

Федеральное государственное бюджетное образовательное учреждение высшего образования «Тамбовский государственный университет имени Г.Р. Державина»
Инженерно-технический институт
Кафедра математического моделирования и информационных технологий

УТВЕРЖДАЮ:
И.о. директора института



Т. В. Пасько
«17» июня 2026 г.

РАБОЧАЯ ПРОГРАММА

по дисциплине Б1.В.ДВ.05.05 Спортивное программирование: уровень 2

Направление подготовки/специальность: 03.03.02 - Физика

Профиль/направленность/специализация: Физические основы робототехнических систем

Уровень высшего образования: бакалавриат

Квалификация: Бакалавр

год набора: 2026

Автор программы:

Кандидат педагогических наук, доцент Клыгина Елена Владимировна

Рабочая программа составлена в соответствии с ФГОС ВО по направлению подготовки 03.03.02 - Физика (уровень бакалавриата) (приказ Министерства науки и высшего образования РФ от «07» августа 2020 г. № 891).

Рабочая программа принята на заседании Кафедры математического моделирования и информационных технологий «16» июня 2026 г. Протокол № 11

Рассмотрена и одобрена на заседании Ученого совета Инженерно-технического института, Протокол от «17» июня 2026 г. № 10.

СОДЕРЖАНИЕ

1. Цели и задачи дисциплины.....	4
2. Место дисциплины в структуре ОП бакалавриата.....	6
3. Объем и содержание дисциплины.....	6
4. Контроль знаний обучающихся и типовые оценочные средства.....	10
5. Методические указания для обучающихся по освоению дисциплины (модуля).....	30
6. Учебно-методическое и информационное обеспечение дисциплины.....	31
7. Материально-техническое обеспечение дисциплины, программное обеспечение, профессиональные базы данных и информационные справочные системы.....	32

1. Цели и задачи дисциплины

1.1 Цель дисциплины – формирование компетенций:

УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач

1.2 Типы задач профессиональной деятельности, к которым готовятся обучающиеся в рамках освоения дисциплины:

- научно-исследовательский
- педагогический

1.3 Дисциплина ориентирована на подготовку обучающихся к профессиональной деятельности в сферах: 01 Образование и наука (в сферах: реализации образовательных программ среднего общего образования, среднего профессионального образования, высшего образования и дополнительных профессиональных программ; научных исследований и научно-конструкторских разработок), 40 Сквозные виды деятельности в промышленности (в сферах: фундаментальных основ физики живых систем и физико-химической биологии, применения диагностического и лечебного оборудования, участия в инновационных и опытно-конструкторских разработках; эксплуатации электронных приборов и систем различного назначения; мониторинга параметров материалов; мониторинга состояния окружающей среды)

1.4 В результате освоения дисциплины у обучающихся должны быть сформированы:

Обобщенные трудовые функции / трудовые функции / трудовые или профессиональные действия (при наличии профстандарта)	Код и наименование компетенции ФГОС ВО, необходимой для формирования трудового или профессионального действия	Индикаторы достижения компетенций
	УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	Анализирует задачу, выделяя ее базовые составляющие. Осуществляет декомпозицию задачи. Определяет, интерпретирует и ранжирует информацию, требуемую для решения поставленной задачи. Находит и критически анализирует информацию, необходимую для решения поставленной задачи. Рассматривает различные варианты решения задачи, оценивая их достоинства и недостатки. Определяет и оценивает практические последствия возможных решений задачи

1.5 Согласование междисциплинарных связей дисциплин, обеспечивающих освоение компетенций:

УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач

№ п/п	Наименование дисциплин, определяющих междисциплинарные связи	Форма обучения				
		Очная (семестр)				
		1	2	3	4	8
1	CAD-системы автоматического проектирования		+			
2	CAE-системы автоматического проектирования			+		

3	САМ-системы автоматического проектирования				+	
4	Алгоритмы машинного обучения			+		
5	Безопасность компьютерных сетей			+		
6	Бэкэнд - программирование			+		
7	Введение в высшую математику	+				
8	Введение в спортивное программирование		+			
9	Высшая математика		+			
10	Вычислительные среды обработки данных		+			
11	Графический дизайн			+		
12	Демонстрационный эксперимент в школе		+			
13	Инструменты визуализации данных				+	
14	Комбинаторный анализ		+			
15	Компьютерная графика				+	
16	Математическое и компьютерное моделирование			+		
17	Математическое моделирование и визуализация научных данных средствами языка Python				+	
18	Методы решения геометрических задач				+	
19	Организация школьного кабинета физики			+		
20	Программирование Java		+			
21	Программирование на PHP				+	
22	Программирование устройств телекоммуникаций на языках высокого уровня				+	
23	Программирование устройств телекоммуникаций на языках низкого уровня			+		

24	Проективная геометрия			+		
25	Решение олимпиадных физических задач				+	
26	Системы многоуровневой защиты информации				+	
27	Спортивное программирование: уровень 1			+		
28	Стандарты в области информационной безопасности		+			
29	Физика и химия твёрдого тела		+			
30	Философия					+
31	Цифровая культура	+				
32	Цифровая обработка изображений		+			
33	Экспертные системы		+			

2. Место дисциплины в структуре ОП бакалавриата:

Дисциплина «Спортивное программирование: уровень 2» относится к части, формируемой участниками образовательных отношений, учебного плана ОП по направлению подготовки 03.03.02 - Физика.

Дисциплина «Спортивное программирование: уровень 2» изучается в 4 семестре.

3. Объем и содержание дисциплины

3.1. Объем дисциплины: 2 з.е.

Очная: 2 з.е.

Вид учебной работы	Очная (всего часов)
Общая трудоёмкость дисциплины	144
Контактная работа	64
Лабораторные (Лаб. раб.)	64
Самостоятельная работа (СР)	80
Зачет	-

3.2. Содержание курса:

№ темы	Название раздела/темы	Вид учебной работы, час.		Формы текущего контроля
		Лаб · раб.	СР	
		О	О	
4 семестр				

1	Рекурсивные вычисления	5	6	Опрос; Выполнение практических заданий
2	Основы визуального программирования в среде Delphi	5	6	Опрос; Выполнение практических заданий
3	Работа с графикой в среде Delphi	5	7	Выполнение практических заданий; Выполнение практических заданий - контрольный срез
4	Указатели. Динамическая память	5	7	Опрос; Выполнение практических заданий
5	Динамические структуры данных	6	7	Выполнение практических заданий; Тестирование; Выполнение практических заданий - контрольный срез
6	Модули	6	7	Выполнение практических заданий

Тема 1. Рекурсивные вычисления

Лекция.

Рекурсивные определения. Свойства рекурсивных алгоритмов. Формы рекурсивных записей. Рекурсивные и итерационные алгоритмы обработки данных. Условия, обеспечивающие завершение последовательности рекурсивных вызовов. Идеи реализации рекурсивных вызовов в подпрограммах. Взаимосвязь итерации и рекурсии. Быстрая сортировка. Стековый калькулятор. Задача о Ханойских башнях.

Лабораторные работы.

Практическое занятие 1. Разработка и реализация рекурсивных программ. Решение задач с использованием рекурсии.

Практическое занятие 2. Разработка и реализация рекурсивных программ. Стековый калькулятор. Задача о Ханойских башнях.

Практическое занятие 3. Использование рекурсии.

Задания для самостоятельной работы.

1. Написать подпрограмму записи десятичного числа в двоичном представлении.
2. Написать подпрограмму записи числа в обратном порядке.
3. Написать подпрограмму вычисления значения функции $f(n)$, определенной следующим образом: $f(1)=3$, $f(2)=1$, $f(n)=2*f(n-1)-f(n-2)$, если $n>2$.

Тема 2. Основы визуального программирования в среде Delphi

Лекция.

Принципы визуального программирования. Основы работы в среде Delphi. Проекты Delphi. Организация проекта в Delphi, основные файлы проектов. Структура модулей. Заголовок модуля и связь модулей друг с другом. Интерфейсная часть. Исполняемая часть. Инициализирующая и завершающая части. Структура файла проекта. Управление проектом. Создание и сохранение нового проекта. Структура приложения. Общая характеристика визуальных компонентов. Основные свойства формы, типы форм и их установка. Простейшие визуальные компоненты (текстовые, кнопки, списки, индикаторы, группирующие компоненты, компоненты прокрутки и т.д.): свойства, методы, события. Применение диалоговых компонентов.

Лабораторные работы.

Лабораторная работа 1. Разработка приложений с использованием простейших компонентов ввода-вывода, управления (Edit, Label, StringGrid, Memo, Button, RadioGroup, RadioButton).

Разработка простейших приложений.

Лабораторная работа 2. Основы работы в визуальной среде программирования Delphi. Создание интерфейса программы. Элементы управления для выбора из списков. (ListBox, ComboBox).

Элементы UpDown, DateTimePicker и MonthCalendar.

Лабораторная работа 3. Основы работы в визуальной среде программирования Delphi. Разработка приложений. Диалоговые компоненты (OpenDialog, SaveDialog, FontDialog, PrintDialog, MainMenu, PopupMenu).

Лабораторная работа 4. Основы работы в визуальной среде программирования Delphi. Разработка приложений с усложнённым интерфейсом. Многооконные приложения. Модальные и немодальные окна.

Лабораторная работа 5. Основы работы в визуальной среде программирования Delphi. Работа с массивами.

Лабораторная работа 6. Основы работы в визуальной среде программирования Delphi. Работа с числовыми, текстовыми и файлами записей.

Задания для самостоятельной работы.

