

Форма обучения	Курс	Семестр	Количество часов					Форма итоговой аттестации
			Трудоёмкость	Лекции	Лабораторные занятия	Промежуточный контроль	Самостоятельная работа	
Очная	3	5	144	32	32	27	53	Экзамен
Заочная	3	5	144	8	8	6	122	Экзамен

Махачкала, 2025

I. Цель и задачи дисциплины

Цель дисциплины:

– изучение языков и методов программирования для их осознанного использования в профессиональной деятельности в будущем.

Задачи дисциплины:

- сформировать представление о современной методологии проектирования и программирования, принципах трансляции;
- научить студентов правильно ставить задачи и структуризировать информационные проблемы, которые должны быть практически разрешены путем применения тех или иных языков программирования;
- научить студентов практически разрабатывать программное обеспечение, а также управлять информационными процессами в программах как общего, так и учебного назначения;
- сформировать алгоритмический стиль мышления студентов.

II. Место дисциплины в структуре ОПОП ВО

Учебная дисциплина «Языки и системы программирования» относится к обязательной части учебного плана, Модуля "Предметно-содержательный" по направлению Профессиональное обучение

Для изучения дисциплины необходимы компетенции, сформированные у студентов в результате освоения дисциплин: математика; информатика.

Дисциплины, для которых данная дисциплина является предшествующей:

- математические основы информатики;
- компьютерная и абстрактная алгебра;
- компьютерное моделирование;
- основы дискретной математики;
- методы оптимизации;
- алгоритмы решения экстремальных задач;
- комбинаторные алгоритмы;
- рекурсивные алгоритмы в курсе информатики;
- базы данных на Delphi;
- реализация численных методов на ЭВМ;
- исследование операций.

III. Планируемые результаты обучения по дисциплине

результате освоения содержания программы у бакалавра должны быть сформированы компетенции:

Формируемые компетенции	Перечень планируемых результатов обучения по дисциплине (Код и наименование индикатора достижения компетенции)
Профессиональные компетенции	
ПК-6. Способен использовать математические методы, алгоритмы и современные компьютерные технологии для поиска, хранения, обработки и передачи информации.	Знает: 3-ПК-6.1. Основные понятия курса высшей математики. 3-ПК-6.2. Понятие алгоритма, виды, свойства, способы записи, описания и уточнения. 3-ПК-6.3. Математические методы обработки информации с использованием компьютерных технологий. 3-ПК-6.4. Методы формализованных исчислений. 3-ПК-6.5. Алгоритмы решения прикладных задач. 3-ПК-6.6. Создание программных файлов и хранимых процедур для работы с БД. 3-ПК-6.7. Методы обработки информации настольными

	<p>издательскими системами и сетевыми технологиями.</p> <p>3-ПК-6.8.Численные методы решения практических задач.</p> <p>3-ПК-6.9.Технологии программирования.</p> <p>3-ПК-6.10.Пакеты прикладных программ.</p> <p>3-ПК-6.11.Подготовка проектов БД.</p> <p>3-ПК-6.12.Разработка современных приложений – IDE.</p> <p>3-ПК-6.13.Гипертекстовая система WWW.</p> <p>Умеет:</p> <p>У-ПК-6.1.Подбирать математические методы и алгоритмы для компьютерной обработки информации.</p> <p>У-ПК-6.1.Анализировать содержание задач и создавать их формализованное представление.</p> <p>У-ПК-6.2.Подбирать алгоритмы решения задач, составлять и отлаживать программы на ПК.</p> <p>У-ПК-6.3.Проектировать БД и создавать простые приложения.</p> <p>У-ПК-6.4.Обрабатывать информацию и создавать мультимедийные презентации.</p> <p>У-ПК-6.5.Использовать сетевые технологии.</p> <p>У-ПК-6.6.Применять численные методы, технологии программирования и навыки работы с математическими пакетами для решения практических задач.</p> <p>У-ПК-6.7.Разрабатывать АРМ на базе современных СУБД.</p> <p>У-ПК-6.8.Работать с гипертекстовой системой WWW и создавать Web-страницы.</p> <p>Владеет:</p> <p>В-ПК-6.1.Математическими методами и алгоритмами компьютерной обработки информации.</p> <p>В-ПК-6.2.Приемами формализации прикладных задач.</p> <p>В-ПК-6.3.Приемами разработки программ для решения практических задач на компьютере;</p> <p>В-ПК-6.4.Навыками работы с БД, обработки информации, создания приложений.</p> <p>В-ПК-6.5.Способами применения численных методов и технологий программирования для решения прикладных задач.</p> <p>В-ПК-6.6.Технологиями проектирования и реализации БД и приложений на ПК.</p> <p>В-ПК-6.7.Методами работы в гипертекстовой системе WWW и технологиями создания Web-страниц</p>
<p>ПК-12. Способен применять в профессиональной деятельности современные языки программирования и языки баз данных</p>	<p>Знает:</p> <p>3-ПК-12.1.Основные алгоритмические конструкции.</p> <p>3-ПК-12.2.Назначение вспомогательных алгоритмов.</p> <p>3-ПК-12.3.Основные свойства величин в алгоритмах обработки информации.</p> <p>3-ПК-12.4.Назначение языков программирования.</p> <p>3-ПК-12.5.Основы создания баз данных и работы с ними.</p> <p>3-ПК-12.6.Различие между языками программирования высокого уровня и машинно-ориентированными языками.</p> <p>3-ПК-12.7.Правила представления данных и записи программ.</p> <p>3-ПК-12.8.Назначение систем программирования.</p> <p>3-ПК-12.9.Содержание этапов разработки программ.</p> <p>3-ПК-12.10.Способы создания баз данных и работа с ними.</p> <p>3-ПК-12.11.Технологии программирования для решения профессиональных задач.</p> <p>3-ПК-12.12.Технологии создания сложных баз данных.</p>

	<p>Умеет: У-ПК-12.1.Выполнять трассировку алгоритма для известного исполнителя. У-ПК-12.2.Определять и использовать вспомогательные алгоритмы. У-ПК-12.3.Составлять несложные алгоритмы и программы. У-ПК-12.4.Использовать созданные базы данных. У-ПК-12.5.Выполнять трассировку алгоритма для известного исполнителя. У-ПК-12.6.Определять и использовать вспомогательные алгоритмы. У-ПК-12.7.Составлять несложные алгоритмы и программы. У-ПК-12.8.Использовать созданные базы данных. У-ПК-12.9.Реализовать технологии программирования для решения профессиональных задач. У-ПК-12.10.Создавать сложные базы данных.</p> <p>Владеет: В-ПК-12.1.Основами разработки алгоритмов и составления программ. В-ПК-12.2.Основами создания баз данных. В-ПК-12.3.Правилами конструирования алгоритмов и составления программ для решения задач средней сложности. В-ПК-12.4.Способами создания простых баз данных. В-ПК-12.5.Технологиями программирования для решения профессиональных задач. В-ПК-12.6.Способами создания сложных баз данных.</p>
--	---

Таблица 1

IV. Объем дисциплины и виды учебной работы

Вид учебной работы	Всего часов	
	<i>Очная форма обучения</i>	<i>Заочная форма обучения</i>
	<i>5-й семестр</i>	<i>5-й семестр</i>
Аудиторные занятия (всего)	64	16
в том числе:	–	–
<i>лекции</i>	32	8
<i>лабораторные занятия</i>	32	8
<i>промежуточный контроль</i>	27	6
Самостоятельная работа (всего)	53	122
в том числе:		–
<i>курсовая работа</i>		
Итоговая аттестация	Экзамен	Экзамен
Общая трудоемкость (в часах)	144	144
в зачетных единицах	4	4

V. Содержание дисциплины

Таблица 2

V.1. Содержание разделов программы

№ п/п	Раздел программы	Содержание раздела
Модуль 1. Основные элементы языка программирования Турбо-Паскаль		
1.1	Основные этапы решения задач на ЭВМ	Постановка задачи. Математическое описание задачи. Метод решения. Разработка алгоритма. Составление блок-схемы. Реализация на ЭВМ (программирование). Отладка программы. Получение результатов. Обработка результатов. Диалоговые программы. Критерии качества программы. Дружественность. Классификация языков программирования. История развития языков программирования высокого уровня. Трансляция. Элементы языков программирования.
1.2	Основные понятия языка программирования Турбо-Паскаль	Интегрированная среда Turbo Pascal. Идентификаторы. Алфавит языка. Зарезервированные слова. Структура программы. Простые типы данных. Переменные. Типы переменных. Выражения и оператор присваивания. Контроль типов данных. Организация ввода-вывода. Форматы вывода. Директивы компилятора и управляющие символы.
1.3	Управляющие конструкции	Условные и безусловные операторы. Оператор выбора варианта CASE. Операторы цикла: с параметром FOR..., с предусловием While..., с постусловием Repeat... Вложенные циклы. Правила пунктуации.
1.4	Строковый тип данных. Типы данных, определяемые программистом	Тип STRING. Строковые выражения. Процедуры и функции для работы со строками. Перечисляемый тип данных. Интервальный тип данных (тип-диапазон). Арифметические процедуры и функции. Скалярные процедуры и функции. Функции преобразования типов. Специальные процедуры и функции.
Модуль 2. Структурированные типы данных		
2.1	Структурированный тип данных массивы.	Описание типа. Одномерные и двумерные массивы. Действия над массивами. Действия над элементами массива. Способы заполнения массивов. Многомерные массивы. Алгоритмы сортировки данных. Алгоритмы поиска.
2.2	Структурированные типы данных записи и множества.	Записи. Описание типа. Оператор With ... do. Записи с вариантами. Множества. Описание типа. Операции над множествами.
2.3	Подпрограммы	Процедуры и функции. Фактические и формальные параметры. Локальные и глобальные переменные. Передача параметров подпрограмме. Блочная структура. Рекурсия. Нисходящее и восходящее программирование
2.4	Расширение возможностей ввода-вывода	Работа с файлами. Чтение из файла. Открытие файла. Запись в файл. Текстовые файлы. Типизированные файлы.
Модуль 3. Стандартные и собственные модули		
3.1	Способы конструирования программ. Стандартные модули.	Программный модуль. Структура модуля. Модуль CRT. Установка текстовых режимов. Очистка экрана и управление строками на экране. Вывод на экран. Управление цветом и фоном. Управление яркостью. Управление курсором. Текстовые окна. Управление клавиатурой. Управление звуком. Модуль System. Распределение памяти. Работа с HEAP-областью. Модуль DOS. Назначение. Обслуживание прерываний. Системное время и дата. Сервисные функции DOS. Обработка ошибок модуля DOS. Состояние внешнего носителя. Представление файлов в DOS. Чтение оглавления диска. Командный процессор. Модули PRINTER, TURBO3, GRAPH3. Модуль OVERLAY.
3.2	Графика Турбо-Паскаля.	Инициализация графического режима. Драйверы. Система координат. Основные процедуры модуля GRAPH для рисования графических примитивов. Вывод текста. Мультипликация.
3.3	Динамические структуры данных	Динамическое распределение памяти. Адреса и указатели. Стек. Списки: основные виды и способы реализации. Очереди. Деревья.
Модуль 4. Основы работы в среде Дельфи		
4.1	Объектно - ориентированное программирование	Понятие объекта. Понятие класса. Основные принципы объектно-ориентированного программирования: инкапсуляция, полиморфизм, наследование. Поля. Методы. Типы методов. События и их обработка.

