

Федеральное государственное бюджетное образовательное учреждение высшего образования «Тамбовский государственный университет имени Г.Р. Державина»
Инженерно-технический институт
Кафедра функционального анализа

УТВЕРЖДАЮ:
И.о. директора института



Т. В. Пасько
«17» июня 2026 г.

РАБОЧАЯ ПРОГРАММА

по дисциплине Б1.В.11 Современные технологии программирования

Направление подготовки/специальность: 01.03.01 - Математика

Профиль/направленность/специализация: Искусственный интеллект и моделирование

Уровень высшего образования: бакалавриат

Квалификация: Бакалавр

год набора: 2026

Авторы программы:

Рыбаков Михаил Анатольевич

Кандидат физико-математических наук, Переславцева Оксана Николаевна

Рабочая программа составлена в соответствии с ФГОС ВО по направлению подготовки 01.03.01 - Математика (уровень бакалавриата) (приказ Министерства образования и науки РФ от «10» января 2018 г. № 8).

Рабочая программа принята на заседании Кафедры функционального анализа «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.....	5
3. Объем и содержание дисциплины.....	5
4. Контроль знаний обучающихся и типовые оценочные средства.....	11
5. Методические указания для обучающихся по освоению дисциплины (модуля).....	29
6. Учебно-методическое и информационное обеспечение дисциплины.....	30
7. Материально-техническое обеспечение дисциплины, программное обеспечение, профессиональные базы данных и информационные справочные системы.....	31

1. Цели и задачи дисциплины

1.1 Цель дисциплины – формирование компетенций:

ПК-7 Способен использовать систематические теоретические и практические знания для постановки и решения исследовательских задач в области образования

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 сфере общего, профессионального и дополнительного профессионального образования; в сфере научных исследований), 40 Сквозные виды профессиональной деятельности в промышленности (в сфере научно-исследовательских и опытноконструкторских разработок)

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
	ПК-7 Способен использовать систематические теоретические и практические знания для постановки и решения исследовательских задач в области образования	Оперировать математическими понятиями и категориями

1.5 Согласование междисциплинарных связей дисциплин, обеспечивающих освоение компетенций:

ПК-7 Способен использовать систематические теоретические и практические знания для постановки и решения исследовательских задач в области образования

№ п/п	Наименование дисциплин, определяющих междисциплинарные связи	Форма обучения			
		Очная (семестр)			
		1	3	4	7
1	PyTorch	+			
2	История математики	+			
3	Научно-исследовательская работа (получение первичных навыков научно-исследовательской работы)			+	
4	Основы программирования на C++	+			
5	Теория чисел		+		

6	Функционально-дифференциальные уравнения и включения				+
---	--	--	--	--	---

2. Место дисциплины в структуре ОП бакалавриата:

Дисциплина «Современные технологии программирования» относится к части, формируемой участниками образовательных отношений, учебного плана ОП по направлению подготовки 01.03.01 - Математика.

Дисциплина «Современные технологии программирования» изучается в 3 семестре.

3.Объем и содержание дисциплины

3.1.Объем дисциплины: 4 з.е.

Очная: 4 з.е.

Вид учебной работы	Очная (всего часов)
Общая трудоёмкость дисциплины	144
Контактная работа	48
Лекции (Лекции)	16
Лабораторные (Лаб. раб.)	32
Самостоятельная работа (СР)	60
Экзамен	36

3.2.Содержание курса:

№ темы	Название раздела/темы	Вид учебной работы, час.			Формы текущего контроля
		Лек ции	Лаб · раб.	СР	
		О	О	О	
3 семестр					
1	Основные понятия и классификация парадигм программирования	2	2	1	Тестирование; Лабораторная работа
2	История развития языков программирования	2	2	1	Тестирование; Лабораторная работа
3	Принципы ООП: инкапсуляция, наследование, полиморфизм.	2	4	2	Тестирование; Лабораторная работа
4	Применение ООП в современных языках программирования (C++, Java, Python и др.).	2	4	10	Тестирование; Лабораторная работа; Контрольная работа

5	Основные концепции функционального программирования .	2	4	12	Тестирование; Лабораторная работа
6	Языки функционального программирования : Haskell, Lisp, F# и др.	2	6	10	Тестирование; Лабораторная работа
7	Принципы логического программирования .	2	4	12	Тестирование; Лабораторная работа
8	Языки логического программирования : Prolog, Datalog и др.	2	6	12	Тестирование; Лабораторная работа; Контрольная работа

Тема 1. Основные понятия и классификация парадигм программирования (ПК-7)

Лекция.