1. Разработать приложение «Калькулятор для научных вычислений».
2. Организация форм для регистрации пользователя программы с сохранением данных во внешнем файле.
3. Разработать приложение, с помощью которого можно отредактировать текст, задать параметры шрифта для отдельно выделяемых фрагментов, сохранить в файле на диске.

Тема 3. Работа с графикой в среде Delphi

Лекция.

Обработка графики. Холст. Карандаш и кисть. Карандаш. Кисть. Вывод текста. Методы вычерчивания графических примитивов. Линия. Ломаная линия. Окружность и эллипс. Дуга. Прямоугольник. Многоугольник. Сектор. Точка. Вывод иллюстраций. Создание анимации. Метод базовой точки. Построение графиков функций. Использование рекурсии в графических построениях.

Лабораторные работы.

Лабораторная работа 1. Разработка приложений с использованием графических возможностей Delphi.

Лабораторная работа 2. Построение графиков функций.

Лабораторная работа 3. Построение анимационных изображений.

Лабораторная работа 4. Использование рекурсии в графических построениях.

Задания для самостоятельной работы.

1. Разработать приложение демонстрации летящего шара (круг), разбрасывающего искры (разноцветные точки).
2. Разработать приложение, при запуске которого на форме рисуется круг и при нажатии на кнопку круг передвигается вправо на 5-7 миллиметров. При решении задачи использовать метод базовой точки.

3. Разработать рекурсивную программу построения кривой Леви.
4. Создать простейший графический редактор, позволяющий рисовать фигуры, а также сохранять и открывать файлы типа *.bmp.

Тема 4. Указатели. Динамическая память

Лекция.

Указатель. Типизированные и нетипизированные указатели. Работа с динамической памятью. Достоинства и недостатки динамической организации памяти. Основные процедуры и функции работы с динамическими переменными. Способы описания динамических массивов. Расположение массивов в памяти. Основные принципы работы с динамическими массивами. Примеры.

Лабораторные работы.

Лабораторная работа 1. Работа с указателями. Распределение памяти. Динамическое выделение памяти.

Лабораторная работа 2. Работа с динамическими массивами.

Задания для самостоятельной работы.

1. Дана последовательность целых чисел, количество которых заранее не определено. Считать концом последовательности ввод нулевого значения. Ввести и напечатать на разных строках четные и нечетные числа данной последовательности.
2. Создать динамический одномерный массив, заполнить его случайными числами и вывести на печать. Добавить после каждого четного элемента элемент со значением 0.
3. Создать динамический двумерный массив, заполнить его случайными числами и вывести на печать. Добавить K строк в начало матрицы.
4. Дана последовательность символов латинского алфавита, количество которых заранее не определено. Считать концом последовательности символ пробела. Вывести на печать гласные буквы из данной последовательности.

Тема 5. Динамические структуры данных

Лекция.

Динамические структуры данных. Организация линейных списков. Линейные списки. Построение списка. Задача поиска элемента в списке. Задача обхода списка. Задача сортировки списка. Рекурсивные алгоритмы работы со списками. Освобождение памяти, занятой элементами списка, возвращение памяти системе. Стеки, деки, очереди. Моделирование структур данных. Реализация стеков на базе массива и списка. Работа со структурами данных. Примеры.

Лабораторные работы.

Лабораторная работа 1. Работа с динамическими структурами данных: списки.

Цель: научиться разрабатывать программы создания и обработки динамической структуры данных списки.

Лабораторная работа 2. Работа с динамическими структурами данных: стеки.

Цель: научиться разрабатывать программы создания и обработки динамической структуры данных стеки.

Лабораторная работа 3. Работа с динамическими структурами данных: очередь.

Цель: научиться разрабатывать программы создания и обработки динамической структуры данных очередь.

Лабораторная работа 4. Работа с динамическими структурами данных: дек.

Цель: научиться разрабатывать программы создания и обработки динамической структуры данных дек.

Задания для самостоятельной работы.

1. Разработать функцию нахождения среднего арифметического элементов непустого списка.
2. Разработать рекурсивную функцию проверки наличия в списке заданного значения.
3. Разработать процедуру перестановки первого и последнего элементов непустого списка.
4. Разработать процедуру удаления из списка второго по величине элемента, если такой есть.

5. Примеры использования структуры данных «стек».
6. Примеры использования структуры данных «очередь».

Тема 6. Модули

Лекция.

Структура модулей. Заголовок модуля и связь модулей друг с другом. Интерфейсная часть. Исполняемая часть. Инициализирующая и завершающие части. Доступ к объединенным в модуле объектам. Типы модулей в Delphi: модули данных, модули динамических объектов, пакеты, модули потока команд.

Лабораторные работы.

Лабораторная работа 1. Модули. Структура модуля.

Лабораторная работа 2. Стандартные модули Delphi.

Задания для самостоятельной работы.

1. Разработать модуль для работы с комплексными числами.
2. Разработать модуль с типами и функциями для вычисления площади и периметра трапеции.
3. Разработать модуль с типами и функциями для вычисления площади и длины окружности.
4. Разработать модуль с типами и функциями для вычисления площади поверхности и объема шара.
5. Разработать модуль для вычисления площади и периметра многоугольника, заданного координатами вершин.

4. Контроль знаний обучающихся и типовые оценочные средства

4.1. Распределение баллов:

4 семестр

- текущий контроль – 50 баллов
- контрольные срезы – 2 среза по 10 баллов каждый

Распределение баллов по заданиям:

№ темы	Название темы / вид учебной работы	Формы текущего контроля / срезы	Мак. кол-во баллов	Методика проведения занятия и оценки
1.	Рекурсивные вычисления	Опрос	5	5 баллов – студент умеет сопоставить полученную при подготовке к практическому занятию информацию, сравнивать разные точки зрения на анализируемую проблему, умеет четко формулировать свои вопросы и отвечать на задаваемые ему вопросы; 4 балла - студент умеет применять полученную при подготовке к практическому занятию информацию, отвечать на большинство вопросов, вести дискуссию; 3 балла – студент владеет теоретическим материалом по теме практического занятия, иногда затрудняется при ответе на вопросы; 2 балла - студент в неполной мере владеет теоретическим материалом по теме практического занятия, затрудняется сформулировать свою точку зрения на обсуждаемую проблему. Если студент не владеет проблематикой практического занятия, не может отвечать на вопросы, зачитывает ответ по напечатанному тексту – ответ баллами не оценивается.

		Выполнение практических заданий	5	5 баллов – практическое задание самостоятельно выполнено в полном объеме, студент свободно владеет материалом, демонстрирует глубокие, систематизированные знания, уверенно, логично, последовательно и аргументировано отвечает на вопросы, используя профессиональную терминологию; 4 балла – практическое задание самостоятельно и в основном правильно выполнено, студент владеет представленным материалом, логично, последовательно и аргументировано отвечает на заданные вопросы, используя профессиональную терминологию; 3 балла - практическое задание в целом выполнено, но в процессе выполнения задания допущены существенные ошибки, студент не в полной мере владеет информацией по теме; 2 балла - практическое задание выполнено не в полной мере, студент слабо владеет информацией по теме, затрудняется с ответами на задаваемые вопросы.
2.	Основы визуального программирования в среде Delphi	Опрос	5	5 баллов – студент умеет сопоставить полученную при подготовке к практическому занятию информацию, сравнивать разные точки зрения на анализируемую проблему, умеет четко формулировать свои вопросы и отвечать на задаваемые ему вопросы; 4 балла - студент умеет применять полученную при подготовке к практическому занятию информацию, отвечать на большинство вопросов, вести дискуссию; 3 балла – студент владеет теоретическим материалом по теме практического занятия, иногда затрудняется при ответе на вопросы; 2 балла - студент в неполной мере владеет теоретическим материалом по теме практического занятия, затрудняется сформулировать свою точку зрения на обсуждаемую проблему. Если студент не владеет проблематикой практического занятия, не может отвечать на вопросы, зачитывает ответ по напечатанному тексту – ответ баллами не оценивается.
		Выполнение практических заданий	5	5 баллов – практическое задание самостоятельно выполнено в полном объеме, студент свободно владеет материалом, демонстрирует глубокие, систематизированные знания, уверенно, логично, последовательно и аргументировано отвечает на вопросы, используя профессиональную терминологию; 4 балла – практическое задание самостоятельно и в основном правильно выполнено, студент владеет представленным материалом, логично, последовательно и аргументировано отвечает на заданные вопросы, используя профессиональную терминологию; 3 балла - практическое задание в целом выполнено, но в процессе выполнения задания допущены существенные ошибки, студент не в полной мере владеет информацией по теме; 2 балла - практическое задание выполнено не в полной мере, студент слабо владеет информацией по теме, затрудняется с ответами на задаваемые вопросы.