4.2	Среда визуальной разработки Delphi. Основные компоненты Standard. Отладка программы.	Графический интерфейс системы Delphi : главное меню, панели инструментов, палитра компонентов, форма, окно редактора кода, инспектор объектов. Создание проекта. Структура проекта Delphi. Сохранение проекта. Открытие проекта. Добавление новой формы. Структура программы. Модульный принцип компоновки программы. Типы данных в Delphi. Причины ошибок. Синтаксические ошибки. Логические ошибки. Выполнение по шагам. Просмотр значений. Просмотр и изменение значений. Просмотр и анализ кода. Причины ошибок. Компоненты вкладки Standard : Frames (Фрейм), Edit (текстовое поле), Mainmenu (меню), Popupmenu (контекстное меню), Label (подпись), Memo (текстовая область), Button (кнопка), Checkbox (флажок), Radiobutton (переключатель), Radiogroup (группа переключателей), Listbox (Список), Scrollbar (Полоса прокрутки), Combobox (поле со списком), Groupbox (группа элементов), Panel (рамка). Их свойства, методы, события.
-----	--	--

Таблица 3

V.2. Тематический план изучения дисциплины

№	Раздел программы	Виды учебной работы и их трудоемкость											Формируемые компетенции	
		Лекции из них Практическая подготовка				Лабораторные Занятия из них практическая подготовка				Промежуточный контроль		Самостоятельная работа		
		Очно	Заочно	Очно	Заочно	Очно	Заочно	Очно	Заочно	Очно	Заочно			
Модуль 1. Основные элементы языка программирования Турбо-Паскаль														
1.1	Основные этапы решения задач на ЭВМ	1	1			1	1			2	1	2	10	ПКО-1, ПКО-7
1.2	Основные понятия языка программирования Турбо-Паскаль	2		1	1	1	1			2		2	10	ПКО-1, ПКО-7
1.3	Управляющие конструкции	1	1			2		1	1	2	1	4	10	ПКО-1, ПКО-7
1.4	Строковый тип данных. Типы данных, определяемые программистом	1	1			2				2		4	19	ПКО-1, ПКО-7
<i>Промежуточный контроль</i>										6	2			
Модуль 2. Структурированные типы данных														
2.1	Структурированный тип данных массивы.	1	1	1	1	2				2		4	10	ПКО-1, ПКО-7
2.2	Структурированные типы данных записи и множества.	2				1	1			2	1	4	12	ПКО-1, ПКО-7
2.3	Подпрограммы	1	1			1	1	1	1	2		5	12	ПКО-1, ПКО-7

2.4	Расширение возможностей ввода-вывода	1	1			1	1			2	1	4	12	ПКО-1, ПКО-7
													9	
<i>Промежуточный контроль</i>										7	2			
Модуль 3. Стандартные и собственные модули														
3.1	Способы конструирования программ	2	2			1	1	1	1	2		4	12	ПКО-1, ПКО-7
3.2	Графика Турбо-Паскаля.	2		2		2	2	1	1	2		4	12	ПКО-1, ПКО-7
3.3	Динамические структуры данных	3	1			3	1			2	1	4	12	ПКО-1, ПКО-7
													9	
<i>Промежуточный контроль</i>										7	2			
4.1	Объектно - ориентированное программирование	2	2			1	1			1		6		ПКО-1, ПКО-7
4.2	Среда визуальной разработки Delphi. Основные компоненты вкладки Standard. Отладка программы.	4		1	1	1	1			1	1	6		ПКО-1, ПКО-7
Промеж контроль										27	6		9	
Итого		32		8		32		8		27	6	53	122	

Таблица 4

V.3 Темы лабораторных занятий

№ п/п	Раздел программы	Тема лабораторного занятия	Цель лабораторного занятия	Учебно-методические материалы	Результаты
Модуль 1. Основные элементы языка программирования Турбо-Паскаль					
1.1	Основные этапы решения задач	Основные этапы решения задач. Построение блок-схем алгоритмов	Формирование знаний и умений по работе с блок-схемами алго-	8, 9, 1, 7	Знает: основные этапы решения задач на ЭВМ.

	на ЭВМ		ритмов		Умеет: выделять основные этапы для решения задачи на ЭВМ, составлять алгоритм и строить блок-схему алгоритма.
1.2	Основные понятия языка программирования Турбо-Паскаль	Изучение меню интегрированной среды (ИС) программирования Паскаль. Программирование алгоритмов линейной структуры	Формирование знаний и умений по работе с интегрированной средой Турбо-Паскаль, по составлению программ с использованием операторов ввода – вывода данных, присваивания	8, 9, 1, 7, 13, 3, 5	Знает: команды меню. Умеет: составлять программы с использованием простых типов данных, операторов ввода-вывода и присваивания
1.3	Управляющие конструкции	Программирование на языке Паскаль с использованием условного оператора, оператора Case и операторов цикла	Формирование знаний и умений по работе с условным оператором, оператором Case и операторами цикла	8, 9, 1, 7, 11, 3, 5, 21, 22, 24	Знает: описание и принципы работы перечисленных операторов. Умеет: составлять программы с их использованием
1.4	Строковый тип данных. Типы данных, определяемые программистом	Программы на языке Паскаль с использованием стандартных строковых процедур и функций	Формирование знаний и умений по работе со строками, приобретение навыков работы со стандартными строковыми процедурами и функциями	8, 9, 1, 7, 11, 3, 5, 21, 22, 24	Знает: описание строкового типа, строковые процедуры и функции. Умеет: составлять программы с их использованием.
Модуль 2. Структурированные типы данных					
2.1	Структурированный тип данных массивы.	Работа с одномерными и двумерными массивами	Формирование знаний и умений по работе с массивами	8, 9, 1, 7, 11, 3, 5, 21, 22, 24	Знает: понятие массива одномерного, ..., n-мерного и их описание на Турбо-паскаль. Умеет: выполнять действия над массивами .
2.2	Структурированные типы данных записи и множества.	Составление программ на языке Паскаль с использованием записей и множеств	Формирование знаний и умений по работе с записями и множествами	8, 9, 1, 7, 11, 3, 5, 21, 22, 24	Знает: понятие записи, множества и их описание. Умеет: выполнять операции над множествами, использовать оператор With, составлять программы с их применением
2.3	Подпрограммы	Составление программ на Паскале с использованием подпрограмм, определенных пользователем	Формирование знаний и умений по работе с подпрограммами	8, 9, 1, 7, 11, 3, 5, 21, 22, 24	Знает: понятия процедуры и функции, их описания. Умеет: писать процедуры и функции и использовать их в основной программе.
2.4	Расширение возможностей ввода-вывода	Программирование на языке Паскаль с использованием файловых переменных	Формирование знаний и умений по работе с файлами	8, 9, 1, 7, 11, 3, 5, 21, 22, 24	Знает: понятие файла, типы файлов. Умеет: выполнять действия над файлами в Турбо-Паскаль.
Модуль 3. Стандартные и собственные модули					
3.1	Способы конструирования программ	Программирование на языке Паскаль с использованием разработанного собственного модуля	Формирование знаний и умений по работе с модулями, приобретение навыков создания собственных модулей	8, 9, 1, 7, 11, 3, 5, 21, 22, 24	Знает: стандартные модули, структуру модуля. Умеет: составлять программы с использованием процедур и функций стандартных модулей; создавать собственные модули
		Изучение наиболее часто употребляемых универсальных процедур и оформление их в виде личной библиотеки программиста	Приобретение навыков по созданию собственной библиотеки	8, 9, 1, 7, 11, 3, 5, 21, 22, 24	Умеет: создавать собственную библиотеку процедур и функций
3.2	Графика Турбо-Паскаля.	Графические возможности Паскаля	Формирование знаний и умений по работе с графикой	8, 9, 1, 7, 11, 3, 5, 21, 22, 24	Знает: процедуры и функции модуля Graph. Умеет: составлять программы, содержащие графические объекты
		Построение движущихся объектов на языке Паскаль	Приобретение навыков создания движущихся объектов	8, 9, 1, 7, 11, 3, 5, 21, 22, 24	Знает: способы организации анимации на Турбо-паскаль. Умеет: создавать движущиеся объекты на Турбо-Паскаль
3.3	Динамические структуры данных	Использование динамических структур данных при составлении программ на	Формирование знаний и умений по работе с динамической памятью	8, 9, 7, 11, 3, 5, 21, 22,	Знает: указатели и адреса.