Введение в парадигмы программирования, их основные характеристики и принципы. Обзор основных парадигм: процедурного, объектно-ориентированного, функционального и логического программирования. Сравнение различных подходов и их применение в разработке программного обеспечения. Роль парадигм в решении задач и построении алгоритмов.

Лабораторные работы.

Лабораторная работа №1

Задания:

- 1 1. Исследование и анализ: Проведите исследование на тему истории развития трех различных языков программирования (например, Fortran, C, Python). Подготовьте краткий отчет (2-3 страницы), включающий:
 - Историю создания каждого языка.
 - Основные особенности и ключевые нововведения каждого языка.
 - Влияние каждого языка на развитие программирования.
- 1 2. Практическая часть: Напишите небольшую программу на каждом из выбранных языков, выполняющую одну и ту же задачу (например, вычисление суммы элементов массива). Проанализируйте различия в синтаксисе и подходах.
- 2 3. Отчет: Подготовьте отчет с выводами о различиях и сходствах в подходах к программированию на различных языках.

Задания для самостоятельной работы.

Самостоятельная работа №1

Задания:

- Напишите эссе на тему "Парадигмы программирования и их классификация". Опишите основные парадигмы (процедурное, объектно-ориентированное, функциональное, логическое) и приведите примеры языков программирования, которые поддерживают каждую из них. Объясните, как каждая парадигма решает проблемы разработки программного обеспечения.

- Создайте таблицу, в которой перечислены основные парадигмы программирования. Для каждой парадигмы укажите ключевые характеристики, примеры языков и типичные случаи использования. Напишите краткий обзор применения каждой парадигмы в реальных проектах.

Тема 2. История развития языков программирования. (ПК-7)

Лекция.

1. История развития языков программирования. Лекция. Этапы развития языков программирования от первых языков (например, Assembler, Fortran) до современных решений (Java, Python). Влияние изменений в языках программирования на развитие технологий и программирования. Анализ ключевых языков и их особенностей, а также их вклад в развитие парадигм программирования.

Лабораторные работы.

Лабораторная работа №2

Задания:

1. Сравнение парадигм: Исследуйте и сравните три современные парадигмы программирования: процедурное, объектно-ориентированное и функциональное программирование. Подготовьте таблицу, в которой будут представлены ключевые особенности, преимущества и недостатки каждой парадигмы.
2. Практическая часть: Реализуйте одну и ту же задачу (например, управление списком студентов с возможностью добавления, удаления и поиска) с использованием:
 - Процедурного подхода на C.
 - Объектно-ориентированного подхода на Java или Python.
 - Функционального подхода на Haskell или F#.
3. Отчет: Напишите отчет, включающий сравнение подходов, сложности реализации и удобства использования каждой парадигмы.

Задания для самостоятельной работы.

Самостоятельная работа №2

Задания:

- Подготовьте хронологическую таблицу, которая включает ключевые вехи в истории развития языков программирования. Включите в таблицу такие языки, как Fortran, COBOL, C, C++, Java, Python и другие. Опишите основные особенности и вклад каждого языка в развитие программирования.
- Напишите сравнительный анализ двух языков программирования, которые появились в разное время (например, C и Python). Обсудите их историческое развитие, философию дизайна, основные особенности и влияние на развитие технологий.

Тема 3. Принципы ООП: инкапсуляция, наследование, полиморфизм. (ПК-7)

Лекция.

3. Принципы ООП: инкапсуляция, наследование, полиморфизм. Лекция. Основные принципы ООП: инкапсуляция (скрытие данных и реализация интерфейса), наследование (создание новых классов, на основе существующих) и полиморфизм (использование одного интерфейса для различных типов данных). Примеры реализации этих принципов и их значение для создания гибких и расширяемых программных систем.

Лабораторные работы.

Лабораторная работа №3

Задания:

1. Инкапсуляция:

Реализуйте класс BankAccount, который инкапсулирует данные о балансе счета и предоставляет методы для депозита и снятия средств. Убедитесь, что баланс можно изменять только через эти методы.

2. Наследование:

Создайте класс `SavingsAccount`, который наследует от `BankAccount` и добавляет метод `applyInterest`, который увеличивает баланс на процентный доход.

1 3. **Полиморфизм:**

Реализуйте метод `display()` в классе `Person`, который будет выводить информацию о человеке. Создайте подклассы `Student` и `Teacher`, которые переопределяют этот метод для отображения дополнительной информации.

Абстракция:

Создайте абстрактный класс `Shape`, который имеет абстрактные методы `area()` и `perimeter()`. Реализуйте два конкретных класса `Rectangle` и `Circle`, которые наследуют `Shape` и реализуют эти методы.