3.	Работа с графикой в среде Delphi	Выполнение практических заданий	5	5 баллов – практическое задание самостоятельно выполнено в полном объеме, студент свободно владеет материалом, демонстрирует глубокие, систематизированные знания, уверенно, логично, последовательно и аргументировано отвечает на вопросы, используя профессиональную терминологию; 4 балла – практическое задание самостоятельно и в основном правильно выполнено, студент владеет представленным материалом, логично, последовательно и аргументировано отвечает на заданные вопросы, используя профессиональную терминологию; 3 балла - практическое задание в целом выполнено, но в процессе выполнения задания допущены существенные ошибки, студент не в полной мере владеет информацией по теме; 2 балла - практическое задание выполнено не в полной мере, студент слабо владеет информацией по теме, затрудняется с ответами на задаваемые вопросы.
		Выполнение практических заданий - контрольный срез (контрольный срез)	10	Контрольный срез выполняются по тематике практических занятий. Максимальное количество 10 баллов ставится за абсолютно правильно выполненное практическое задание. Студент показал полный объем знаний, умений в освоении пройденных тем и применение их на практике. Нет ошибок в логических рассуждениях, студент свободно владеет материалом, демонстрирует глубокие, систематизированные знания, свободно отвечает на вопросы, используя профессиональную терминологию; 9 баллов – задание выполнено полностью, но допущены 1-3 легко устранимых недочета. Студент владеет представленным материалом, отвечает на заданные вопросы; 8 баллов – за полностью выполненное задание с 1-2 незначительными ошибками; 7 баллов - за полностью выполненное задание с 1-2 ошибками; 6 баллов – за 70% выполнения задания без ошибок; 5 баллов – за 70% выполнения задания с небольшими недочетами; 4 балла – за 60% выполнения задания с небольшими недочетами. 1-3 балла - задание выполнено не полностью, допущены грубые ошибки. 0 баллов – работа не сдана.
4.	Указатели. Динамическая память	Опрос	5	5 баллов – студент умеет сопоставить полученную при подготовке к практическому занятию информацию, сравнивать разные точки зрения на анализируемую проблему, умеет четко формулировать свои вопросы и отвечать на задаваемые ему вопросы; 4 балла - студент умеет применять полученную при подготовке к практическому занятию информацию, отвечать на большинство вопросов, вести дискуссию; 3 балла – студент владеет теоретическим материалом по теме практического занятия, иногда затрудняется при ответе на вопросы; 2 балла - студент в неполной мере владеет теоретическим материалом по теме практического занятия, затрудняется сформулировать свою точку зрения на обсуждаемую проблему. Если студент не владеет проблематикой практического занятия, не может отвечать на вопросы, зачитывает ответ по напечатанному тексту – ответ баллами не оценивается.

		Выполнение практических заданий	5	5 баллов – практическое задание самостоятельно выполнено в полном объеме, студент свободно владеет материалом, демонстрирует глубокие, систематизированные знания, уверенно, логично, последовательно и аргументировано отвечает на вопросы, используя профессиональную терминологию; 4 балла – практическое задание самостоятельно и в основном правильно выполнено, студент владеет представленным материалом, логично, последовательно и аргументировано отвечает на заданные вопросы, используя профессиональную терминологию; 3 балла - практическое задание в целом выполнено, но в процессе выполнения задания допущены существенные ошибки, студент не в полной мере владеет информацией по теме; 2 балла - практическое задание выполнено не в полной мере, студент слабо владеет информацией по теме, затрудняется с ответами на задаваемые вопросы.
5.	Динамические структуры данных	Выполнение практических заданий	5	5 баллов – практическое задание самостоятельно выполнено в полном объеме, студент свободно владеет материалом, демонстрирует глубокие, систематизированные знания, уверенно, логично, последовательно и аргументировано отвечает на вопросы, используя профессиональную терминологию; 4 балла – практическое задание самостоятельно и в основном правильно выполнено, студент владеет представленным материалом, логично, последовательно и аргументировано отвечает на заданные вопросы, используя профессиональную терминологию; 3 балла - практическое задание в целом выполнено, но в процессе выполнения задания допущены существенные ошибки, студент не в полной мере владеет информацией по теме; 2 балла - практическое задание выполнено не в полной мере, студент слабо владеет информацией по теме, затрудняется с ответами на задаваемые вопросы.
		Тестирование	5	За тестирование максимальное количество баллов 5 ставится за правильное выполнение 90-100% тестовых заданий; 4 балла – за 75-89% выполнения тестовых заданий; 3 балла – за 60-74% выполнения тестовых заданий; 2 балла – за 50-59% выполнения тестовых заданий. Менее 50% правильных ответов баллов не дает.
		Выполнение практических заданий - контрольный срез (контрольный срез)	10	Контрольный срез выполняются по тематике практических занятий. Максимальное количество 10 баллов ставится за абсолютно правильно выполненное практическое задание. Студент показал полный объем знаний, умений в освоении пройденных тем и применение их на практике. Нет ошибок в логических рассуждениях, студент свободно владеет материалом, демонстрирует глубокие, систематизированные знания, свободно отвечает на вопросы, используя профессиональную терминологию; 9 баллов – задание выполнено полностью, но допущены 1-3 легко устранимых недочета. Студент владеет представленным материалом, отвечает на заданные вопросы; 8 баллов – за полностью выполненное задание с 1-2 незначительными ошибками; 7 баллов - за полностью выполненное задание с 1-2 ошибками; 6 баллов – за 70% выполнения задания без ошибок; 5 баллов – за 70% выполнения задания с небольшими недочетами; 4 балла – за 60% выполнения задания с небольшими недочетами. 1-3 балла - задание выполнено не полностью, допущены грубые ошибки. 0 баллов – работа не сдана.

6.	Модули	Выполнение практических заданий	5	5 баллов – практическое задание самостоятельно выполнено в полном объеме, студент свободно владеет материалом, демонстрирует глубокие, систематизированные знания, уверенно, логично, последовательно и аргументировано отвечает на вопросы, используя профессиональную терминологию; 4 балла – практическое задание самостоятельно и в основном правильно выполнено, студент владеет представленным материалом, логично, последовательно и аргументировано отвечает на заданные вопросы, используя профессиональную терминологию; 3 балла - практическое задание в целом выполнено, но в процессе выполнения задания допущены существенные ошибки, студент не в полной мере владеет информацией по теме; 2 балла - практическое задание выполнено не в полной мере, студент слабо владеет информацией по теме, затрудняется с ответами на задаваемые вопросы.
7.	Итого за семестр		70	

Итоговая оценка по зачету выставляется в 100-балльной шкале и в традиционной четырехбалльной шкале. Перевод 100-балльной рейтинговой оценки по дисциплине в традиционную четырехбалльную осуществляется следующим образом:

100-балльная система	Традиционная система
50 - 100 баллов	Зачтено
0 - 49 баллов	Не зачтено

4.2 Типовые оценочные средства текущего контроля

Выполнение практических заданий

Тема 1. Рекурсивные вычисления

Практическое занятие 1. Разработка и реализация рекурсивных программ. Решение задач с использованием рекурсии.

Практические задания.

1. Определить функцию $S(n)$, вычисляющую сумму цифр заданного натурального числа, и с ее помощью написать соответствующую подпрограмму-функцию.
2. Для целого неотрицательного n вычислить значение функции $f(n)$, определенной следующим образом: $f(0)=0$, $f(1)=1$, при $n>2$ $f(n) = f(n \text{ div } 2) + f(n \text{ div } 3)$.
3. Дать рекурсивное определение суммы и вычислить сумму элементов одномерного массива.
4. Определить, является ли заданная строка палиндромом.
5. Дать рекурсивное определение максимума элементов последовательности и определить максимум одномерного массива.
6. Дать рекурсивное определение возведения числа в степень и описать в программе.
7. Рекурсивно записать число в обратном порядке.
8. Написать подпрограмму вычисления N -го числа Фибоначчи: $F(n)=F(n-1)+F(n-2)$.
9. Рекурсивно определить число ближайшее в последовательности Фибоначчи.
10. Рекурсивно определить наибольший общий делитель двух натуральных чисел: $\text{Нод}(x,y)=x$, если $x=y$, $\text{Нод}(x,y)=\text{Нод}(x-y,y)$, если $x>y$, $\text{Нод}(x,y)=\text{Нод}(x,y-x)$, если $y>x$. Написать подпрограмму вычисления значения $\text{Нод}(x,y)$.
- 11.

Практическое занятие 2. Разработка и реализация рекурсивных программ. Стековый калькулятор. Задача о Ханойских башнях.

Практические задания.

1. Реализовать программу «Стековый калькулятор».
2. Реализовать рекурсивную программу решения задачи о Ханойских башнях.

Практическое занятие 3. Использование рекурсии.

Практические задания.

1. Реализуйте программу для решения задачи о восьми ферзях.
2. Напишите программу для реализации рисования снежинок Коха.
3. Напишите программу реализации задачи «Ход конем». Цель задачи – заставить коня ступить на каждую клетку шахматной доски всего один раз.
4. Реализация методов сортировки с использованием рекурсии. Быстрая сортировка.

Тема 2. Основы визуального программирования в среде Delphi

Лабораторная работа 1. Разработка приложений с использованием простейших компонентов ввода-вывода, управления (Edit, Label, StringGrid, Memo, Button, RadioGroup, RadioButton).

Разработка простейших приложений.

Практические задания.

1. Создать простейшее приложение и разработать его интерфейс: 1) изменить заголовок, размер, цвет и стиль формы; 2) поместить на форму надпись и изменить ее свойства (цвет, шрифт, начертание и размер).
2. При помощи визуальных компонентов создайте приложение «Вычисление арифметических операций».
3. На форме отобразить два компонента StringGrid1 и StringGrid2. В StringGrid1 с клавиатуры ввести значения элементов массива, а в StringGrid2 эти значения отобразить.

Лабораторная работа 2. Основы работы в визуальной среде программирования Delphi. Создание интерфейса программы. Элементы управления для выбора из списков. (ListBox, ComboBox).

Элементы UpDown, DateTimePicker и MonthCalendar.

Практические задания.