	ных	языке Паскаль.	тью, приобретение навыков работы с динамическими переменными, указателями.	24	Умеет: работать со стеком, очередью
		Разработка программы создания связанного списка	Приобретение навыков работы по созданию связанного списка	8, 9, 7, 11, 3, 5, 21, 22, 24	Умеет: создавать связанные списки на Турбо-Паскаль
Модуль 4. Основы работы в среде Дельфи					
4.1	Объектно - ориентированное программирование	Общая организация работы в Дельфи.	Приобретение навыков работы с проектом Дельфи	8, 9, 6, 2, 6, 20	Знает: принципы работы в Дельфи. Умеет: работать с проектом в Дельфи
4.2	Среда визуальной разработки Delphi. Основные компоненты вкладки Standard. Отладка программы.	Работа с компонентами вкладки Standard.	Изучение свойств, методов и событий основных компонентов вкладки Standard	8, 9, 2, 6, 20	Знает: свойства, методы, события некоторых компонентов Дельфи. Умеет: создавать проекты с их использованием.
		Разработка консольного приложения на Дельфи	Приобретение навыков создания консольных приложений на Дельфи	8, 9, 2, 6, 20	Умеет создавать консольное приложение на Дельфи

V.4. Самостоятельная работа студентов

Основные направления самостоятельной работы:

- изучение литературы и лекционного материала;
- подготовка к лабораторным работам, их завершение, оформление отчетов и их защита;
- разработка учебных проектов;
- разработка программных продуктов;
- выполнение индивидуальных заданий;
- выполнение рефератов;
- курсовая работа;
- подготовка к промежуточному тестированию.

V.4.1. Темы курсовых работ

1. Алгоритмы поиска в курсе информатики средней школы.
2. Формирование навыков использования циклов при решении задач целочисленной арифметики.
3. Занимательные задачи по программированию на Паскале.
4. Методика начального обучения и контроля знаний по теме «Алгоритмы целочисленной арифметики».
5. Развитие представлений учащихся о программном обеспечении ЭВМ.
6. Типы циклических задач на языке программирования Паскаль.
7. Роль языковой подготовки в формировании учителя информатики.
8. Работа с символьными переменными на Паскале.
9. Использование занимательных задач на уроках информатики.
10. Методика начального обучения и контроля знаний по теме «Запись и объект на языке Turbo Pascal 7.0».
11. Современные системы и среды для обучения основам алгоритмизации и программирования.
12. Методика начального обучения и контроля знаний по теме «Оператор цикла на языке Turbo Pascal 7.0».
13. Методика изложения и контроля знаний по теме «Массивы и строки на языке Turbo Pascal 7.0».
14. Методика изложения и контроля знаний по теме «Числовые типы на языке Turbo Pascal 7.0».
15. Методика начального обучения и контроля знаний по теме «Основные операторы языка Turbo Pascal 7.0».

16. Методика начального обучения и контроля знаний по теме «Условный оператор и оператор выбора на языке Turbo Pascal 7.0».

V.4.2. Творческие задания

1. Разработка программы нахождения определенного интеграла в среде Турбо Паскаль 7.0.
2. Разработка элективного курса «Программируем на Паскале».
3. Регистрация участников конференции (Паскаль - данные хранятся в файле).
4. Разработка проекта «Решение квадратного уравнения».
5. Разработка проекта «Диалог».
6. Разработка проекта «Словарь».
7. Разработка проекта «Цвета в формате RGB».
8. Разработка проекта «Анкета студента».
9. Проектирование 3-х мерной фигуры и вращение ее вокруг оси.

V.4.3. Вопросы для самостоятельного изучения

1. Главное меню интегрированной среды Turbo Pascal.
2. Директивы компилятора и управляющие символы.
3. Правила пунктуации.
4. Принципы тестирования программ. Уровни сложности тестирования. Этапы тестирования.
5. Автоматизированные способы тестирования. Основы доказательства правильности.
6. Встроенные процедуры и функции.
7. Вложенные циклы
8. Способы завершения цикла
9. Способы завершения работы программы
10. Записи с вариантами.
11. Нисходящее и восходящее программирование
12. Модуль System.
13. Распределение памяти.
14. Работа с HEAP-областью.
15. Модуль DOS. Назначение. Обслуживание прерываний.
16. Системное время и дата. Процедуры работы с датой и временем.
17. Сервисные функции DOS. Обработка ошибок модуля DOS.
18. Состояние внешнего носителя. Процедуры и функции проверки состояния диска.
Чтение оглавления диска.
19. Процедуры и функции работы с каталогами.
20. Представление файлов в DOS. Процедуры и функции работы с файлами. Командный процессор.
21. Модуль PRINTER
22. Модуль TURBO3
23. Модуль GRAPH3
24. Модуль OVERLAY.
25. Рекурсивные алгоритмы.
26. Методы и искусство программирования. Проектирование программ.
27. Методы проектирования, ориентированные на обработку.
28. Методы проектирования, ориентированные на данные
29. Модуль WINDOS.
30. Функции управления операционной средой.
31. Процедуры обслуживания прерываний.
32. Элементы теории графов. Метод сортировки выбором с помощью дерева.
33. Основы работы с базами данных на Delphi.
34. Работа с клиент-серверными СУБД.
35. Взаимодействие приложений
36. Многоуровневые приложения
37. Программирование для Интернета

38. Создание собственных компонентов

V.4.4. Практические задания

1. Заданы две строки s и x . Длина первой строки n , длина второй – m , причем $0 \leq m \leq n$. Требуется ответить на вопрос, является ли строка x подстрокой строки s , при этом поиск подстроки должен обнаруживать первое вхождение x в s
2. Багаж пассажира характеризуется количеством вещей и общим весом вещей. Дан массив, содержащий сведения о багаже пассажиров. Найти число пассажиров, имеющих более двух вещей и число пассажиров, количество вещей которых превосходит среднее число вещей
3. *Задача о шахматном коне*. Существуют способы обойти шахматным конем шахматную доску, побывав на каждом поле по одному разу. Составить программу нахождения всех возможных способов обхода доски
4. *Ханойские башни*. Имеются три стержня A, B, C . На стержень A нанизано n дисков радиуса 1, 2, ..., n таким образом, что диск радиуса i является i -м сверху. Требуется переместить все диски на стержень B , сохраняя их порядок расположения (диск с большим радиусом находится ниже). За один раз можно перемещать только один диск с любого стержня на любой другой стержень. При этом должно выполняться следующее условие: на каждом стержне, ни в какой момент времени, никакой диск не может находиться выше диска с меньшим радиусом.
5. *Delphi*. Написать программу, которая выводит на экран гистограмму, например, результатов контрольной работы
6. *Delphi*. Напишите программу, которая добавляет в базу данных «Погода», реализованную в виде текстового файла, информацию о дневной температуре. Для ввода даты используйте компонент *Monthcalendar*. Если файл данных отсутствует, то программа должна его создать. Рекомендуемый вид формы программы приведен на рис.
7. *Delphi*. Написать программу, используя которую можно оценить способность испытуемого запоминать числа. Программа должна выводить числа, а испытуемый – вводить эти числа с клавиатуры. Время, в течение которого игрок будет видеть число, ограничьте, одной секундой. По окончании теста программа должна вывести результат: количество показанных чисел и количество чисел, который испытуемый запомнил и ввел правильно
8. Построение графиков функции на Дельфи.

Таблица 5

Задания для самостоятельного выполнения

№ п/п	Раздел программы	Задания	Формы отчетности и контроля
<i>Модуль 1. Основные элементы языка программирования Турбо-Паскаль</i>			
1.1	Основные этапы решения задач на ЭВМ	1. Изучить литературу 11, 3, 7,4 2. Изучить вопрос 1 из раздела V.4.3 3. Изучить методические указания к лаб. раб. 1.1 4. Оформить отчет по л/р 1.1	Опрос теоретического материала, отчет по л/р, защита выполненных заданий
1.2	Основные понятия языка программирования Турбо-Паскаль	1. Изучить литературу 11, 3, 7,4 2. Изучить вопрос 2-5 из раздела V.4.3 3. Изучить методические указания к лаб. раб. 1.2 4. Оформить отчет по л/р №1.2	Опрос теоретического материала, отчет по л/р, защита выполненных заданий
1.3	Управляющие конструкции	1. Изучить литературу 11, 3, 7,4 2. Изучить вопрос 6 из раздела V.4.3 3. Изучить методические указания к лаб. раб. 1.3 4. Оформить отчет по л/р 1.3 5. Выполнить творческое задание 1 из раздела V.4.2.	Опрос теоретического материала, отчет по л/р, защита выполненных заданий
1.4	Строковый тип данных. Типы данных, определяемые программистом	1. Изучить литературу 1, 11, 3, 7, 5 2. Выполнить практическое задание 1 из раздела V.4.4. 3. Изучить методические указания к лаб. раб. 1.4 4. Оформить отчет по л/р 1.4. 5. Подготовка к промежуточной аттестации	Опрос теоретического материала, отчет по л/р, защита выполненных заданий, тестирование
<i>Модуль 2. Структурированные типы данных</i>			