1 4. **Отчет:**

Подготовьте отчет, в котором опишите реализацию каждого из классов, приведите примеры использования и прокомментируйте, как реализованные принципы ООП помогают в организации кода.

Задания для самостоятельной работы.

Самостоятельная работа №3

Задания:

- Создайте диаграмму классов, иллюстрирующую принципы инкапсуляции, наследования и полиморфизма. Включите в диаграмму примеры классов и их взаимосвязей, используя любой язык программирования (например, UML диаграмма).
- Реализуйте программу на выбранном языке программирования, которая демонстрирует использование инкапсуляции, наследования и полиморфизма. Примером может быть система управления библиотекой, где есть базовый класс `Book`, и его подклассы `Ebook` и `PrintedBook`. Продемонстрируйте, как эти принципы применяются на практике.

Тема 4. Применение ООП в современных языках программирования (C++, Java, Python и др.).

(ПК-7)

Лекция.

Применение ООП в современных языках программирования (C++, Java, Python и др.). Лекция. Применение принципов ООП в различных языках программирования. Сравнение реализации ООП в C++, Java и Python, включая особенности синтаксиса, механизмов наследования, полиморфизма и инкапсуляции. Примеры реальных приложений и подходов в этих языках

Лабораторные работы.

Лабораторная работа №4

Задания:

1 1. **Создание класса и объектов:**

Реализуйте класс `Car` на C++, Java и Python, который будет иметь следующие атрибуты: `make` (марка), `model` (модель), `year` (год). Добавьте методы для отображения информации о машине и обновления года выпуска.

1 2. **Наследование:**

Создайте класс `ElectricCar`, который наследует от класса `Car` и добавляет атрибут `batteryCapacity` (емкость батареи) и метод `displayBattery()` для отображения емкости батареи.

1 3. **Полиморфизм:**

Реализуйте метод `startEngine()` в классах `Car` и `ElectricCar`. В `Car` метод должен выводить сообщение о запуске двигателя, а в `ElectricCar` — сообщение о включении системы батареи.

1 4. **Абстракция и интерфейсы:**

Реализуйте абстрактный класс `Shape` с абстрактными методами `area()` и `perimeter()`. Создайте конкретные классы `Rectangle` и `Circle`, которые реализуют эти методы.

1 5. **Отчет:**

Подготовьте отчет, описывающий реализацию классов и методов, приведение примеров их использования и анализ, как принципы ООП помогли организовать код и улучшить его структуру.

Задания для самостоятельной работы.

Самостоятельная работа №4

Задания:

- Напишите программу на C++, Java и Python, реализующую аналогичную функциональность (например, управление списком сотрудников). Убедитесь, что вы используете принципы ООП и продемонстрируйте, как один и тот же подход может быть реализован в разных языках.
- Проведите сравнение применения ООП в C++, Java и Python. Опишите особенности реализации ООП в каждом языке, такие как работа с классами, наследование, интерфейсы/абстрактные классы и управление памятью.

Тема 5. Основные концепции функционального программирования.

(ПК-7)

Лекция.

Тема №5. Основные концепции функционального программирования. Лекция. Основные принципы функционального программирования: неизменяемость данных, чистые функции, функции высшего порядка, рекурсия. Преимущества функционального подхода, такие как повышение надежности и легкость тестирования. Примеры применения функционального программирования для решения задач.

Лабораторные работы.

Лабораторная работа №5

Задания:

1. Изучение концепций: Изучите концепции чистых функций, каррирования и лямбда-выражений в функциональном программировании. Подготовьте краткое описание каждой концепции и ее преимуществ.
2. Практическая часть: Реализуйте несколько небольших программ на языке Haskell или F#, демонстрирующих:
 - Использование чистых функций.
 - Примеры каррирования.
 - Применение лямбда-выражений для обработки списков (например, фильтрация, маппинг).
3. Отчет: Подготовьте отчет с примерами кода и пояснениями к каждому из них.

Задания для самостоятельной работы.

Самостоятельная работа №5

Задания:

- Подготовьте обзор основных концепций функционального программирования, таких как функции высшего порядка, неизменяемость данных, рекурсия и чистые функции. Приведите примеры кода для каждой концепции на любом языке функционального программирования.
- Реализуйте программу на языке, поддерживающем функциональное программирование (например, Haskell или F#), которая демонстрирует использование функций высшего порядка и рекурсии. Объясните, как эти концепции помогают в решении задач.

Тема 6. Языки функционального программирования: Haskell, Lisp, F# и др.