1. Разработать программу, вводящую вещественные значения a , b , c и выполняющую один из вариантов вычисления: 1) определение максимального из данных чисел; 2) определение минимального из данных чисел в зависимости от выбора пользователя. Выбор варианта вычисления оформить с помощью компонента ListBox. Вывод результата выполнить в поле Label.
2. Разработать программу, вводящую вещественные значения сторон треугольника a , b , c и выполняющую один из вариантов вычисления: 1) определение периметра треугольника $p=a+b+c$; 2) определение площади треугольника $S=\sqrt{k*(k-a)*(k-b)*(k-c)}$, $k=(a+b+c)/2$ в зависимости от выбора пользователя. Выбор варианта вычисления оформить с помощью компонента ComboBox. Вывод результата выполнить в поле Label.
3. Заполнить одномерный массив случайными числами и вывести полученный результат в ListBox.
4. Разработать программу, в которой организовать подсчет количества дней между двумя датами, выбранных из календаря.
5. Разработать программу, в которой организовать ввод мероприятий, намеченных на определенную дату в будущем, и своевременного напоминания о них.

Лабораторная работа 3. Основы работы в визуальной среде программирования Delphi. Разработка приложений. Диалоговые компоненты (OpenDialog, SaveDialog, FontDialog, PrintDialog, MainMenu, PopupMenu).

Практические задания.

1. Разработать приложение с возможностью выбора загружаемого файла, выбора шрифта, настройки принтера, работа с главным и контекстным и другими диалоговыми компонентами.
2. Создать простейший текстовый редактор, позволяющий вводить и редактировать текст, управлять шрифтом и цветом окна, а также открывать и сохранять файлы (аналог стандартного приложения Блокнот).

Лабораторная работа 4. Основы работы в визуальной среде программирования Delphi. Разработка приложений с усложнённым интерфейсом. Многооконные приложения. Модальные и немодальные окна.

Практические задания.

1. Разработать приложение интерфейс которого должен содержать несколько окон, которые должны последовательно менять друг друга при нажатии кнопки Далее. Первая форма должна содержать окно приветствия. Данная форма должна быть главной формой, остальные окна – модальными.
2. Разработать приложение интерфейс которого должен быть организован с использованием компонентов TabControl и PageControl.

Лабораторная работа 5. Основы работы в визуальной среде программирования Delphi. Работа с массивами.

Практические задания.

1. Заполнить одномерный массив случайными числами и вывести полученный результат в строку, в столбец. Заполнить двумерный массив случайными числами и вывести полученный результат.
2. Разработать приложение, в котором можно выбирать различные методы поиска элементов массива (минимальный/максимальный элемент, поиск элемента методом перебора/ дихотомии и т.д.).
3. Заполнить одномерный массив случайными числами. Отсортировать элементы массива методом «Пузырька». Организовать в компоненте Метод вывод исходного массива и вывод содержимого массива в каждом прохождении.
4. Заполнить одномерный массив случайными числами. Отсортировать элементы массива методом Шелла. Организовать в компоненте Метод вывод исходного массива и вывод содержимого массива в каждом прохождении.
5. Разработать приложение, в котором можно выбирать различные методы сортировки для упорядочивания элементов массива.

Лабораторная работа 6. Основы работы в визуальной среде программирования Delphi. Работа с числовыми, текстовыми и файлами записей.

Практические задания.

1. Разработать программу, вычисляющую сумму и среднее арифметическое чисел, хранящихся в указанном пользователе файле данных. Первая строка файла данных содержит единственное число, равное количеству обрабатываемых чисел. При вычислении суммы и среднего арифметического это число не должно учитываться.
2. Дан текстовый файл. Разработать программу, которая подсчитывает количество слов в файле. Программа должна предложить пользователю ввести имя текстового файла в поле ввода и после щелчка по кнопке подсчитать количество слов.
3. Создать файл с типом записи, в котором каждая запись содержит следующие поля о каждом абоненте фамилию, адрес (улица, номер дома), номер телефона. Вывести на экран всю базу данных и информацию об абонентах (фамилия номер телефона), проживающих на определенной улице.
4. Создать файл с типом записи, в котором каждая запись содержит следующие поля об автолюбителях: фамилия, марка автомобиля, цвет, год выпуска. Вывести на экран всю базу данных и информацию о владельцах (фамилия), имеющих автомобиль конкретной марки и определенного цвета.
5. Создать файл с типом записи, в котором каждая запись содержит следующие поля, содержащие фамилии и даты (день, месяц, год) рождения ваших друзей. Вывести на экран всю базу данных и информацию о тех, чье день рождения будет в следующем месяце.

Тема 3. Работа с графикой в среде Delphi

Лабораторная работа 1. Разработка приложений с использованием графических возможностей Delphi.

Практические задания.

1. Разработать программу построения 20 концентрических окружностей разного радиуса, но имеющих общий центр.
2. Разработать программу построения на форме 5 кругов и 5 четырехугольников. Каждый круг и четырехугольник имеет различный цвет, толщину контура, а также различный цвет и стиль заливки.

3. Разработать программу вычерчивания полукруга и квадрата, имеющих общую линию внешнего касания, по заданным радиусу полукруга и длине стороны квадрата. Залить полукруг и квадрат разными цветами. Ввод исходных данных выполнить на основной стартовой форме программы, а вывод графики – на дополнительной форме.

4. Разработать программу построения различных простых фигур (круг, квадрат, прямоугольник и т.д). Реализовать возможность установки цвета линий и заливки фигур.

Лабораторная работа 2. Построение графиков функций.

Практические задания.

1. Разработать программу построения графика функции $f(x)=\sin(x)+\cos(2x)$ в диапазоне изменения аргумента x от a до b .

2. Разработать программу построения на одной координатной сетке графиков функций: $f_1(x)=2-\cos(x)$; $f_2(x)=2\sin(2x)+1$;

$f_3(x)=2f_1+f_2$ в диапазоне изменения аргумента x от $-?$ до $?$.

Лабораторная работа 3. Построение анимационных изображений.

Практические задания.

1. Разработать приложения имитации капли дождя на поверхности воды.

2. Разработать приложения имитации движения шара по бильярдному столу.

3. Разработать приложения построения траектории движения тела под углом к горизонту.

Организовать возможность задания начальной скорости и угла.

4. Разработать приложение демонстрации движения планет солнечной системы.

Лабораторная работа 4. Использование рекурсии в графических построениях.

Практические задания.

1. Разработать рекурсивную программу построения кривой Гильберта.

2. Разработать рекурсивную программу построения Канторова множества.

3. Разработать рекурсивную программу построения кривая Госпера (снежинка Госпера).

4. Разработать рекурсивную программу построения треугольника Серпинского.

5. Разработать рекурсивную программу построения квадрата Серпинского.

6. Разработать рекурсивную программу построения дерева Пифагора.

7. Разработать рекурсивную программу построения кривой Дракона.

Тема 4. Указатели. Динамическая память

Практические занятия.

Лабораторная работа 1. Работа с указателями. Распределение памяти. Динамическое выделение памяти.

Практические задания.

1. Разработать программу демонстрирующую создание, присваивание, сравнение, обнуление (пустой указатель, нулевая ссылка, используется для любого типа), уничтожение динамических переменных.

2. Используя функции и процедуры для работы с памятью и указателями среды программирования Delphi, разработать программу,

в которой реализованы следующие функций:

а) получить значения указателей полей текущего диспетчера памяти; получить указатель на заданный адрес памяти; п

олучить указатель на участок памяти, которую занимает объект;

б) выделить на куче блок памяти заданного размера; выделить блок памяти заданного размера;

с) определить размер памяти, занимаемый переменной; определить, используется ли в настоящий момент диспетчер памяти, установленный по умолчанию, или был установлен другой диспетчер;

д) изменить размер динамически распределенного блока памяти; выполнить бинарное сравнение двух участков памяти;

установить значения полей диспетчера памяти.

3. Используя процедуры для работы с динамическими переменными среды программирования Delphi,

разработать программу, в которой реализованы следующие функций:

- создать динамическую переменную;
- создать динамическую переменную, выделяя под нее заданный объем памяти;
- инициализировать динамическую переменную;
- перераспределяет память для динамической переменной;
- деинициализировать динамическую переменную;
- высвободить память из-под динамической переменной используя процедуры: Dispose, FreeMem, SysFreeMem.

Лабораторная работа 2. Работа с динамическими массивами.

Практические задания.

1. Сформировать динамический одномерный массив, заполнить его случайными числами и вывести на печать.
2. Сформировать динамический одномерный массив. Удалить элемент равный среднему арифметическому элементов массива.
3. Сформировать динамический одномерный массив. Добавить по K элементов в начало и в конец массива.
4. Сформировать динамический двумерный массив, заполнить его случайными числами и вывести на печать.
5. Сформировать динамический двумерный массив. Добавить K столбцов, начиная со столбца с номером N.
6. Сформировать динамический двумерный массив. Удалить строки, начиная со строки K1 и до строки K2.
7. Сформировать динамический двумерный массив. Добавить столбец после столбца, содержащего наибольший элемент.
8. Разработать процедуру, в которой реализованы следующие действия: создание динамического массива, нахождения суммы элементов массива, уничтожение массива. Используя данную процедуру организовать работу с тремя двумерными массивами.
9. Разработать программу, в которой динамический массив используется для вычисления средней оценки произвольного количества студентов.
10. Разработать программу нахождения произведения элементов, расположенных под главной диагональю массива.

Для решения задачи использовать треугольный динамический массив (в данном массиве один из индексов всегда больше другого или равен ему).

11. Разработать программу, предназначенную для записи и хранения на диске названий и авторов книг личной библиотеки.

В программе использовать два динамических массива: один - для названия книг, другой – для имен и фамилий авторов.

Программа должна позволять пользователю вводить информацию о книге, а также находить или удалять данные о ней. Ограничение на количество книг нет.

12. Компания выпускает тостеры, миксеры и электрические хлеборезки. Каждый проданный тостер приносит компании доход в 100 рублей, каждый миксер – 150 рублей, каждая хлеборезка – 50 рублей. Бухгалтер компании накапливает информацию о продажах за последние три месяца.