2.1	Структурированный тип данных массивы	1. Изучить литературу 1, 11, 3, 7, 5 2. Выполнить практическое задание 2 из раздела V.4.4. 3. Изучить методические указания к лаб. раб. 2.1. 4. Оформить отчёт по л/р 2.1	Опрос теоретического материала, отчёт по л/р, защита выполненных заданий
2.2	Структурированные типы данных записи и множества	1. Изучить литературу 1, 11, 3, 7, 5 2. Изучить вопрос 7 из раздела V.4.3 3. Выполнить практическое задание 3 из раздела V.4.4. 4. Изучить методические указания к лаб. раб. 2.2 5. Оформить отчёт по л/р 2.2	Опрос теоретического материала, отчёт по л/р, защита выполненных заданий
2.3	Подпрограммы	1. Изучить литературу 1, 11, 3, 7, 5, 9 2. Изучить вопросы 8, 22 из раздела V.4.3 3. Изучить методические указания к лаб. раб. 2.3. 4. Оформить отчёт по л/р 2.3. 5. Выполнить практическое задание 4 из раздела V.4.4.	Опрос теоретического материала, отчёт по л/р, защита выполненных заданий, программа ЭВМ
2.4	Расширение возможностей ввода-вывода	1. Изучить литературу 1, 11, 3, 7, 5, 9 2. Выполнить творческое задание 2 из раздела V.4.2 3. Изучить методические указания к лаб. раб. 2.4 4. Оформить отчёт по л/р 2.4 5. Подготовка к промежуточной аттестации	Опрос теоретического материала, отчёт по л/р, защита выполненных заданий, программа ЭВМ, тестирование
Модуль 3. Стандартные и собственные модули			
3.1	Способы конструирования программ	1. Изучить литературу 11, 3, 7, 5 2. Изучить вопрос 9-17 из раздела V.4.3 3. Изучить методические указания к лаб. раб. 3.1 4. Оформить отчёт по л/р 3.1 5. Выбор темы курсовой работы и анализ литературы	Опрос теоретического материала, отчёт по л/р, собеседование
3.2	Графика Турбо-Паскаля	1. Изучить литературу 11, 3, 7, 5, 9 2. Изучить вопросы 18-21, 23, 24 из раздела V.4.3 3. Изучить методические указания к лаб. раб. 3.2 4. Оформить отчёт по л/р 3.2. 5. Выполнить творческое задание 3 из раздела V.4.2	Опрос теоретического материала, отчёт по л/р, защита выполненных заданий, программа ЭВМ
3.3	Динамические структуры данных	1. Изучить литературу 11, 3, 7, 5 2. Изучить вопрос 25-29 из раздела V.4.3 3. Изучить методические указания к лаб. раб. 3.3 4. Оформить отчёт по л/р 3.3 5. Разработка 1 и 2 разделов курсовой работы 6. Подготовка к промежуточной аттестации	Опрос теоретического материала, отчёт по л/р, собеседование, тестирование
Модуль 4. Основы работы в среде Дельфи			
4.1	Объектно - ориентированное программирование	1. Изучить литературу 2, 6, 19, 25 2. Выполнить творческое задание 4, 5 из раздела V.4.3 3. Изучить методические указания к лаб. раб. 4.1 4. Оформить отчёт по л/р 4.1	Опрос теоретического материала, отчёт по л/р, защита выполненных заданий
4.2	Среда визуальной разработки Delphi. Основные компоненты вкладки Standard. Отладка программы	1. Изучить литературу 2, 6, 19, 23 2. Изучить методические указания к лаб. раб. 4.2 3. Оформить отчёт по л/р 4.2. 4. Выполнить творческое задание 6 из раздела V.4.2. 5. Разработка практической части курсовой работы 6. Подготовка к промежуточной аттестации	Опрос теоретического материала, отчёт по л/р, защита выполненных заданий, проект Дельфи, собеседование, тестирование

VI. Образовательные технологии

- Лекционные (информационная лекция, мультимедийная лекция, проблемная лекция);
- лабораторные (углубление знаний, полученных на теоретических занятиях, решение задач, составление, тестирование и отладка программ);
- работа с литературой при подготовке к лабораторным работам
- самостоятельная работа студентов (самостоятельное изучение материала, подготовка рефератов, разработка проектов).

VII. Оценочные средства контроля текущей успеваемости и промежуточной аттестации студентов

тов

VII.1. Тестовые задания по модулю №1

1. Когда необходимо составлять блок-схему программы?

- 1) до начала составления самой программы;

- 2) после составления программы;
- 3) в процессе составления программы

2. Наиболее наглядной формой описания алгоритма является структурно-стилизированный метод

- 1) представление алгоритма в виде схемы;
- 2) язык программирования высокого уровня;
- 3) словесное описание алгоритма

3. Перевод программ с языка высокого уровня на язык более низкого уровня обеспечивает программа:

- 1) ассемблер;
- 2) паскаль;
- 3) компилятор;
- 4) фортран

4. Разработка алгоритма решения задачи – это

- 1) сведение задачи к математической модели, для которой известен метод решения;
- 2) выбор наилучшего метода из имеющихся;
- 3) точное описание данных, условий задачи и ее целого решения;
- 4) определение последовательности действий, ведущих к получению результатов

5. Языком высокого уровня является:

- 1) макроассемблер;
- 2) шестнадцатеричный язык;
- 3) фортран;
- 4) ассемблер

6. Алгоритм, в котором действия выполняются друг за другом, не повторяясь называется:

- 1) линейным;
- 2) циклическим;
- 3) разветвленным;
- 4) простым

7. Вещественный тип данных объявляется служебным словом:

- 1) INTEGER;
- 2) REAL;
- 3) LONGINT;
- 4) SHORTINT

8. Оператор цикла с постусловием:

- 1) Repeat... until
- 2) While...do
- 3) For ... to...do
- 4) For...downto...do

9. Логический тип данных объявляется служебным словом:

- 1) BOOLEAN;
- 2) LOGIC;
- 3) BYTE;
- 4) IF

10. Раздел переменных определяется служебным словом:

- 1) LABEL
- 2) TYPE
- 3) VAR
- 4) ARRAY

11. Раздел типов определяется служебным словом:

- 1) LABEL;
- 2) TYPE;
- 3) VAR;
- 4) BEGIN

12. Какие из приведенных типов данных относятся к целочисленному типу данных?

- 1) integer, real;
- 2) integer, word, longint;
- 3) comp, double
- 4) single, extended

13. Из приведенных описаний переменных неправильно объявлены переменные

- 1) VAR f,g,d,t:INTEGER;I,t:REAL;
- 2) var a,b:real;c:real;
- 3) var I,j,max,min: real;
- 4) var a,b,c,d:real; I,j,k:integer;

14. Какие из приведенных типов данных относятся к вещественному типу данных?

- 1) byte, real;
- 2) shortint;
- 3) word, double;
- 4) real, single, extended;

15. Для вычисления экспоненты применяется процедура:

- 1) ORD(X);
- 2) SQR(X);
- 3) TRUNC(X);
- 4) EXP(X);

16. Результатом выполнения фрагмента программы S:=-5;x:=0;repeat s:=s*(x+2);x:=x+1; until x<2;write(s); будет сообщение:

- 1) -10.0
- 2) -30.0
- 3) 0.0
- 4) -120.0

17. Для возведения в квадрат применяется функция:

- 1) LN(X);
- 2) LOG(X);
- 3) SQRT(X);
- 4) SQR(X)

18. Для вычисления квадратного корня применяется функция:

- 1) SQR(X);
- 2) SQRT(X);
- 3) ORD(X);
- 4) EXP(X)

19. Раздел констант объявляется служебным словом:

- 1) CONST;
- 2) KONST;
- 3) CONCT;
- 4) VAR

20. Имеется логическое выражение $(x*x + y*y) \leq 1$. Для какой области на плоскости это выражение будет иметь значение TRUE?

- 1) для круга единичного радиуса;
- 2) для точек, расположенных вне круга с единичным радиусом;
- 3) для точек плоскости, расположенных внутри квадрата с единичными сторонами;
- 4) для точек плоскости, расположенных вне квадрата с единичными сторонами

21. Укажите правильную запись вычисления функции в виде одного условного оператора:

- 1) if $0 < x < 2$ then $y := \cos(x)$ else $y := 1 - \sin(x)$;
- 2) if $(0 < x)$ and $(x < 2)$ then $y := \cos(2*x)$ else $y := 1 - \sin(3*x)$;
- 3) if $(0 < x)$ and $(x < 2)$ then $y := \cos(2x)$ else $y := 1 - \sin(3x)$;
- 4) if $(0 < x)$ or $(x < 2)$ then $y := \cos(x)$ else $y := 1 - \sin(x)$

22. Цикл с предусловием определяется служебным словом:

- 1) FOR;
- 2) WHILE;
- 3) REPEAT;

4) IF

23. Цикл с постусловием определяется служебным словом:

- 1) FOR;
- 2) WHILE;
- 3) CASE;
- 4) REPEAT

24. Если в цикле с параметром: for i:= A to B do S; значение B меньше, чем значение A, то ...

- 1) оператор S не выполняется ни разу;
- 2) оператор S выполняется один раз.
- 3) оператор S выполняется B-A раз.
- 4) оператор S выполняется B-A+1 раз.

25. Если в цикле с параметром: for i:= A downto B do S; значение B больше, чем значение A, то ...

- 1) оператор S не выполняется ни разу;
- 2) оператор S выполняется один раз
- 3) оператор S выполняется B-A раз.
- 4) оператор S выполняется B-A+1 раз.

26. В операторе цикла с предусловием: while B do A; при B = FALSE оператор A ...

- 1) выполняется один раз;
- 2) не выполняется ни разу;
- 3) выполняется до тех пор, пока B не равно TRUE.

27. Определить результат работы следующего фрагмента программы k:=6; for i:=1 to 5 do inc(k); write(k, ' ');

- 1) 7 8 9 10 11
- 2) 11
- 3) 6
- 4) 1 2 3 4 5

28. При выполнении последовательности операторов A:=4; B:=2; X:=A mod B значение переменной X равно:

- 1) выражение записано неверно
- 2) 8
- 3) операция невыполнима, так как деление на 0 невозможно
- 4) 2

29. В результате выполнения последовательности операторов x:=sqr(2); y:=sqrt(4)-abs(-2); write(x, ' ', y) на экране будет выведено:

- 1) 0 -1
- 2) 4 6
- 3) 4 0
- 4) 1 0

30. Определить результат работы следующего фрагмента программы k:=6; for i:=1 to 5 do begin inc(k); write(k, ' ') end;

- 1) 7 8 9 10 11
- 2) 11
- 3) 6
- 4) 1 2 3 4 5

31. Неверно записано описание переменных:

- 1) VAR A,B:real; I,J:integer;
- 2) VAR j,g,t:integer; i:real;
- 3) VAR i,max:integer; a,max:real;
- 4) VAR a,b:real; c,d:real;

32. При выполнении последовательности операторов a:=1.0; b:=3; x:=(a+b)/a*b-a; значение переменной x равно:

- 1) 2
- 2) 1
- 3) 0.3333
- 4) 11

33. Имеется описание заголовка процедуры: procedure proba(var a, b, c:integer); При использовании процедуры с таким заголовком значения фактических параметров передаются ...