(ПК-7)

Лекция.

Тема №6. Языки функционального программирования: Haskell, Lisp, F# и др. Лекция. Обзор языков функционального программирования: Haskell, Lisp, F#, и их особенности. Сравнение синтаксиса и подходов различных языков к реализации функциональных концепций. Примеры решения задач с использованием функциональных языков.

Лабораторные работы.

Лабораторная работа №6

Задания:

- 1 1. Сравнение языков: Исследуйте основные различия и сходства между Haskell, Lisp и F#. Подготовьте сравнительную таблицу, включающую:
 - Типизацию (статическая vs динамическая).
 - Модели вычислений (ленивые vs строгие).
 - Особенности синтаксиса.
- 1 2. Практическая часть: Реализуйте несколько простых программ на Haskell, Lisp и F#, выполняющих:
 - Рекурсивное вычисление факториала.
 - Функцию для вычисления чисел Фибоначчи.
 - Простую обработку списка (например, вычисление среднего значения).
- 1 3. Отчет: Составьте отчет, в котором опишите опыт работы с каждым из языков, сложности и преимущества каждого подхода.

Задания для самостоятельной работы.

Самостоятельная работа №6

Задания:

- Напишите краткий отчет о функциональном языке программирования, таком как Haskell, Lisp или F#. Опишите его ключевые особенности, синтаксис, и особенности работы с функциями. Включите примеры кода, иллюстрирующие основные возможности языка.
- Реализуйте одну и ту же функцию на двух различных функциональных языках (например, Haskell и Lisp). Включите описание кода и объяснение, как особенности каждого языка влияют на реализацию функции.

Тема 7. Принципы логического программирования.

(ПК-7)

Лекция.

Тема №7. Принципы логического программирования. Лекция. Основы логического программирования, включая принципы декларативного подхода, правила и факты. Как логическое программирование используется для решения задач, связанных с правилами и условиями. Примеры применения логического программирования в различных областях.

Лабораторные работы.

Лабораторная работа №7

Задания:

- 1 1. Изучение основ: Ознакомьтесь с основами логического программирования и языком Prolog. Подготовьте краткое резюме, объясняющее основные концепции, такие как факты, правила, запросы и унификация.
- 2 2. Практическая часть: На языке Prolog реализуйте базу знаний для семейного дерева, включающую:
 - Факты о людях (родители, дети, братья, сестры).
 - Правила для определения дедушек, бабушек, дядей, тетей и т.д.
 - Запросы для проверки ваших правил (например, найти всех дедушек конкретного человека).

- 1 3. Отчет: Подготовьте отчет с пояснениями, примерами запросов и их результатов.

Задания для самостоятельной работы.

Самостоятельная работа №7

Задания:

- Напишите обзор принципов логического программирования, таких как использование логических правил, фактов и вывода. Опишите, как логическое программирование отличается от других парадигм программирования.
- Реализуйте простую программу на языке логического программирования (например, Prolog), которая решает классическую задачу, такую как задача о поколениях (например, родственные отношения). Объясните, как логические правила используются для решения задачи.

Тема 8. Языки логического программирования: Prolog, Datalog и др.

(ПК-7)

Лекция.

Тема №8. Языки логического программирования: Prolog, Datalog и др. Лекция. Обзор языков логического программирования, таких как Prolog и Datalog. Основы синтаксиса и применения этих языков для решения задач, связанных с обработкой данных и выводом логических заключений. Примеры написания программ на Prolog и Datalog

Лабораторные работы.

Лабораторная работа №8

Задания:

- 1 1. Исследование возможностей Prolog: Изучите дополнительные возможности Prolog, такие как работа с арифметическими операциями, списками и рекурсией. Подготовьте примеры использования каждой возможности.
- 2 2. Практическая часть: Напишите программу на Prolog для решения задачи поиска маршрутов в графе (например, нахождение всех возможных путей между двумя городами). Включите в программу:
 - Определение узлов и ребер графа как фактов.
 - Правила для поиска маршрутов.
 - Запросы для поиска всех путей и кратчайшего пути.
- 1 3. Отчет: Оформите отчет, описывающий разработку программы и результаты ее выполнения.

Задания для самостоятельной работы.

Самостоятельная работа №8

Задания:

- Подготовьте обзор языка логического программирования Prolog. Опишите его основные синтаксические конструкции, особенности работы с фактами и правилами, и приведите примеры кода, иллюстрирующие работу языка.
- Реализуйте программу на Prolog, которая моделирует простую базу данных (например, база данных о студентах и курсах). Используйте факты и правила для выполнения запросов к базе данных и продемонстрируйте использование логического вывода.