Ему нужно вычислить доход за каждый месяц, общий доход за три месяца и средний ежемесячный доход. Бухгалтер выполняет этот расчет ежеквартально.

Разработать программу для выполнения всех расчетов. Для ввода данных использовать текстовый файл, который содержит три названия товаров,

доход от продажи каждого товара и объем продаж товара данного вида за каждый из трех месяцев.

Тема 5. Динамические структуры данных

Практическое занятие.

Лабораторная работа 1. Работа с динамическими структурами данных: списки.

Цель: научиться разрабатывать программы создания и обработки динамической структуры данных списки.

Практические задания.

1. Разработать программу создания списка целых чисел и вывести их на экран.

2. Разработать программу реализации типовых операций над списками:

- создание списка;
- добавление звена в начало списка;
- удаление звена из начала списка;
- добавление звена в произвольное место списка, отличное от начала (например, после звена, указатель на которое задан);
- удаление звена из произвольного места списка, отличного от начала (например, после звена, указатель на которое задан);
- проверка, пуст ли список;
- очистка списка;
- печать списка.

3. Дан файл целых чисел, состоящий из нескольких отдельных последовательностей целых чисел, каждая из которых заканчивается нулем.

Ноль не является частью последовательности. Получить новый файл, состоящий из тех же последовательностей,

но внутри каждой числа должны быть записаны в обратном порядке.

Пример:

Дано: 8 10 3 12 0 7 9 2 0 4 12 0

Получить: 12 3 10 8 0 2 9 7 0 12 4 0

Лабораторная работа 2. Работа с динамическими структурами данных: стеки.

Цель: научиться разрабатывать программы создания и обработки динамической структуры данных стеки.

Практические задания.

1. Разработать программу создания стека целых чисел на базе массива и вывести их на экран.

2. Разработать программу создания стека целых чисел на базе списка и вывести их на экран.

3. Разработать программу реализации типовых операций над стеком:

- включение нового элемента;
- исключение элемента из стека;
- чтение элемента из стека;
- очистка стека;
- проверка пустоты стека;
- поиск элемента в стеке;
- определение текущего числа элементов в стеке.

4. Создайте программу, проверяющую скобочную последовательность на наличие ошибок их расстановки.

Лабораторная работа 3. Работа с динамическими структурами данных: очередь.

Цель: научиться разрабатывать программы создания и обработки динамической структуры данных очередь.

Практические задания.

1. Разработать программу создания очереди целых чисел на базе массива и вывести их на экран.

2. Разработать программу создания очереди целых чисел на базе списка и вывести их на экран.

3. Разработать программу реализации типовых операций над очередью:

- включение нового элемента в очередь;
- исключение элемента;
- определение размера очереди;
- очистка очереди;
- поиск элемента в очереди;
- чтение элементов из очереди.

Лабораторная работа 4. Работа с динамическими структурами данных: дек.

Цель: научиться разрабатывать программы создания и обработки динамической структуры данных дек.

Практические задания.

1. Разработать программу создания дека целых чисел на базе одномерного массива и вывести их на экран.
2. Разработать программу создания дека целых чисел на базе списка и вывести их на экран.
3. Разработать программу реализации типовых операций над деком:
 - добавление элемента в начало дека;
 - добавление элемента в конец дека;
 - извлечение элемента из начала дека;
 - извлечение элемента из конца дека;
 - определение размера дека;
 - проверка пустоты дека;
 - поиск элемента в деке;
 - очистка дека.

Тема 6. Модули

Практические занятия.

Лабораторная работа 1. Модули. Структура модуля.

Практические задания.

1. Разработать модуль для вычисления расстояний между точками на плоскости и площади треугольника, заданного тремя точками.
 2. Разработать модуль для работы со стеком. В данном модуле реализовать основные операции над стеком (включение нового элемента, исключение элемента из стека, чтение элемента из стека, очистка стека, проверка пустоты стека, поиск элемента в стеке, определение текущего числа элементов в стеке).
 3. Разработать модуль, реализующий работу с очередью.
 4. Разработать модуль для вычисления расстояний в 3-мерном пространстве.
 5. Разработать модуль с типами и функциями для вычисления площади поверхности и объема куба.
- Лабораторная работа 2. Стандартные модули Delphi.

Практические задания.

1. Разработайте программу демонстрирующую работу с математическими функциями модуля Math.
2. Разработать модуль с дополнительными тригонометрическими функциями.
3. Разработать модуль с типами и функциями для вычисления площади и периметра прямоугольника.
4. Разработайте программу демонстрирующую работу с процедурами и функциями для работы с файлами, дисками, ошибками, строками и т.д. модуля SysUtils.
5. Разработать модуль для работы с матрицами: ввод и вывод значений, проверка на симметричность, заполнение случайными числами и т.д.

Выполнение практических заданий - контрольный срез

Тема 3. Работа с графикой в среде Delphi

В соответствии с вариантом задания разработать программу, демонстрирующую движение объекта.

1. Разработать программу, демонстрирующую вращение отрезка относительно своей середины.
2. Разработать программу, демонстрирующую вращение равнобедренного треугольника вокруг своей высоты, расположенной параллельно вертикальной оси экрана.
3. Разработать программу, демонстрирующую движение точки по внутреннему ободу колеса.
4. Разработать программу, демонстрирующую вращение квадрата относительно центра и одновременным перемещением вдоль экрана по горизонтали.
5. Разработать программу, демонстрирующую отражение ромба относительно осей OX , OY .
6. Разработать программу, демонстрирующую отражение отрезка относительно осей OX и OY .
7. Разработать программу, демонстрирующую масштабирование отрезка.
8. Разработать программу, демонстрирующую поворот треугольника на заданный угол.
9. Разработать программу, демонстрирующую отражение треугольника относительно осей OX , OY .
10. Разработать программу, демонстрирующую поворот ромба на заданный угол.
11. Разработать программу, демонстрирующую поворот отрезка на заданный угол.
12. Разработать программу, демонстрирующую масштабирование треугольника.
13. Разработать программу, демонстрирующую масштабирование ромба.
14. Разработать программу, демонстрирующую «калейдоскоп».
15. Разработать программу, демонстрирующую вращение правильного треугольника вокруг своего центра.
16. Разработать программу, демонстрирующую вращение прямоугольника вокруг одной из своих вершин.
17. Разработать программу, демонстрирующую закон Архимеда.
18. Разработать программу, демонстрирующую броуновское движение.
19. Разработать программу, демонстрирующую колебание тела.
20. Разработать программу, демонстрирующую вращение рамки в однородном магнитном поле.
21. Разработать программу, демонстрирующую преломление света в различных средах.
22. Разработать программу, демонстрирующую траектории движение искусственного спутника.
23. Разработать программу, демонстрирующую движение тела по наклонной плоскости.
24. Разработать программу, демонстрирующую движение тела по окружности.
25. Разработать программу, демонстрирующую модель атома.
26. Разработать программу, демонстрирующую электролиз.

Тема 5. Динамические структуры данных

Создать линейный односвязный список. Разработать программу обработки линейного односвязного списка в соответствии с вариантом задания.

1. Из списка удалить последнюю пару соседних элементов, сумма которых меньше заданной величины, а затем вставить заданное число перед минимальным элементом списка и после него. Освободить память, занимаемую элементами списка.
2. Из списка удалить элемент, стоящий перед минимальным элементом, затем элементы списка расположить в порядке убывания и вставить заданное число, не нарушая порядка расположения. Освободить память, занимаемую элементами списка.
3. Из списка удалить нулевые элементы, затем вставить заданное число перед максимальным элементом списка и после него. Освободить память, занимаемую элементами списка.
4. Из списка удалить последнюю пару соседних элементов, модуль разности между которыми превышает заданную величину, а затем элементы списка расположить в порядке возрастания и вставить заданное число, не нарушая порядка расположения. Освободить память, занимаемую элементами списка.
5. Из списка удалить положительные элементы, превышающие заданную величину, а затем вставить заданное число перед минимальным положительным элементом списка и после него. Освободить память, занимаемую элементами списка.