- 1) по ссылке;
- 2) по значению.
- 3) как константа

- 4) как процедура
34. Имеется описание заголовка процедуры: procedure proba(a, b, c:integer); При использовании процедуры с таким заголовком значения фактических параметров передаются ...
- 1) по ссылке;
 - 2) по значению
 - 3) как константа
 - 4) как процедура
35. Какое выражение неверно?
- 1) Структура подпрограммы аналогична структуре основной программы
 - 2) Подпрограмма предназначена для выполнения какой-то последовательности действий
 - 3) В Паскале имеется две разновидности подпрограмм – процедуры и функции
 - 4) В подпрограмме нельзя использовать глобальные переменные
36. Длину строки можно определить с помощью процедуры:
- 1) LENGTH;
 - 2) ORD;
 - 3) CHR;
 - 4) RECORD;
37. Переменная строкового типа обозначается идентификатором
- 1) string;
 - 2) word;
 - 3) char;
 - 4) byte;
38. Строки программы в Турбо Паскале имеют максимальную длину:
- 1) 80 символов;
 - 2) 126 символов;
 - 3) 256 символов;
 - 4) 512 символов.
39. Какие данные могут выступать в качестве индекса?
- 1) вещественные типы данных
 - 2) константы и переменные порядковых типов
 - 3) операторы
 - 4) целые типы данных
40. Выберите правильную операцию отношения так, чтобы следующее условие принимало значение True: 'balkon' > 'balken'
- 1) >
 - 2) <
 - 3) =
 - 4) <=
41. Что появится на экране в результате выполнения следующих операторов: ws:='электрификация'; sw:='тр'; p:=pos(sw,ws); write(p:2);
- 1) 1;
 - 2) 3;
 - 3) и;
 - 4) 5
42. Выберите правильный результат действия программы на Паскале:
- ```
const n=2;
var k:integer; m,p:real;
begin p:=0;
For k:=1 to 6 do
M:=k/n;
p:=p+m
end.
```
- 1) p=10.5;
  - 2) p=20.5;
  - 3) p=1.5;
  - 4) p=10.
43. Какое значение получит c после выполнения фрагмента программы: ... c:=10; x:=3; While x>=0 do begin c:=c-x; x:=x-1; end; ...
- 1) 3
  - 2) 5
  - 3) 4
  - 4) 2

44. Какое значение t получит после выполнения фрагмента программы:

```
... r:real;
begin
r:=4.869; t:=trunc(r); ...
End.
```

- 1) t=4.7;                    2) t=2.4;                    3) t=5;                    4) t=4.

45. Что будет выведено на печать в результате выполнения следующего фрагмента программы:

```
... y:=5; For x:=1 To 3 Do
If x>y Then y:=y+x Else y:=y-x; y:=2+y;
Writeln(x,y);
```

- 1) 3.7                    2) 4.2                    3) 3.5                    4) 7.4

46. Что будет напечатано в результате выполнения следующего фрагмента программы:

```
... a:=10; i:=4;
Repeat a:=a-i; i:=i-1 Until i<=0;
writeln(a); ...
```

- 1) 2                    2) 1                    3) -1                    4) 0.

47. Определите значение переменной s после выполнения следующих операторов:

```
S:=0; i:=1;
Repeat s:=s+20 div i; i:=i+1
Until i<=3
```

- 1) 36                    2) 20                    3) 15                    4) 10

48. Какая из данных программ записана без ошибок:

- 1) var b:boolean; begin b:= 'a'; writeln('результат: ', b); end.
- 2) var b:boolean; begin b:=false; if not b then writeln('Привет!') end.
- 3) var b:boolean; begin b:= 'Hello'; writeln(b); end.
- 4) var b:boolean; c:real; begin c:=sqr(b); writeln('результат: ', c); end.

49. В операторе присваивания  $summa := \text{sqr}(x) + 3 * a$  переменными являются

- 1) a, x, summa
- 2) x, a
- 3) sqr, x, a
- 4) summa, sqr, x, a

50. Процедура INC(x,k)

- 1) преобразует десятичное число x в строку из k символов;
- 2) увеличивает значение переменной x на величину k;
- 3) уменьшает значение переменной x на величину k;
- 4) преобразует строку символов x в число, содержащее k десятичных знаков

## VII.2. Тестовые задания по модулю №2

1. Процедуры и функции – это:

- 1) операторы;
- 2) подпрограммы;
- 3) имена;
- 4) переменные.

2. Обращение к процедуре в программе имеет вид:

- 1) <имя процедуры>(<список глобальных параметров>);
- 2) <оператор процедуры>;
- 3) <имя процедуры>:тип значения;
- 4) <имя процедуры>(<Список фактических параметров>)

3. Каков будет результат выполнения программы:

```
type digits=set of 0..9;
var d1,d2,d3:digits;
begin
d1=[1,3,5];
d2:=[0,4,5];
d3:=d1*d2;
end.
```

- 1) d3=[0,1,3,4,5];                    2) d3=[0,1,3,4];                    3) d3=[5];                    4) d3=[1,3,5,0,4,5].

4. Каков будет результат выполнения программы:

```
type digits=set of 0..9;
var d1,d2,d3:digits;
begin
d1=[1,3,5];
d2:=[0,4,5];
d3:=d1+d2;
end.
```

- 1) d3=[0,1,3,4,5];      2) d3=[0,1,3,4];      3) d3=[5];      4) d3=[1,3,5,0,4,5].

5. Где правильно описана функция?

- 1) FUNCTION (a,b,c:real):integer;
- 2) FUNCTION Faktor (a:integer):integer;
- 3) FYNCTION Sum (a,b,c:real):integer;
- 4) FUNKTION A(a,b,c:real): integer

6. Глобальные переменные действуют:

- 1) во всех процедурах;    2) во всех функциях;    3) во всех модулях;    4) во всей программе.

7. Укажите правильное обращение к процедуре, объявленной как:

**Procedure** Tangens(f:real; var t:real);

- 1) tangens (3.14,t,k);
- 2) tangens(pi/2,t);
- 3) tangens(a,3.14);
- 4) tangen(pi/6,t);

8. Укажите правильный вызов процедуры Procedure LOG10(Argument:Real; var LOG:real);

- 1) LOG10(x,2.3);
- 2) LOG10(x,2\*y);
- 3) LOG10(1.5,y);
- 4) LOG10(x,y,z)

9. Где правильно описана процедура?

- 1) PROCEDURA (a:array [1..10] of integer; b:real);
- 2) PROSEDURA (var a:array [1..10] of integer; b:real);
- 3) PROSEDURA (a:array of integer; b:real);
- 4) PROCEDURE (a:array of integer; var b:real)

10. Какие виды подпрограмм есть в Паскале?

- 1) модуль и функция;
- 2) процедура и функция;
- 3) модуль и процедура;
- 4) процедура

11. Как называются имена, представленные в разделе описаний основной программы?

- 1) формальными;
- 2) параметрами-константами;
- 3) локальными;
- 4) глобальными

12. Каких параметров подпрограммы не существует?

- 1) параметров-переменных
- 2) параметров-констант
- 3) параметров-значений
- 4) параметров-модулей

13. Подпрограммы располагаются в разделе описания...?

- 1) переменных
- 2) типов
- 3) процедур и функций
- 4) меток

14. Каков будет результат выполнения программы:

```
var dl,d2,d3:set of char;
begin
d1=['a','b','c','d','m'];
d2=['b','f','d','c','e'];
d3:=dl-d2;
end.
```

- 1) d3=['b','c','d'];    2) d3=[];    3) d3=['a','m'];    4) d3=['a','b','c','d','e','f','m'].

15. Что будет выведено на печать в результате выполнения фрагмента программы:

```
... var c,d:integer;
Procedure z(x:integer; var y:integer);
begin x:=x+1; y:=y+3; end;
begin c:=3; d:=0; z(c,d);
writeln(d); end.
```

- 1) 3                    2) -2                    3) 4                    4) 1

16. Что будет выведено на экране после выполнения следующего фрагмента программы:

```
B:=0; a:=1200;
While a<>0 do
Begin
b:=b*10+ a mod 10;
a:=a div 10;
End; Write(b);
```

- 1) 0                    2) 12                    3) 2                    4) 21

17. Каков будет результат выполнения программы:

```
var s1,s2:string;
begin
s1:='информатика';
delete(s1,3,4);
writeln(s1); end.
```

- 1) инатика; 2) форма; 3) инфотика; 4) инфо.

18. Определите значение переменной s после выполнения следующих операторов:

```
S:=0; i:=1;
While i>1 do
Begin
S:=s+100 div I; dec(i) end;
```

- 1) 100                    2) 0                    3) 99                    4) 1

19. Чему равны значения переменных a и b после выполнения последовательности действий:

```
a:=31 mod 4 div 2; b:=(34 div 3)*(16 mod 5);
```

- 1) 0, 3                    2) 3, 0                    3) 3, 14                    4) 1, 11

20. Каков будет результат выполнения программы:

```
var r,t:real;
begin
r:=4.869; t:=trunc(r);
writeln(t);End.
```

- 1) 4.7;                    2) 2.4;                    3) 5;                    4) 4

21. Определите значение переменной s после выполнения следующих операторов:

```
S:=0; n:=5;
For i:=2 to n do s:=s+10 div i;
```

- 1) 10                    2) 9                    3) 12                    4) 8

22. Множество — это:

- 1) список элементов, заключенный в круглые скобки, вида: <имя поля>:<значение>;
- 2) неупорядоченная совокупность отличных друг от друга однотипных элементов;
- 3) совокупность с фиксированным числом однотипных элементов, отличных только индексами;
- 4) совокупность отличных друг от друга элементов разных типов.

23. Что будет напечатано в результате выполнения фрагмента программы:

```
... a:=1; i:=3;
```

```
Repeat a:=a+i; i:=i-1 until i<=0; writeln (a); ...
```

1) 8                      2) 5                      3) 6                      4) 7

24. Какие значения примут переменные *c* и *d* в результате выполнения программы:

```
program v;
```

```
var a, b, c, d: integer;
```

```
begin
```

```
a:=6;
```

```
b:=2*a+8;
```

```
if b>a then c:=b - a else d:=a - b;
```

```
writeln ('c=', c); writeln ('d=', d);
```

```
end.
```

1) c=14, d=0;    2) c=38, d=14;    3) c=14, d= - 14;    4) c= - 3, d=3.