4. Контроль знаний обучающихся и типовые оценочные средства

4.1. Распределение баллов:

3 семестр

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

Распределение баллов по заданиям:

№ те мы	Название темы / вид учебной работы	Формы текущего контроля / срезы	Мак. кол-во баллов	Методика проведения занятия и оценки
1.	Основные понятия и классификация парадигм программирования	Тестирование	2	2 балла за 100% верных отвеов
		Лабораторная работа	5	5 баллов за выполненную лабораторную работу
2.	История развития языков программирования.	Тестирование	1	1 балл за 100% верных ответов
		Лабораторная работа	5	5 баллов за выполнение лабораторной работы
3.	Принципы ООП: инкапсуляция, наследование, полиморфизм.	Тестирование	1	100% верных отвео - 1 балл
		Лабораторная работа	5	5 баллов - за выполнение лабораторной работы
4.	Применение ООП в современных языках программирования (C++, Java, Python и др.).	Тестирование	1	100% верных отвеов - 1 балл
		Лабораторная работа	5	5 баллов за выполнение лабораторной работы
		Контрольная работа(контрольный срез)	10	90 - 100% правильных отвеов - 10 баллов 70-89% правильных ответов - 8 баллов 50-69% правильных ответов - 5 баллов Менее 49% правильных отвеов - 0 балло
5.	Основные концепции функционального программирования.	Тестирование	1	100% верных ответов - 1 балл
		Лабораторная работа	5	5 баллов за выполнение лабораторной работы
6.	Языки функционального программирования: Haskell, Lisp, F# и др.	Тестирование	2	2 балла за 100% верных отвеов в тесте
		Лабораторная работа	5	За выполнение лабораторной работы начисляется 5 баллов
7.	Принципы логического программирования.	Тестирование	1	100% верных отвеов - 1 балл
		Лабораторная работа	5	За выполнение лабораторной работы начисляется 5 баллов
8.	Языки логического программирования: Prolog, Datalog и др.	Тестирование	1	100% верных отвеов - 1 балл
		Лабораторная работа	5	За выполнение лабораторной работы начисляется 5 баллов

	Контрольная работа(контрольный срез)	10	90 - 100% правильных ответов - 10 баллов 70-89% правильных ответов - 8 баллов 50-69% правильных ответов - 5 баллов Менее 49% правильных ответов - 0 баллов
9.	Итого за семестр	70	

Итоговая оценка по экзамену 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
85 - 100 баллов	Отлично
70 - 84 баллов	Хорошо
50 - 69 баллов	Удовлетворительно
Менее 50	Неудовлетворительно

4.2 Типовые оценочные средства текущего контроля

Контрольная работа

Тема 4. Применение ООП в современных языках программирования (C++, Java, Python и др.).

Контрольный срез №1

1 1. Что такое парадигма программирования?

- A) Конкретный язык программирования
- B) Подход к программированию, основанный на определённых концепциях и методах
- C) Алгоритм решения задачи
- D) Программа, написанная на любом языке программирования

Правильный ответ: B

1 2. Какие основные парадигмы программирования существуют?

- A) Объектно-ориентированное, логическое, процедурное, функциональное
- B) Математическое, научное, инженерное, управленческое
- C) Императивное, функциональное, логическое, декларативное
- D) Блочное, модульное, секвенциальное, асинхронное

Правильный ответ: A

1 3. Какой язык программирования считается первым?

- A) COBOL
- B) C
- C) Fortran
- D) Assembly

Правильный ответ: C

1 4. Какие ключевые изменения внесли языки третьего поколения (например, C, Pascal)?

- A) Появление сборщиков мусора
- B) Поддержка объектно-ориентированного программирования
- C) Введение абстракции данных и структур данных
- D) Работа с высокоуровневыми типами данных

Правильный ответ: C

1 5. Что такое инкапсуляция в ООП?

- A) Процесс создания объекта

- В) Ограничение доступа к компонентам объекта
- С) Наследование свойств и методов класса
- Д) Полиморфизм объектов

Правильный ответ: В

1 6. Что означает принцип полиморфизма?

- А) Способность объекта принимать множество форм
- В) Повторное использование кода
- С) Создание объектов на основе других объектов
- Д) Управление доступом к данным

Правильный ответ: А

1 7. Что такое наследование в контексте ООП?

- А) Передача методов и свойств от одного класса другому
- В) Создание независимых объектов
- С) Операция преобразования типов
- Д) Метод управления памятью

Правильный ответ: А

1 8. Какие языки программирования поддерживают объектно-ориентированное программирование?