6. Из списка удалить минимальный и минимальный положительный элементы, а затем положительные элементы расположить в порядке возрастания и вставить заданное положительное число, не нарушая порядка расположения. Освободить память, занимаемую элементами списка.
7. Из списка удалить элемент, стоящий после минимального элемента, а затем вставить заданное число перед максимальным отрицательным элементом и после него. Освободить память, занимаемую элементами списка.
8. Из списка удалить отрицательные элементы, а затем оставшиеся элементы расположить в порядке убывания и вставить заданное положительное число, не нарушая порядка расположения. Освободить память, занимаемую элементами списка.
9. Из списка удалить положительные элементы, а затем вставить число, равное минимальному элементу списка, перед максимальным элементом и после него. Освободить память, занимаемую элементами списка.
10. Из списка удалить максимальный и максимальный отрицательный элементы, затем отрицательные элементы расположить в порядке возрастания и вставить заданное отрицательное число, не нарушая порядка расположения. Освободить память, занимаемую элементами списка.
11. Из списка удалить элемент, стоящий перед максимальным элементом, а затем вставить заданное число после каждого отрицательного элемента. Освободить память, занимаемую элементами списка.
12. Из списка удалить положительные элементы, не превышающие заданную величину, а затем отрицательные элементы списка расположить в порядке убывания и вставить заданное отрицательное число, не нарушая порядка расположения. Освободить память, занимаемую элементами списка.
13. Из списка удалить максимальный элемент, а затем вставить число, равное минимальному положительному элементу, перед минимальным элементом и после него. Освободить память, занимаемую элементами списка.
14. Из списка удалить элементы между первым и вторым элементами, модули которых меньше заданной величины, а затем расположить элементы в порядке возрастания модулей и вставить заданное число, не нарушая порядка расположения. Освободить память, занимаемую элементами списка.
15. Из списка удалить элемент, стоящий после максимального элемента, а затем перед последним нулевым элементом и после него вставить число, равное последнему отрицательному элементу списка. Освободить память, занимаемую элементами списка.
16. Из списка удалить максимальный элемент, а затем расположить элементы списка в порядке убывания модулей и вставить заданное число, не нарушая порядка расположения. Освободить память, занимаемую элементами списка.
17. Из списка удалить элементы, модули которых меньше заданной величины, затем вставить число, равное последнему положительному элементу, перед первым положительным элементом и после него. Освободить память, занимаемую элементами списка.
18. Из списка удалить элемент, стоящий перед первым нулевым элементом, а затем вставить заданное число перед каждым положительным элементом списка. Освободить память, занимаемую элементами списка.
19. Из списка удалить первые два нулевых элемента, а затем вставить заданное число перед первым положительным элементом и после последнего положительного элемента. Освободить память, занимаемую элементами списка.
20. Из списка удалить максимальный по модулю элемент, а затем вставить число, равное первому положительному элементу, перед последним отрицательным элементом и после него. Освободить память, занимаемую элементами списка.
21. Из списка удалить элементы, стоящие между первым и вторым положительными элементами, а затем вставить число, равное максимальному элементу списка, перед минимальным элементом и после него. Освободить память, занимаемую элементами списка.
22. Из списка удалить максимальный и минимальный элементы, а затем вставить заданное число перед первым отрицательным элементом и после последнего отрицательного элемента. Освободить память, занимаемую элементами списка.

23. Из списка удалить первые два положительных элемента, а затем вставить число, равное минимальному элементу списка, перед максимальным отрицательным элементом и после него. Освободить память, занимаемую элементами списка.
24. Из списка удалить первый отрицательный элемент, а затем вставить заданное число перед первым положительным элементом и после последнего положительного элемента. Освободить память, занимаемую элементами списка.
25. Из списка удалить второй отрицательный элемент, а затем вставить число, равное максимальному элементу списка, перед максимальным по модулю отрицательным элементом и после него. Освободить память, занимаемую элементами списка.
26. Из списка удалить элементы, стоящие между первым и вторым отрицательными элементами, а затем вставить заданное число перед максимальным элементом и после него. Освободить память, занимаемую элементами списка.
27. Из списка удалить последний отрицательный элемент, а затем вставить заданное число перед минимальным элементом списка и после минимального положительного элемента. Освободить память, занимаемую элементами списка.
28. Из списка удалить элементы, стоящие между первым и вторым нулевыми элементами, а затем вставить число, равное последнему элементу списка, перед первым отрицательным элементом и после него. Освободить память, занимаемую элементами списка.
29. Из списка удалить первый положительный элемент, а затем вставить заданное число перед максимальным по модулю элементом списка и после него. Освободить память, занимаемую элементами списка.
30. Из списка удалить первый и последний отрицательные элементы, а затем вставить число, равное второму положительному элементу, перед первым положительным элементом и после него. Освободить память, занимаемую элементами списка.

Опрос

Тема 1. Рекурсивные вычисления

1. Дайте определение рекурсии.
 2. Приведите примеры содержательных задач, где для решения может быть использован рекурсивный вспомогательный алгоритм.
 3. Что такое граничное условие и каково его назначение в рекурсивной подпрограмме?
 4. Охарактеризуйте формы рекурсивных процедур.
 5. Что такое рекурсивный спуск? Пример.
 6. Что такое рекурсивный подъём? Пример.
 7. Что такое глубина рекурсии? Пример.
 8. На каком этапе выполнения рекурсивной подпрограммы могут выполняться её операторы?
 9. Почему приведённый ниже алгоритм посимвольного формирования строки завершится аварийно?
- ```
Function Stroka : String;
Var C : Char;
Begin
 Write('Введите очередной символ: '); ReadLn(C);
 Stroka:=Stroka+C
End;
```
- На каком этапе выполняются действия в этом алгоритме?

### Тема 2. Основы визуального программирования в среде Delphi

1. Какие функции выполняет интегрированная среда программирования Delphi? Из каких частей она состоит?

2. Для чего служат визуальные компоненты? Как осуществляется управление визуальными компонентами?
3. Какие функции выполняет свойство компонента? Перечислите типы свойств.
4. Чем отличается работа со свойствами в режиме проектирования и выполнения программы?
5. Какие типы файлов используются в приложении Delphi?
6. Основное назначение каждого типа файлов в приложении Delphi?
7. Охарактеризуйте обобщенную структуру приложения Delphi.
8. Какие задачи решает Менеджер Проектов?
9. Для чего предназначены окна дерева объектов и инспектора объектов?
10. Перечислите панели инструментов, используемые для разработки программ Delphi.
11. Назначение главного окна Delphi.
12. Назначение конструктора формы.
13. Что такое редактор кода?
14. Объектный инспектор и дерево объектов, их назначение.
15. Насколько важен этап создания интерфейса приложения? Можно ли без него обойтись?
16. Какие средства являются основой визуального программирования в среде Delphi?
17. Как помещаются на форму и настраиваются компоненты?
18. Как создаются обработчики событий?
19. Как осуществляется компиляция и проверка функционирования проекта?
20. Как узнать состав и назначение файлов проекта?
21. Как установить параметры компиляции и сборки проекта?
22. Как создать модальную форму в Delphi - проекте?
23. Как установить основные свойства модальной формы в проекте?
24. Как установить основные события модальной формы в проекте?
25. Как создать немодальную форму в проекте?
26. Что такое управляющие элементы окна, какие они бывают и для чего служат?
27. Назовите общие свойства управляющих элементов.
28. Назовите общие события управляющих элементов.
29. Что такое Палитра Компонентов и как она используется?
30. Как осуществляется управление компонентами в приложении?
31. Как установить основные свойства компонента в приложении?
32. Как установить основные события главной формы в приложении?
33. Что такое компоненты меню, какие они бывают и для чего служат?
34. Что такое свойства пунктов меню, какие они бывают и для чего служат?
35. Что такое Дизайнер Меню, какие функции и как он выполняет?
36. Как создаются пункты меню?
37. Как обрабатываются команды пунктов меню?
38. Как осуществляется синхронизация событий одноименных пунктов главного и всплывающего меню в приложении?
39. Какие основные типы ошибок в программах Вы знаете?
40. Какие средства отладки программ существуют в среде Delphi?
41. Как включить режим отладки?
42. Как установить контрольную точку?
43. Как отладить программу без отладчика?
44. Как разместить компонент в форме?
45. Как изменить размер компонента?

#### Тема 4. Указатели. Динамическая память

1. Для чего предназначены переменные ссылочного (указательного) типа?
2. Как осуществляется выделение и освобождение динамической памяти?
3. Как осуществляется доступ к динамическим переменным?
4. Что называется указателем?
5. Для чего необходимо значение `nil` при работе с указателями?
6. Назовите операции, которые допускаются над значениями ссылочного типа?
7. Какие стандартные процедура реализуют основные действия над динамическими переменными?
8. К чему приводит «потеря» указателя на данные, хранимые в динамической памяти?
9. Ограничена ли динамическая память?
10. Как получить указатель на участок памяти, которую занимает объект?
11. Как получить значения указателей полей текущего диспетчера памяти?
12. Как получить указатель на заданный адрес памяти?
13. Как можно выделить на куче блок памяти заданного размера?
14. Как можно выделить блок памяти заданного размера?
15. Как можно изменить размер динамически распределенного блока памяти?
16. Как можно выполнить бинарное сравнение двух участков памяти?
17. Как можно узнать размер памяти, занимаемый переменной?
18. Как можно установить значения полей диспетчера памяти?
19. Как можно определить, используется в настоящий момент диспетчер памяти, установленный по умолчанию, или был установлен другой диспетчер?
20. Как можно создать динамическую переменную?
21. Как можно создать динамическую переменную, выделяя под нее заданный объем памяти?
22. Как можно инициализировать динамическую переменную?
23. Как можно перераспределяет память для динамической переменной?
24. Как можно деинициализировать динамическую переменную?
25. Как можно высвободить память из-под динамической переменной?

### Тестирование

#### Тема 5. Динамические структуры данных

1. Структуры данных делятся на:
  - 1) Файлы и данные статической структуры
  - 2) Данные статической структуры и данные динамической структуры
  - 3) Данные статической структуры и файлы
2. К простым данным статической структуры не относятся:
  - 1) записи
  - 2) файлы
  - 3) массивы
3. К данным стандартных типов не относятся:
  - 4) символ
  - 5) список
  - 6) логические
4. К арифметическим типам не относятся:
  - 1) Integer
  - 2) Boolean
  - 3) Real
5. Составные данные статической структуры делятся на:

- 1) однородные и простые
  - 2) однородные и неоднородные
  - 3) скалярные и неоднородные
6. К однородным типам не относятся:
- 1) строки
  - 2) записи
  - 3) множества
7. К неоднородным типам не относятся:
- 1) записи
  - 2) массивы
  - 3) объекты
8. К однородным типам не относят:
- 1) множества
  - 2) объекты
  - 3) строки
9. Данные динамической структуры делятся на:
- 1) Файлы, несвязанные динамические данные, составные данные
  - 2) Файлы, несвязанные динамические данные, связанные динамические данные
  - 3) Массивы, несвязанные динамические данные, связанные динамические данные
10. Данные, имеющие дополнительную связь между последним и первым элементами списка, - это:
- 1) очередь
  - 2) кольцевые списки
  - 3) стек
11. Частный случай линейного односвязного списка, для которого разрешены только два действия: добавление элемента в конец списка и удаление элемента из начала списка, - это:
- 1) кольцевой список
  - 2) очередь
  - 3) дек
12. Частный случай линейного односвязного списка, для которого разрешено добавлять или удалять элементы только с одного конца списка, который называется вершиной, - это:
- 1) очередь
  - 2) стек
  - 3) дек
13. Последовательность элементов, в которой новые данные могут быть присоединены с обеих сторон и также с обеих можно удалять ее элементы, - это:
- 1) стек
  - 2) дек
  - 3) очередь
14. Какую динамическую структуру можно представить в виде трубы с запаянным концом?
- 1) стек
  - 2) дерево
  - 3) очередь
  - 4) таблица
15. Укажите недопустимую операцию со стеком
- 1) проверить пуст стек или нет
  - 2) поменять два элемента стека местами
  - 3) сделать стек пустым
  - 4) взять элемент из стека
16. Какому закону подчиняются элементы стека?