25. Какое значение примет переменная *c* в результате выполнения программы:

```
program v;
```

```
var a, b, c: integer;
```

```
begin
```

```
a:=2;
```

```
b:=a*3+1;
```

```
if b>a+5 then a:=4;
```

```
if b
```

```
if b=a+5 then a:=b - a;
```

```
c:=a*b+a - b;
```

```
writeln ('c=', c);
```

```
end.
```

0) 33;    2) 25;    3) 15;    4) 41.

### VII.3. Тестовые задания по модулю №3

1. Для динамических переменных выделение и очистка памяти происходит:

- 1) на этапе трансляции;    2) на этапе компиляции;  
3) на этапе отладки;    4) в ходе выполнения программы.

2. Связь программы с принтером осуществляется процедурой:

1) assign(f,'aux');    2) assign (f,'usr');    3) assign (f, '1st');    4) assign (f, '№ порта принтера').

3. Значением указателя динамической переменной является:

- 1) адрес сегмента носителя информации, в котором будет храниться соответствующая динамическая величина;  
2) адрес ячейки памяти, начиная с которой будет храниться соответствующая динамическая величина;  
3) № кластера жесткого диска, в котором будет храниться соответствующая динамическая величина;  
4) значение динамической величины.

4. Файл – это:

- 1) база данных;    2) поименованный участок на внешних носителях памяти;  
3) список;    4) раздел на жестком диске.

5. Каков будет результат выполнения программы:

```
type digits=set of 0..9;
```

```
var d1,d2,d3:digits;
```

```
begin
```

```
d1=[1,3,5];
```

```
d2:=[0,4,5];
```

```
d3:=d1-d2;
```

```
end.
```

1) d3=[0,1,3,4,5];    2) d3=[1,3];    3) d3=[5];    4) d3=[1,3,5,0,4,5].

6. Рекурсия – это:

- 1) повторение выполнения функции или процедуры внутри себя;  
2) оператор;    3) цикл;    4) совокупность однотипных элементов.

7. Массив — это:

- 1) запись множества переменных разного типа;
- 2) неупорядоченная совокупность отличных друг от друга однотипных элементов;
- 3) последовательность, состоящая из фиксированного числа однотипных элементов;
- 4) тип одномерных величин.

8. Что будет выведено на печать в результате выполнения фрагмента программы:

```
. var a,b:integer;
Procedure k(x:integer; var y:integer);
begin x:=x+2; y:=y+2; writeln(x); end;
begin a:=0; b:=0; k(a,b); . . .
1)2 2) 0 3) 1 4) 3
```

9. Какие файловые типы данных поддерживает Турбо Паскаль?

1. текстовые
2. графические
3. системные
4. архивные

10. Какое из ниже перечисленных выражений не может быть выражением-константой?

- 1) Ord('Z') - Ord('A') + 1;
- 2) ['0'..'9'];
- 3) Blink + Red \* 16 + func(White);
- 4) 80 - Length(Error) div 2;
- 5) (Max - Min) div 2

11. Какое из следующих описаний переменных является ошибочным?

Operator: (plus, minus, times);  
Matrix: array[1..10,1..10] of Real;  
Done,Error: boolean;  
I,J,K: integer;

12. Какой оператор не относится к группе операторов ввода-вывода языка Паскаль ?

```
Read(A1,A2,...AK);
WriteLn(A1,A2,...AK);
PrintLn;
ReadLn;
```

13. В каком из условных операторов допущена синтаксическая ошибка?

- 1) if B = 0 then Writeln('Деление на нуль невозможно.');
- 2) if a > b then max := a else max := b;
- 3) if (a>b) and (b>0) then c:=a+b;
- 4) if a < b then min := a; else min := b;

14. Какие файловые типы данных поддерживает Турбо Паскаль?

- 1) Текстовые
- 2) Графические
- 3) Компонентные
- 4) Системные
- 5) Бестиповые

15. Какие из перечисленных операторов являются операторами ввода данных?

- 1) And, or.
- 2) Read, readln.
- 3) If, then.
- 4) Write, writeln.

16. Какие значения примут переменные c и d в результате выполнения программы:

```
program v;
var a, b, c, d: integer;
begin
a:=7;
```

```

b:=a - 3;
if b>a then c:=b - a else d:=a - b;
writeln ('c=', c); writeln ('d=', d);
end.

```

- 1) c=4, d=0.
- 2) c= - 3, d=4.
- 3) c=4, d= - 4.
- 4) c=0, d=3.

17. Что такое массив?

- 1) Ограниченная упорядоченная совокупность однотипных величин
- 2) Ограниченная совокупность различных элементов
- 3) Совокупность ограниченного числа логически связанных компонент, принадлежащих к разным типам

18. Значения переменных A и B после выполнения фрагмента программы

```
a:=1; b:=10; a:=a+b; b:=a - b; a:=a - b;
```

- 1) останутся прежними;
- 2) поменяются местами.
- 3) станут равными соответственно сумме и разности прежних своих значений.

19. Какого раздела не существует в программе, написанной на языке Паскаль?

- 1) заголовка;
- 2) примечания;
- 3) описаний;
- 4) операторов

20. Какая из процедур не работает с динамической областью памяти в Турбо Паскале?

- 1) New
- 2) Dispose
- 3) Allocmem
- 4) Freemem
- 5) release

21. Создание и использование связанного списка. Пусть требуется создать связанный список из записей, содержащих сведения об автомобилях, а также реализовать операции со связанным списком: запись первым в список, удаление первого объекта из списка, просмотр всего списка, удаление объекта, следующего за указанным.

#### VII.4. Тестовые задания по модулю №4

1. Что произойдет при вызове процедуры test?

```

var
 s1 : set of 1..5 = [1, 2, 3];
 s2 : set of 3..7 = [3, 4, 5];

procedure Test;
begin
 s1 := s1 + s2;
 if (5 in s2) then
 ShowMessage ('Да')
 else
 ShowMessage ('Нет')
end;

```

- 1) Появится окно диалога с сообщением «Да»
- 2) Вызов будет невозможен, ошибка компиляции
- 3) Появится окно диалога с сообщением «Нет»

2. Что произойдет при вызове процедуры Test?

```

procedure Test;
var
 MyColors: (Red,Blue,Green);
begin
 MyColors := Succ(Red);
end;

```

- 1) Вызов будет невозможен, ошибка компиляции
- 2) Переменная MyColors примет значение Blue
- 3) Переменная MyColors примет набор значений Red
- 4) Переменная MyColors примет значение Green

3. Какое значение вернет функция Test(0)?

```

function Test(a : integer): integer;
begin
 result := 0;
 if ((a + 12) > 0) OR (a > (result - 12))
 then
 a := -1;
 result := 1
end;

```

- 1) -100
- 2) 1
- 3) -1

4. Что произойдет при вызове процедуры Test?

```

var
 arr : array[1..5] of integer = (2,4,1,5,3);
 i : integer = 0;

```

```

procedure Test;
begin
 i := i + 1;
 ShowMessage(IntToStr(arr[i + i]))
end;

```

- 1) Появится окно диалога с сообщением 5
- 2) Появится окно диалога с сообщением 3
- 3) Появится окно диалога с сообщением 2
- 4) Появится окно диалога с сообщением 4
- 5) Появится окно диалога с сообщением 1

5. Что произойдет при вызове метода MyTest?

```

type
 TForm1 = class(TForm)
 Edit1: TEdit;
 procedure MyTest;
 end;
var
 Form1: TForm1;
implementation
 {$R *.dfm}
procedure TForm1.MyTest;
begin
 Edit1.Text := IntToStr(Edit1.Height);
end;

```

- 1) В Edit1 отобразится ширина Edit1
- 2) Вызов будет невозможен, у форм (Form) нет метода MyTest
- 3) В Edit1 отобразится высота Edit1
- 4) В Edit1 отобразится высота Form1

6. Зачем используется ключевое слово *private*?

```
unit Unit2;
interface
type
 MyClass = class
 private
 Fx: integer;
 Fy: integer;
 public
 function GetX: integer;
 function GetY: integer;
 end;
```

- 1) Чтобы сделать переменные Fx, Fy доступными только для чтения
- 2) Чтобы сделать переменные Fx, Fy доступными только в функциях GetX, GetY
- 3) Чтобы сделать переменные Fx, Fy доступными только в модуле Unit2

7. Какое значение вернет функция *Sign(-100)*?

```
function Sign(a : integer): integer;
begin
 result := 0;
 if (a > 0)
 then
 a := -1;
 result := 1
end;
```

- 1) -100
- 2) -1
- 3) 1

8. Как создать элемент графического интерфейса, содержащий текст, который нельзя изменять, но можно взять в буфер обмена

- 1) Использовать компонент Label, установить свойство Enabled в false
- 2) Использовать компонент Edit, установить свойство ReadOnly в true
- 3) Использовать компонент Edit, установить свойство Enabled в false
- 4) Использовать компонент Label, установить свойство Enabled в true

9. Что произойдет при вызове процедуры *Test*?

```
procedure Test;
var
 MyColor: (Red, Blue, Green);
begin
 MyColor := Red;
 MyColor := Black;
end;
```

- 1) Вызов будет невозможен, ошибка компиляции
- 2) Переменная MyColor примет значение Black
- 3) Переменная MyColor примет значение Red

10. Что произойдет при вызове *ShowMessage(Test(10))*?

```

function Test(a: string): string; overload;
begin
 result := '1';
end;
function Test(a: integer): string; overload;
begin
 result := '2';
end;

```

- 1) Появится окно диалога с сообщением 1
- 2) Появится окно диалога с сообщением 2
- 3) Вызов будет невозможен, ошибка компиляции
- 4) Появится окно диалога с сообщением 10

11. Что произойдет при вызове процедуры Test?

```

procedure Test;
var
 a, b: integer;
begin
 a := 1;
 b := 2;
 ShowMessage('2' + '1');
end;

```

- 1) Появится окно диалога с сообщением 21
- 2) Вызов будет невозможен, ошибка компиляции
- 3) Появится окно диалога с сообщением 3
- 4) Появится окно диалога с сообщением Несоответствие типов

12. Для кнопки Button1 определен обработчик. Что произойдет при нажатии на кнопку?

```

procedure TForm1.Button1Click(Sender: TObject);
begin
 Button1.Left := Button1.Left + 2
end;