- А) C++, Java, Python
- В) SQL, HTML, CSS
- С) Assembly, BASIC, COBOL
- Д) XML, JSON, YAML

Правильный ответ: А

1 9. Как реализуется инкапсуляция в C++?

- А) С помощью интерфейсов
- В) Через модификаторы доступа `private`, `protected`, `public`
- С) Путём использования статических переменных
- Д) С помощью глобальных переменных

Правильный ответ: В

1 10. Как в Python поддерживается наследование?

- А) С помощью функций
- В) Через классы и подклассы
- С) С использованием глобальных переменных
- Д) С помощью модулей

Правильный ответ: В

1 11. Что является основной концепцией функционального программирования?

- А) Использование циклов и условий
- В) Применение функций как объектов первого класса
- С) Использование указателей и ссылок
- Д) Управление доступом к данным

Правильный ответ: В

1 12. Что означает термин "чистая функция" в функциональном программировании?

- А) Функция, не изменяющая состояние программы
- В) Функция, работающая только с целыми числами
- С) Функция, не возвращающая значения
- Д) Функция, всегда возвращающая одинаковый результат для одинаковых входных данных

Правильный ответ: Д

1 13. Как в функциональном программировании реализуется управление состоянием?

- А) Через изменение глобальных переменных

- В) Через чистые функции и неизменяемые данные
- С) Через использование циклов и условных операторов
- Д) С помощью процедур

Правильный ответ: В

1 14. Что означает термин "функция высшего порядка"?

- А) Функция, использующая глобальные переменные
- В) Функция, которая принимает в качестве аргументов другие функции или возвращает функции
- С) Функция, написанная на высокоуровневом языке
- Д) Функция, работающая с массивами

Правильный ответ: В

1 15. Какая функция в Haskell является "чистой"?

- А) Функция, которая изменяет состояние программы
- В) Функция, которая работает с файлами
- С) Функция, которая вызывает побочные эффекты
- Д) Функция, которая возвращает одно и то же значение для одинаковых входных данных

Правильный ответ: Д

Тема 8. Языки логического программирования: Prolog, Datalog и др.

Контрольный срез №2

1 1. Какой из перечисленных языков является функциональным?

- А) Java
- В) Haskell
- С) C++
- Д) Python

Правильный ответ: В

1 2. Что такое лямбда-выражение в функциональном программировании?

- А) Глобальная переменная
- В) Анонимная функция
- С) Конструктор класса
- Д) Функция, вызывающая другой файл

Правильный ответ: В

1 3. Какая из следующих концепций является ключевой для Lisp?

- А) Управление памятью
- В) Рекурсия и работа со списками
- С) Наследование
- Д) Многопоточность

Правильный ответ: В

1 4. Как в F# реализуются функции высшего порядка?

- А) С помощью интерфейсов
- В) Через списки
- С) С использованием стандартных библиотек
- Д) С помощью передачи функций в качестве аргументов или возврата их в виде результата

Правильный ответ: Д

1 5. Что является основой логического программирования?

- А) Декларация фактов и правил
- В) Использование глобальных переменных
- С) Математические вычисления
- Д) Графическое представление данных

Правильный ответ: А

1 6. Какую роль выполняет механизм унификации в логическом программировании?

- А) Соединяет несколько программ в одну
- В) Определяет соответствие между терминами
- С) Выполняет арифметические операции
- Д) Управляет памятью

Правильный ответ: В

1 7. Что такое дедуктивная база данных в контексте логического программирования?

- А) База данных, хранящая текстовые документы
- В) База данных, поддерживающая логические выводы на основе заданных правил
- С) База данных, оптимизированная для быстрого поиска
- Д) База данных для графических изображений

Правильный ответ: В

1 8. Что такое факт в Prolog?

- А) Выражение, содержащее переменную
- В) Заранее определенное истинное утверждение
- С) Циклическая структура данных
- Д) Функция для ввода данных

Правильный ответ: В

1 9. Какое из следующих утверждений описывает правило в Prolog?

- А) Утверждение, которое всегда ложно
- В) Выражение, описывающее, когда факт является истинным
- С) Глобальная переменная
- Д) Вызов внешней библиотеки

Правильный ответ: В

1 10. Какая функция языка Prolog используется для поиска решения?

- А) eval()
- В) run()
- С) query()
- Д) solve()

Правильный ответ: С

1 11. Что такое "переменная" в логическом программировании?

- А) Элемент памяти, хранящий данные
- В) Маркер, который можно сопоставить с различными значениями в ходе выполнения программы
- С) Постоянная величина в коде
- Д) Название функции

Правильный ответ: В

1 12. Как в Datalog представляются факты и правила?