- 1) последним вошел - первым вышел
  - 2) последним вошел - последним вышел
  - 3) первым вошел - первым вышел
  - 4) порядок входа и выхода произвольный
17. С применением какой структуры можно решать задачу о ханойских башнях?
- 1) очереди
  - 2) стека
  - 3) дерева
  - 4) таблицы
18. Последовательность элементов, образующаяся в порядке их поступления - это
- 1) очередь
  - 2) стек
  - 3) дек
  - 4) дерево
19. Какая структура данных подчиняется закону: первым вошел - первым вышел?
- 1) дерево
  - 2) стек
  - 3) очередь
  - 4) таблица
20. Как двигаются элементы очереди?
- 1) в одном направлении
  - 2) в противоположных направлениях
  - 3) произвольно
  - 4) элементы не двигаются
21. Укажите недопустимую операцию с очередями
- 1) добавить элемент в конец очереди
  - 2) сделать очередь пустой
  - 3) добавить элемент в начало очереди
  - 4) взять элемент из начала очереди
22. Укажите данные, имеющие иерархическую структуру
- 1) стек
  - 2) дерево
  - 3) очередь
  - 4) таблица
23. На какие структуры можно отображать стек?
- 1) вектор, список
  - 2) вектор, список, очередь
  - 3) вектор, очередь
  - 4) список, очередь, дерево
24. Разновидностью какой структуры являются структуры стек и очередь
- 1) вектор
  - 2) дерева
  - 3) линейного списка
25. Какое название имеет линейный список, в котором добавления и исключения возможны только в конце?
- 1) очередь
  - 2) стек
  - 3) дерево
  - 4) вектор

26. Какое название имеет линейный список, в котором элементы добавляются только в конец, а исключаются только из начала?

- 1) вектор
- 2) стек
- 3) очередь
- 4) дерево

27. Очередь, стек, дек, список относятся к...

- 1) кольцевым данным
- 2) линейным данным
- 3) разветвляющимся данным
- 4) несвязным данным

4.3 Промежуточная аттестация по дисциплине проводится в форме зачета

### **Типовые вопросы зачета (УК-1)**

1. Языки программирования: понятие, классификация. Сравнение языков программирования.
2. Системы программирования: понятие, виды. Трансляторы (компиляторы и интерпретаторы).
3. Этапы решения задач на ЭВМ. Техника редактирования алгоритмов и программ. Устранение синтаксических ошибок. Тестирование программ и устранение алгоритмических ошибок.
4. Понятие алгоритма, исполнителя алгоритма. Основные свойства алгоритмов.
5. Способы записи алгоритмов. Типовые алгоритмические структуры. Примеры.
6. Основные понятия языка Pascal: алфавит, лексемы, имена, служебные слова.
7. Структура программ на языке Pascal. Примеры.
8. Описания в языке Pascal: типы, переменные, константы. Запись арифметических и логических выражений в языке Pascal.
9. Данные. Переменные и константы. Типы данных. Стандартные и программируемые типы.
10. Операторы присвоения, ввода, вывода, комментария в языке Pascal: синтаксис, свойства и область применения.
11. Условный оператор и оператор выбора в языке Pascal: синтаксис, свойства и область применения.
12. Операторы цикла в языке Pascal: синтаксис, свойства, принцип работы и область применения.
13. Функции в языке Pascal: синтаксис, свойства и область применения.
14. Процедуры в языке Pascal: синтаксис, свойства и область применения.
15. Структуры данных: понятие, классификация, общая характеристика. Примеры.
16. Базовые типы данных языка Pascal (целые, вещественные, символьные, логические).
17. Перечислимый тип данных в языке Pascal. Ограниченный тип данных в языке Pascal.
18. Массивы в языке Pascal.
19. Множественный тип данных в языке Pascal.
20. Строковый тип в языке Pascal. Стандартные подпрограммы обработки строк. Примеры.
21. Комбинированный тип данных (записи) в языке Pascal. Примеры.
22. Файловый тип данных: числовые, текстовые файлы, файлы записей. Стандартные процедуры и функции обработки. Примеры.
23. Обзор методов сортировки. Сравнительный анализ.
24. Сортировка выбором. Алгоритм. Примеры.
25. Сортировка вставками. Простые вставки. Алгоритм. Примеры.
26. Сортировка вставками. Метод Шелла. Алгоритм. Примеры.
27. Сортировка обменом. Метод Пузырька. Алгоритм. Примеры.
28. Сортировка обменом. Метод Модифицированного пузырька. Алгоритм. Примеры.
29. Сортировка обменом. Метод Быстрой сортировки. Алгоритм. Примеры.

30. Сортировка методом Подсчета. Алгоритм. Примеры.
31. Сортировка методом слияния.
32. Обзор методов поиска информации. Сравнительный анализ.
33. Поиск минимального (максимального) элемента. Алгоритм. Примеры.
34. Поиск элемента с заданным значением. Поиск в неупорядоченной последовательности. Алгоритм. Примеры.
35. Поиск элемента с заданным значением. Поиск в упорядоченной последовательности. Дихотомический поиск. Алгоритм. Примеры.
36. Примеры алгоритмов решения задач.
37. Рекурсия. Реализация. Формы рекурсивных записей. Примеры. Задача о Ханойских башнях.
38. Визуальная среда программирования Delphi. Возможности, аппаратные и программные требования.
39. Организация проекта в Delphi. Основные файлы проектов.
40. Модули. Структура модулей. Типы модулей в Delphi: модули данных, модули динамических объектов, пакеты, модули потока команд.
41. Форма, свойства и события формы.
42. Компоненты ввода и отображения текстовой информации (Edit, Label, ListBox, StringGrid, Мемо и т.д.): назначение, свойства, методы.
43. Кнопки, индикаторы, управляющие элементы (Button, RadioGroup, RadioButton и т.д.): назначение, свойства, методы.
44. Создание и управление меню. Применение диалоговых компонентов. Примеры.
45. Стандартные события (события мыши, клавиатуры, системные события) визуальных компонент.
46. Графические возможности среды Delphi. Построение графических изображений.
47. Динамическая память. Виды памяти. Правила работы с динамической памятью. Операции NEW и DELETE. Динамические массивы.
48. Указатели, динамическое выделение памяти в языке Pascal.
49. Динамические структуры данных. Описание. Примеры.
50. Списки. Реализация списков. Примеры.
51. Абстрактный тип данных. Стеки. Реализация стеков (через массив, список). Примеры.
52. Абстрактный тип данных. Очереди. Реализация очереди (через массив, список). Примеры.
53. Абстрактный тип данных. Деки. Реализация дека. Примеры.

### **Типовые задания для зачета (УК-1)**

- 1 Разработать программу, содержащую процедуру сортировки последовательности по убыванию методом простых вставок.
- 2 Разработать программу, содержащую процедуру поиска элемента бинарным способом.
- 3 Дано вещественное  $X$ , целое  $N$ . Вычислить сумму:  $\sin(X) * \sin(3X) * \dots * \sin((2N-1)*X)$ .
- 4 Разработать программу, которая находит самую большую цифру целого числа.
- 5 Разработать программу, которая определяет количество вхождений данного символа в строку.
- 6 Разработать программу, которая выделяет из множества целых чисел от 1 до 30 множество чисел, кратных 2
- 7 Разработать программу, которая разбивает массив на два массива: в первом - отрицательные элементы из исходного массива, а во втором – положительные. Относительный порядок элементов сохранить.
- 8 Разработать программу построения графических изображений.
- 9 Разработать программу демонстрирующую работу с динамическими массивами.
- 10 Разработать программу демонстрирующую работу со списками.
- 11 Разработать программу демонстрирующую работу со стеками.

#### 4.4. Шкала оценивания промежуточной аттестации

| Оценка                          | Компетенции | Дескрипторы (уровни) – основные признаки освоения (показатели достижения результата) |
|---------------------------------|-------------|--------------------------------------------------------------------------------------|
| «зачтено»<br>(50 - 100 баллов)  | УК-1        |                                                                                      |
| «не зачтено»<br>(0 - 49 баллов) | УК-1        |                                                                                      |

### 5. Методические указания для обучающихся по освоению дисциплины (модуля)

#### 5.1 Методические указания по организации самостоятельной работы обучающихся:

Приступая к изучению дисциплины, в первую очередь обучающимся необходимо ознакомиться содержанием рабочей программы дисциплины (РПД), которая определяет содержание, объем, а также порядок изучения и преподавания учебной дисциплины, ее раздела, части.