```

- 1) Кнопка переместится на 2 миллиметра вправо
- 2) Кнопка переместится на 2 пикселя вправо
- 3) Кнопка переместится на 2 миллиметра влево
- 4) Кнопка переместится на 2 пикселя влево

13. Что произойдет при вызове процедуры Test?

```

procedure Test;
var
 a, b: integer;
begin
 a := 1;
 b := 2;
 ShowMessage(IntToStr(a) + IntToStr(a));
end;

```

- 1) Появится окно диалога с сообщением 3
- 2) Появится окно диалога с сообщением 11
- 3) Появится окно диалога с сообщением 12
- 4) Вызов будет невозможен, ошибка компиляции

14. Что произойдет при вызове процедуры Test?

```

var
 arr : array[1..5] of integer = (2,4,1,5,3);
 i : integer = 1;

```

```

procedure Test;
begin
 ShowMessage(IntToStr(arr[2] + arr[3]));
end;

```

- 1) Появится окно диалога с сообщением 3
- 2) Появится окно диалога с сообщением 1
- 3) Появится окно диалога с сообщением 2
- 4) Появится окно диалога с сообщением 4
- 5) Появится окно диалога с сообщением 5

15. Для кнопки *Button1* определен обработчик. Что произойдет при нажатии на кнопку?

```

procedure TForm1.Button1Click(Sender: TObject);
begin
 Button1.Top := Button1.Top + 2
end;

```

- 1) Кнопка переместится на 2 миллиметра вниз
- 2) Кнопка переместится на 2 пикселя вверх
- 3) Кнопка переместится на 2 пикселя вниз
- 4) Кнопка переместится на 2 миллиметра вверх

16. Создайте приложение «Угадай число», в котором компьютер будет загадывать случайное число в диапазоне от 0 до 10, а пользователь должен угадать это число. Ошибка угадывания числа должна сопровождаться сообщением типа «загаданное число больше», выводимом в отдельном окне. В случае угадывания пользователем загаданного компьютером числа следует вывести в отдельном окне сообщение «Вы угадали» и указать число попыток.

## VII.5. Тестовые задания по модулю №4

1. Что произойдет при вызове процедуры *Test*?

```

procedure Test;
var
 a, b: integer;
begin
 a := 1;
 b := 2;
 ShowMessage(a + b);
end;

```

- 1) Появится окно диалога с сообщением 1
- 2) Появится окно диалога с сообщением 2
- 3) Появится окно диалога с сообщением 4
- 4) Вызов будет невозможен, ошибка компиляции

2. Что произойдет при вызове процедуры *Test*?

```

procedure Test;
var
 Mark: 1..5;
begin
 Mark := 6;
 Mark := Mark + 6;
 ShowMessage(IntToStr(Mark));
end;

```

- 1) Появится окно диалога с сообщением 6
- 2) Появится окно диалога с сообщением 12
- 3) Появится окно диалога с сообщением 1
- 4) Вызов будет невозможен, ошибка компиляции

3. Что произойдет при вызове процедуры *Test*?

```

procedure Test;
var
 a, b: string;
begin
 a := '1';
 b := '2';
 ShowMessage(a + b)
end;

```

- 1) Появится окно диалога с сообщением 12
  - 2) Вызов будет невозможен, ошибка компиляции
  - 3) Появится окно диалога с сообщением 3
  - 4) Появится окно диалога с сообщением 2
4. Что произойдет при вызове процедуры Test?

```

procedure Test;
var
 a: string;
begin
 a := '2';
 ShowMessage(a + '2');
end;

```

- 1) Вызов будет невозможен, ошибка компиляции
- 2) Появится окно диалога с сообщением 4
- 3) Появится окно диалога с сообщением Несоответствие типов
- 4) Появится окно диалога с сообщением 22

5. Какая из строк вызовет ошибку компиляции

```

procedure Test;
var
 c1 : AnsiChar;
 c2 : WideChar;
 c3 : word;
 c4 : string;
begin
 c1 := 'Ы'; // строка 1
 c2 := 'Ы'; // строка 2
 c3 := 'Ы'; // строка 3
 c4 := 'Ы' // строка 4
end;

```

- 1) строка 3
- 2) строка 2
- 3) строка 4
- 4) строка 1

6. Для кнопки Button1 определен обработчик (на рисунке). Что произойдет при нажатии на кнопку?

```

procedure TForm1.Button1Click(Sender: TObject);
begin
 Button1.Left := Button1.Left + 2
end;

```

- 1) Кнопка переместится на 2 пикселя влево
- 2) Кнопка переместится на 2 пикселя вверх
- 3) Кнопка переместится на 2 пикселя вправо
- 4) Кнопка переместится на 2 пикселя вниз

7. Какое значение вернет функция sTest?

```

function sTest: integer;
var
 i, s : integer;
begin
 for i := 1 to 3 do
 begin
 s := 0;
 s := s + 1;
 end;
 result := s;
end;

```

- 1) Вызов будет невозможен, ошибка компиляции
- 2) 3
- 3) 1
- 4) 6

8. Что произойдет при вызове процедуры Test?

```

procedure Test;
var
 a, b: integer;
begin
 a := 1;
 b := 2;
 ShowMessage(IntToStr(a + b));
end;

```

- 1) Вызов будет невозможен, ошибка компиляции

- 2) Появится окно диалога с сообщением 3
- 3) Появится окно диалога с сообщением 12
- 4) Появится окно диалога с сообщением Несоответствие типов

9. Что произойдет при вызове процедуры Test?

```
var
 s1 : set of 1..5 = [];
 s2 : set of 3..7 = [];

procedure Test;
begin
 if (s1 = s2) then
 ShowMessage ('Да')
 else
 ShowMessage ('Нет');
end;
```

- 1) Появится окно диалога с сообщением «Нет»
- 2) Появится окно диалога с сообщением «Да»
- 3) Вызов будет невозможен, ошибка компиляции

10. Какая строка позволит осуществить сохранение в текстовом файле строк Мемо?

```
Memol.Lines.SaveToFile('F:\temp\doc.txt'); //Строка 1
Memol.SaveToFile('F:\temp\doc.txt'); //Строка 2
Memol.Lines.LoadFromFile('F:\temp\doc.txt'); //Строка 3
Memol.LoadFromFile('F:\temp\doc.txt'); //Строка 4
```

- 1) строка 4
- 2) строка 1
- 3) строка 2
- 4) строка 3

11. Что произойдет при вызове процедуры Test?

```
procedure Test;
var
 MyDay: (mon, sun, sat);
begin
 MyDay := mon;
 ShowMessage(MyDay);
end;
```

- 1) Появится окно диалога с сообщением Mon
- 2) Появится окно диалога с сообщением Sun
- 3) Вызов будет невозможен, ошибка компиляции
- 4) Появится окно диалога с сообщением Sat

12. Что произойдет при вызове процедуры Test?

```
var
 s1 : set of 1..5 = [1];
 s2 : set of 3..7 = [3];

procedure Test;
begin
 s1 := [4];
 s2 := [4];
 if (s1 = s2) then
 ShowMessage ('Да')
 else
 ShowMessage ('Нет');
end;
```

- 1) Вызов будет невозможен, ошибка компиляции
- 2) Появится окно диалога с сообщением «Да»
- 3) Появится окно диалога с сообщением «Нет»

13. Какое выражение возвращает горизонтальную позицию курсора в тексте Метод?

```
Mem1.CaretPos.X; //Строка 1
Mem1.CaretPos.Y; //Строка 2
Mem1.Top; //Строка 3
Mem1.Left; //Строка 4
```

- 1) Выражение 3
- 2) Выражение 4
- 3) Выражение 1
- 4) Выражение 2

14. Что произойдет при вызове процедуры Test?

```
var
s1 : set of 1..5 = [];
s2 : set of 3..7 = [3];
```

```
procedure Test;
begin
s1 := s1 - [3];
s2 := s2 - [3];
if (s1 = s2) then
 ShowMessage('Да')
else
 ShowMessage('Нет')
end;
```

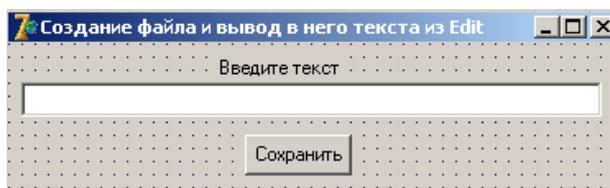
- 1) Появится окно диалога с сообщением «Да»
- 2) Появится окно диалога с сообщением «Нет»
- 3) Вызов будет невозможен, ошибка компиляции

15. Какое значение вернет функция Test(-100)?

```
function Test(a : integer): integer;
begin
result := 0;
a := 1;
if (a > 0)
 then
 result := 1
 else
 a := -1;
 result := -1
end;
```

- 1) -1
- 2) 1
- 3) -100

16. Создайте приложение, которое создает текстовый файл text1.txt и записывает в него текст, введенный пользователем в окно Edit, после чего закрывает файл.



## VII.6. Методика балльно-рейтингового оценивания успеваемости студентов

Контроль и оценка учебных достижений студентов по дисциплине «Языки и системы программирования» проводится в балльно-рейтинговой системе с использованием кредитно-зачетных единиц. Итоговые баллы по результатам изучения дисциплинарных модулей и всего курса основывается на интегральной оценке всех видов учебной (аудиторной, внеаудиторной, самостоятельной).

Текущий контроль по курсу «Языки и системы программирования» включает:

- *лекционные занятия (2 часа)*: неявка на занятие – 0; посещение занятия – 1 балл; за конспектирование лекции или ее самостоятельное составление – 2 балла; за активное участие в лекции – 2 балла (максимальное количество баллов за модуль – 3 лекции × 5 балла = 15 баллов);
- *лабораторные занятия (2 часа)*: неявка на занятие – 0; посещение занятия – 1 балл; за выполнение решенного варианта задания – 1 балл; за выполнение индивидуального задания – 3 балла, за защиту выполненной работы и ответы на контрольные вопросы – 2 балла (максимальное количество баллов за модуль – 5 занятий × (1+1+3+2) балла = 35 баллов).