- А) Как SQL-запросы
- В) В виде логических выражений
- С) Через переменные и массивы
- Д) В виде объектно-ориентированных классов

Правильный ответ: В

1 13. Какой метод используется в Prolog для рекурсивного поиска решений?

- А) Жадный поиск
- В) Поиск с возвратом (backtracking)
- С) Поиск в ширину
- Д) Динамическое программирование

Правильный ответ: В

1 14. Как в Prolog происходит обработка отрицательных утверждений?

- A) С помощью функции abs()
- B) Через отрицание в query
- C) Посредством принципа "отрицание как отказ" (negation as failure)
- D) С использованием дополнительных библиотек

Правильный ответ: С

1 15. Какой оператор используется в Prolog для объединения фактов и правил?

- A) +
- B) .
- C) :-
- D) =

Правильный ответ: С

Лабораторная работа

Тема 3. Принципы ООП: инкапсуляция, наследование, полиморфизм.

Лабораторная работа №3

Задания:

1 1. Инкапсуляция:

Реализуйте класс BankAccount, который инкапсулирует данные о балансе счета и предоставляет методы для депозита и снятия средств. Убедитесь, что баланс можно изменять только через эти методы.

1 2. Наследование:

Создайте класс SavingsAccount, который наследует от BankAccount и добавляет метод applyInterest, который увеличивает баланс на процентный доход.

1 3. Полиморфизм:

Реализуйте метод display() в классе Person, который будет выводить информацию о человеке. Создайте подклассы Student и Teacher, которые переопределяют этот метод для отображения дополнительной информации.

Абстракция:

Создайте абстрактный класс Shape, который имеет абстрактные методы area() и perimeter(). Реализуйте два конкретных класса Rectangle и Circle, которые наследуют Shape и реализуют эти методы.

1 4. Отчет:

Подготовьте отчет, в котором опишите реализацию каждого из классов, приведите примеры использования и прокомментируйте, как реализованные принципы ООП помогают в организации кода.

Лабораторная работа

Тема 1. Основные понятия и классификация парадигм программирования

Лабораторная работа №1

Задания:

1 1. Исследование и анализ: Проведите исследование на тему истории развития трех различных языков программирования (например, Fortran, C, Python). Подготовьте краткий отчет (2-3 страницы), включающий:

- Историю создания каждого языка.
- Основные особенности и ключевые нововведения каждого языка.
- Влияние каждого языка на развитие программирования.

- 1 2. Практическая часть: Напишите небольшую программу на каждом из выбранных языков, выполняющую одну и ту же задачу (например, вычисление суммы элементов массива). Проанализируйте различия в синтаксисе и подходах.
- 2 3. Отчет: Подготовьте отчет с выводами о различиях и сходствах в подходах к программированию на различных языках.

Тема 2. История развития языков программирования.

Лабораторная работа №2

Задания:

- 1 1. Сравнение парадигм: Исследуйте и сравните три современные парадигмы программирования: процедурное, объектно-ориентированное и функциональное программирование. Подготовьте таблицу, в которой будут представлены ключевые особенности, преимущества и недостатки каждой парадигмы.
- 2 2. Практическая часть: Реализуйте одну и ту же задачу (например, управление списком студентов с возможностью добавления, удаления и поиска) с использованием:
 - Процедурного подхода на C.
 - Объектно-ориентированного подхода на Java или Python.
 - Функционального подхода на Haskell или F#.
- 1 3. Отчет: Напишите отчет, включающий сравнение подходов, сложности реализации и удобства использования каждой парадигмы.

Тема 4. Применение ООП в современных языках программирования (C++, Java, Python и др.).

Лабораторная работа №4

Задания:

- 1 1. Создание класса и объектов:
Реализуйте класс Car на C++, Java и Python, который будет иметь следующие атрибуты: make (марка), model (модель), year (год). Добавьте методы для отображения информации о машине и обновления года выпуска.
- 1 2. Наследование:
Создайте класс ElectricCar, который наследует от класса Car и добавляет атрибут batteryCapacity (емкость батареи) и метод displayBattery() для отображения емкости батареи.
- 1 3. Полиморфизм:
Реализуйте метод startEngine() в классах Car и ElectricCar. В Car метод должен выводить сообщение о запуске двигателя, а в ElectricCar — сообщение о включении системы батареи.
- 1 4. Абстракция и интерфейсы:
Реализуйте абстрактный класс Shape с абстрактными методами area() и perimeter(). Создайте конкретные классы Rectangle и Circle, которые реализуют эти методы.
- 1 5. Отчет:
Подготовьте отчет, описывающий реализацию классов и методов, приведение примеров их использования и анализ, как принципы ООП помогли организовать код и улучшить его структуру.