Для самостоятельной работы важное значение имеют разделы «Объем и содержание дисциплины», «Учебно-методическое и информационное обеспечение дисциплины» и «Материально-техническое обеспечение дисциплины, программное обеспечение, профессиональные базы данных и информационные справочные системы».

В разделе «Объем и содержание дисциплины» указываются все разделы и темы изучаемой дисциплины, а также виды занятий и планируемый объем в академических часах.

В разделе «Учебно-методическое и информационное обеспечение дисциплины» указана рекомендуемая основная и дополнительная литература.

В разделе «Материально-техническое обеспечение дисциплины, программное обеспечение, профессиональные базы данных и информационные справочные системы» содержится перечень профессиональных баз данных и информационных справочных систем, необходимых для освоения дисциплины.

#### 5.2 Рекомендации обучающимся по работе с теоретическими материалами по дисциплине

При изучении и проработке теоретического материала необходимо:

- просмотреть еще раз презентацию лекции в системе MOODLe, повторить законспектированный на лекционном занятии материал и дополнить его с учетом рекомендованной дополнительной литературы;
- при самостоятельном изучении теоретической темы сделать конспект, используя рекомендованные в РПД источники, профессиональные базы данных и информационные справочные системы;
- ответить на вопросы для самостоятельной работы, по теме представленные в пункте 3.2 РПД.
- при подготовке к текущему контролю использовать материалы фонда оценочных средств (ФОС).

#### 5.3 Рекомендации по работе с научной и учебной литературой

Работа с основной и дополнительной литературой является главной формой самостоятельной работы и необходима при подготовке к устному опросу на семинарских занятиях, к дебатам, тестированию, экзамену. Она включает проработку лекционного материала и рекомендованных источников и литературы по тематике лекций.

Конспект лекции должен содержать реферативную запись основных вопросов лекции, в том числе с опорой на размещенные в системе MOODLe презентации, основных источников и литературы по темам, выводы по каждому вопросу. Конспект может быть выполнен в рамках распечатки выдачи презентаций лекций или в отдельной тетради по предмету. Он должен быть аккуратным, хорошо читаемым, не содержать не относящуюся к теме информацию или рисунки.

Конспекты научной литературы при самостоятельной подготовке к занятиям должны содержать ответы на каждый поставленный в теме вопрос, иметь ссылку на источник информации с обязательным указанием автора, названия и года издания используемой научной литературы. Конспект может быть опорным (содержать лишь основные ключевые позиции), но при этом позволяющим дать полный ответ по вопросу, может быть подробным. Объем конспекта определяется самим студентом.

В процессе работы с основной и дополнительной литературой студент может:

- делать записи по ходу чтения в виде простого или развернутого плана (создавать перечень основных вопросов, рассмотренных в источнике);
- составлять тезисы (цитирование наиболее важных мест статьи или монографии, короткое изложение основных мыслей автора);
- готовить аннотации (краткое обобщение основных вопросов работы);
- создавать конспекты (развернутые тезисы).

#### 5.4. Рекомендации по подготовке к отдельным заданиям текущего контроля

Собеседование предполагает организацию беседы преподавателя со студентами по вопросам практического занятия с целью более обстоятельного выявления их знаний по определенному разделу, теме, проблеме и т.п. Все члены группы могут участвовать в обсуждении, добавлять информацию, дискутировать, задавать вопросы и т.д.

Устный опрос может применяться в различных формах: фронтальный, индивидуальный, комбинированный. Основные качества устного ответа подлежащего оценке:

- правильность ответа по содержанию;
- полнота и глубина ответа;
- сознательность ответа;
- логика изложения материала;
- рациональность использованных приемов и способов решения поставленной учебной задачи;
- своевременность и эффективность использования наглядных пособий и технических средств при ответе;
- использование дополнительного материала;
- рациональность использования времени, отведенного на задание.

Устный опрос может сопровождаться презентацией, которая подготавливается по одному из вопросов практического занятия. При выступлении с презентацией необходимо обращать внимание на такие моменты как:

- содержание презентации: актуальность темы, полнота ее раскрытия, смысловое содержание, соответствие заявленной темы содержанию, соответствие методическим требованиям (цели, ссылки на ресурсы, соответствие содержания и литературы), практическая направленность, соответствие содержания заявленной форме, адекватность использования технических средств учебным задачам, последовательность и логичность презентуемого материала;
- оформление презентации: объем (оптимальное количество), дизайн (читаемость, наличие и соответствие графики и анимации, звуковое оформление, структурирование информации, соответствие заявленным требованиям), оригинальность оформления, эстетика, использование возможности программной среды, соответствие стандартам оформления;
- личностные качества: ораторские способности, соблюдение регламента, эмоциональность, умение ответить на вопросы, систематизированные, глубокие и полные знания по всем разделам программы;
- содержание выступления: логичность изложения материала, раскрытие темы, доступность изложения, эффективность применения средств ИКТ, способы и условия достижения результативности и эффективности для выполнения задач своей профессиональной или учебной деятельности, доказательность принимаемых решений, умение аргументировать свои заключения, выводы.

## 6. Учебно-методическое и информационное обеспечение дисциплины

### 6.1 Основная литература:

1. Программирование на языке Delphi : лабораторный практикум: в 2 ч., Ч.1. - Тамбов: Изд-во ТГУ, 2010. - 116 с.
2. Программирование на языке Delphi : лабораторный практикум : в 2 ч., Ч.2. - Тамбов: [Издат. дом ТГУ], 2011. - 192 с.
3. Программирование на языке Delphi : лабораторный практикум : в 2 ч., Ч.2. - Тамбов: [Издат. дом ТГУ], 2011. - 192 с.

4. Клыгина Е.В. Основы алгоритмизации и программирования для студентов-заочников специальностей "Математика", "Физика" : учеб. пособие. - Тамбов: [Издат. дом ТГУ им. Г.Р. Державина], 2013. - 152 с.

## 6.2 Дополнительная литература:

1. Клыгина Е.В., Кузнецов В.В., Кузьмина Н.В. Программирование на языке Паскаль. Ч.1 : Учеб. пособие. - Тамбов: ТГУ, 2004. - 112 с.
2. Бойцов Э.А. Алгоритмы и структуры данных : учеб. пособие для вузов. - Тамбов: Изд-во ТГУ, 2010. - 51 с.
3. Вирт, Никлаус Алгоритмы и структуры данных. - 2024-10-28; Алгоритмы и структуры данных. - Саратов: Профобразование, 2019. - 272 с. - Текст : электронный // IPR BOOKS [сайт]. - URL: <http://www.iprbookshop.ru/88753.html>
4. Сальников Ю.Н. Программирование. Базовый курс : учеб. пособ.. - М.: Маркет ДС, 2010. - 335 с.
5. Лопатин Д.В. Объектно-ориентированное программирование : Учебно-метод. пособие. - Тамбов: Изд-во ТГУ, 2001. - 77 с.

## 6.3 Иные источники:

1. Федеральный портал «Российское образование» - <http://www.edu.ru/>
2. Федеральное хранилище «Единая коллекция цифровых образовательных ресурсов» - <http://school-collection.edu.ru/>
3. Национальный Открытый Университет «ИНТУИТ» - <http://www.intuit.ru/>
4. Уроки Delphi начинающим с нуля - <http://www.delphi-manual.ru/>
5. Программирование - это просто - <http://easyprog.ru>
6. Языки программирования. Delphi. Уроки Delphi - <http://delphi-prg.ru/category/delphi-7-uroki>

## 7. Материально-техническое обеспечение дисциплины, программное обеспечение, профессиональные базы данных и информационные справочные системы

Для проведения занятий по дисциплине необходимо следующее материально-техническое обеспечение: учебные аудитории для проведения занятий лекционного и семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, помещения для самостоятельной работы.

Учебные аудитории и помещения для самостоятельной работы укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Помещения для самостоятельной работы укомплектованы компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду Университета.

Для проведения занятий лекционного типа используются наборы демонстрационного оборудования, обеспечивающие тематические иллюстрации (проектор, ноутбук, экран/ интерактивная доска).

Лицензионное и свободно распространяемое программное обеспечение:

Delphi 2007 for Win32 Professional

Microsoft Office Профессиональный плюс 2007

Профессиональные базы данных и информационные справочные системы:

1. Электронный каталог Фундаментальной библиотеки ТГУ. – URL: <https://www.tsutmb.ru/biblio/elektronnyij-katalog/>
2. Университетская библиотека онлайн: электронно-библиотечная система. – URL: <https://biblioclub.ru>
3. Консультант студента. Гуманитарные науки: электронно-библиотечная система. – URL: <https://www.studentlibrary.ru>

4. Научная электронная библиотека eLIBRARY.ru. – URL: <https://elibrary.ru>
5. Российская государственная библиотека: официальный сайт. – URL: <https://www.rsl.ru>
6. Российская национальная библиотека: официальный сайт. – URL: <http://nlr.ru>
7. Президентская библиотека имени Б.Н. Ельцина. – URL: <https://www.prlib.ru>
8. Научная электронная библиотека Российской академии естествознания. – URL: <https://www.monographies.ru>
9. Электронная библиотека РФФИ. – URL: <https://www.rfbr.ru/rffi/ru/library>

### **Электронная информационно-образовательная среда**

[https://auth.tsutmb.ru/authorize?response\\_type=code&client\\_id=moodle&state=xyz](https://auth.tsutmb.ru/authorize?response_type=code&client_id=moodle&state=xyz)

Взаимодействие преподавателя и студента в процессе обучения осуществляется посредством мультимедийных, гипертекстовых, сетевых, телекоммуникационных технологий, используемых в электронной информационно-образовательной среде университета.