Максимальное количество баллов по результатам текущей работы и промежуточного контроля по дисциплинарному модулю (без учета бонусов) – 100 баллов (текущая работа – 50 баллов, промежуточный контроль (тестирование) – 50 баллов).

Промежуточный контроль представляет собой выполнение тестовых заданий.

Модуль №1 (50 тестовых вопросов по 1 баллу) – 50 баллов; модуль №2 (25 тестовых вопросов по 2 балла) – 50 баллов; модуль №3 (20 тестовых вопросов по 2 балла + задание 21 на составление программы – 10 баллов) – 50 баллов; модуль №4 (15 тестовых вопросов по 3 балла + задание 16 на составление программы 5 баллов) – 50 баллов; модуль №5 (15 тестовых вопросов по 3 балла + задание 16 на составление программы 5 баллов) – 50 баллов.

Дополнительные баллы (бонусы):

- инициативное решение учебных задач на занятиях – 1 балл;
- оригинальное решение задачи – 2 балла;
- решение большего количества задач, чем предусмотрено в модуле – 4 балла;
- реферат – 2 балла.

Дополнительные баллы по результатам участия студентов в научно - исследовательской работе по дисциплине:

- реферат – 1 балл;
- научный доклад – 2 балла;
- публикация в печати – 4 балла;
- участие в работе научного кружка – 4 балла.
- доклады на научно-практической конференции:
  - институтской – 2 балла;
  - университетской – 3 балла;
  - республиканской – 4 балла;
  - Российской – 5 баллов;
  - международной – 6 баллов.
- участие в олимпиаде:
  - институтской – 1 балл;
  - университетской – 2 балла;
  - республиканской – 4 балла;
  - Российской – 6 баллов;
  - международной – 8 баллов.
- получение патента, свидетельства на охрану интеллектуальной собственности – 20 баллов.

Минимальное количество баллов, необходимое для получения положительной оценки по данной дисциплине определено – 51 балл.

После завершения изучения дисциплинарного модуля студенту предоставляется одна неделя для добора баллов.

## VIII. Информационное обеспечение дисциплины

### а) Основная литература

1. Аляев Ю.А., Гладков В.П., Козлов О.А. Практикум по алгоритмизации и программированию на языке Паскаль. – М., 2007.
2. Бобровский С. Delphi 7. – СПб, 2004.
3. Бородич Ю.С., Вальвачев А.Н., Кузьмич А.И. Паскаль для персональных компьютеров. – Минск, 1991.
4. Давыдова Н.А. Программирование. – М., 2009.
5. Культин Н.Б. Turbo Pascal в задачах и примерах. – СПб, 2008.
6. Культин Н.Б. Delphi в задачах и примерах. – СПб, 2012.
7. Могилёв А.В., Пак Н.И., Хеннер Е.К. Информатика – М., 2004.
8. Муртузалиева А.С. Языки и системы программирования: лабораторный практикум; – Махачкала: ДГПУ, 2009. – 134 с.

9. Муртузалиева А.С. Языки и системы программирования: Электронный учебно - методический комплекс. – Регистрационное свидетельство №18546 от 1 марта 2010г. Номер государственной регистрации 0321000183.
10. Незнанов А.А. Программирование и алгоритмизация. – М., 2010.
11. Рапаков Г., Ржеуцкая С. Turbo Pascal для студентов и школьников. – СПб, 2012.
12. Сеницын С.В. Программирование на языке высокого уровня. – М., 2010.
13. Фаронов В.В. Turbo Pascal 7.0. – СПб, 2009.

*б) Дополнительная литература*

14. Бен-Ари М. Языки программирования. Практический сравнительный анализ. – М., 2000.
15. Иванова Г.С. Основы программирования. – М., 2001.
16. Иванова Г.С., Ничушкина Т.Н., Пугачев Е.К. Объектно-ориентированное программирование. – М., 2001.
17. Кнут, Дональд, Эрвин. Искусство программирования. Том 1-3. – М., 2001.
18. Лапчик М.П., Семакин И.Г., Хеннер Е.К. Методика преподавания информатики. Учебное пособие для студентов педагогических вузов. – М., 2005.
19. Могилев А.В., Пак Н.И., Хеннер Е.К. Практикум по информатике. – М., 2001.
20. Семакин И.Г., Шестаков А.П. Основы программирования. – М., 2001.
21. Симонович С., Евсеев Г. Занимательное программирование Delphi. – М., 2001.

*в) Интернет-ресурсы*

22. [http://www.bsuir.by/m/12\\_101945\\_1\\_54215.pdf](http://www.bsuir.by/m/12_101945_1_54215.pdf)
23. [http://wiki.iteach.ru/images/b/b8/Лекции\\_-\\_ПаскальFom.pdf](http://wiki.iteach.ru/images/b/b8/Лекции_-_ПаскальFom.pdf)
24. [http://umniki-sm.ucoz.ru/sovet\\_prohitat/paskal.doc](http://umniki-sm.ucoz.ru/sovet_prohitat/paskal.doc)
25. [http://pgtk.edu.ru/legacy/books/delphi\\_prakt/lab6.htm](http://pgtk.edu.ru/legacy/books/delphi_prakt/lab6.htm)
26. [http://mo-mif.gimis-rk.ru/files/02.12.2014\\_72ca36da2110a63e66f76dab0c07021c/](http://mo-mif.gimis-rk.ru/files/02.12.2014_72ca36da2110a63e66f76dab0c07021c/) Информатика\_laboratornye\_raboty\_Pascal.doc
27. <http://citforum.ru/>
28. <http://www.intuit.ru/>

**IX. Методические указания для обучающихся по освоению дисциплины (модуля)**

1. Методические указания к лабораторным работам
2. Тестовые задания для промежуточной аттестации
3. Рабочая программа дисциплины

**X. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю)**

1. Компьютерные программы: среда программирования Turbo Pascal, среда программирования Delphi.

**XI. Материально-техническое обеспечение дисциплины**

1. Лекционная аудитория (оборудованная проектором)
2. Аудитория для лабораторных работ (с наличием персональных компьютеров с соответствующим аппаратным и программным обеспечением)
3. Компьютерный класс для проведения промежуточного контроля (компьютерного тестирования) с локальной сетью.

**Материально-техническое обеспечение дисциплины**

Реализация дисциплины требует наличия лекционной аудитории, экран, мультимедийный проектор, ноутбук, раздаточный материал. Комплект лабораторных работ и карточек заданий из расчета два экземпляра на одного магистра.

**Специальные условия для инвалидов и лиц с ограниченными возможностями здоровья**

Специальные условия обучения и направления работы с инвалидами и лицами с ограниченными возможностями здоровья (далее - обучающиеся с ограниченными возможностями здоровья) определены на основании:

- Федерального закона от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;
- Федерального закона от 24.11.1995 № 181-ФЗ «О социальной защите инвалидов в Российской Федерации»;

- приказа Министерства образования и науки Российской Федерации (Минобрнауки России) от 6 апреля 2021 г. № 245 «Об утверждении Порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры»;

- методических рекомендаций по организации образовательного процесса для обучения инвалидов и лиц с ограниченными возможностями здоровья в образовательных организациях высшего образования, в том числе оснащённости образовательного процесса, утвержденных Минобрнауки России 08.04.2014 № АК-44/05вн).

Под специальными условиями для получения образования обучающихся с ограниченными возможностями здоровья понимаются условия обучения, воспитания и развития таких студентов, включающие в себя использование при необходимости адаптированных образовательных программ и методов обучения и воспитания, специальных учебников, учебных пособий и дидактических материалов, специальных технических средств обучения коллективного и индивидуального пользования, предоставление услуг ассистента (помощника), оказывающего необходимую помощь, проведение групповых и индивидуальных коррекционных занятий, обеспечение доступа в здания вуза и другие условия, без которых невозможно или затруднено освоение образовательных программ обучающихся с ограниченными возможностями здоровья.

Обучение в рамках учебной дисциплины обучающихся с ограниченными возможностями здоровья осуществляется институтом с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.

Обучение по учебной дисциплине обучающихся с ограниченными возможностями здоровья может быть организовано как совместно с другими обучающимися, так и в отдельных группах.

В целях доступности обучения по дисциплине обеспечивается:

- 1) для лиц с ограниченными возможностями здоровья по зрению:
  - наличие альтернативной версии официального сайта института в сети «Интернет» для слабовидящих;
  - весь необходимый для изучения материал, согласно учебному плану (в том числе, для обучающихся по индивидуальным учебным планам) предоставляется в электронном виде на диске.
  - индивидуальное равномерное освещение не менее 300 люкс;
  - присутствие ассистента, оказывающего обучающемуся необходимую помощь;
  - обеспечение возможности выпуска альтернативных форматов печатных материалов (крупный шрифт или аудиофайлы);
  - обеспечение доступа обучающегося, являющегося слепым и использующего собаку-проводника, к зданию института.
- 2) для лиц с ограниченными возможностями здоровья по слуху:
  - наличие микрофонов и звукоусиливающей аппаратуры коллективного пользования (аудиоколонки);
- 3) для лиц с ограниченными возможностями здоровья, имеющих нарушения опорно-двигательного аппарата, материально-технические условия должны обеспечивать возможность беспрепятственного доступа обучающихся в учебные помещения, столовые, туалетные и другие помещения организации, а также пребывания в указанных помещениях (наличие пандусов, поручней, расширенных дверных проемов и других приспособлений).

Перед началом обучения могут проводиться консультативные занятия, позволяющие студентам с ограниченными возможностями адаптироваться к учебному процессу.

В процессе ведения учебной дисциплины профессорско-преподавательскому составу рекомендуется использование социально-активных и рефлексивных методов обучения, технологий социокультурной реабилитации с целью оказания помощи обучающимся с ограниченными возможностями здоровья в установлении полноценных межличностных отношений с другими обучающимися, создании комфортного психологического климата в учебной группе.

Особенности проведения текущей и промежуточной аттестации по дисциплине для обучающихся с ограниченными возможностями здоровья устанавливаются с учетом индивидуальных психофизических особенностей (устно, письменно на бумаге, письменно на компьютере, в форме тестирования и другое). При необходимости предоставляется дополнительное время для подготовки ответа на зачете или экзамене.