Тема 5. Основные концепции функционального программирования.

Лабораторная работа №5

Задания:

- 1 1. Изучение концепций: Изучите концепции чистых функций, каррирования и лямбда-выражений в функциональном программировании. Подготовьте краткое описание каждой концепции и ее преимуществ.

- 2 2. Практическая часть: Реализуйте несколько небольших программ на языке Haskell или F#, демонстрирующих:
 - Использование чистых функций.
 - Примеры каррирования.
 - Применение лямбда-выражений для обработки списков (например, фильтрация, маппинг).
- 1 3. Отчет: Подготовьте отчет с примерами кода и пояснениями к каждому из них.

Тема 6. Языки функционального программирования: Haskell, Lisp, F# и др.

Лабораторная работа №6

Задания:

- 1 1. Сравнение языков: Исследуйте основные различия и сходства между Haskell, Lisp и F#. Подготовьте сравнительную таблицу, включающую:
 - Типизацию (статическая vs динамическая).
 - Модели вычислений (ленивые vs строгие).
 - Особенности синтаксиса.
- 1 2. Практическая часть: Реализуйте несколько простых программ на Haskell, Lisp и F#, выполняющих:
 - Рекурсивное вычисление факториала.
 - Функцию для вычисления чисел Фибоначчи.
 - Простую обработку списка (например, вычисление среднего значения).
- 1 3. Отчет: Составьте отчет, в котором опишите опыт работы с каждым из языков, сложности и преимущества каждого подхода.

Тема 7. Принципы логического программирования.

Лабораторная работа №7

Задания:

- 1 1. Изучение основ: Ознакомьтесь с основами логического программирования и языком Prolog. Подготовьте краткое резюме, объясняющее основные концепции, такие как факты, правила, запросы и унификация.
- 2 2. Практическая часть: На языке Prolog реализуйте базу знаний для семейного дерева, включающую:
 - Факты о людях (родители, дети, братья, сестры).
 - Правила для определения дедушек, бабушек, дядей, тетей и т.д.
 - Запросы для проверки ваших правил (например, найти всех дедушек конкретного человека).

Отчет: Подготовьте отчет с пояснениями, примерами запросов и их результатов.

Тема 8. Языки логического программирования: Prolog, Datalog и др.

Лабораторная работа №8

Задания:

- 1 1. Исследование возможностей Prolog: Изучите дополнительные возможности Prolog, такие как работа с арифметическими операциями, списками и рекурсией. Подготовьте примеры использования каждой возможности.

- 2 2. Практическая часть: Напишите программу на Prolog для решения задачи поиска маршрутов в графе (например, нахождение всех возможных путей между двумя городами). Включите в программу:
- Определение узлов и ребер графа как фактов.
 - Правила для поиска маршрутов.
 - Запросы для поиска всех путей и кратчайшего пути.
- 1 3. Отчет: Оформите отчет, описывающий разработку программы и результаты ее выполнения.

Тестирование

Тема 1. Основные понятия и классификация парадигм программирования

Тест №1

- 1 1. Что такое парадигма программирования?

- A) Метод проектирования алгоритмов
- B) Подход к программированию, основанный на определённых концепциях и методах
- C) Система управления версиями кода
- D) Инструмент для автоматизации тестирования

Правильный ответ: B

- 1 2. Какая из следующих парадигм программирования основана на понятии "функция как объект первого класса"?

- A) Императивная
- B) Логическая
- C) Функциональная
- D) Объектно-ориентированная

Правильный ответ: C

- 1 3. Какую роль играет инкапсуляция в объектно-ориентированном программировании?

- A) Скрытие реализации и управление доступом к данным
- B) Описание последовательности шагов для выполнения задачи
- C) Определение логических отношений между данными
- D) Оптимизация времени выполнения программ

Правильный ответ: A

- 1 4. К какой парадигме программирования относится использование предикатов и логических правил?

- A) Императивная
- B) Объектно-ориентированная
- C) Функциональная
- D) Логическая

Правильный ответ: D

- 1 5. Что является характерной чертой декларативного программирования?

- A) Описание того, что должно быть сделано, а не как это должно быть сделано
- B) Определение пошаговых инструкций для выполнения программы
- C) Использование циклов и условных операторов
- D) Прямое управление памятью

Правильный ответ: A

Тема 2. История развития языков программирования.

Тест №2

- 1 1. Какой язык программирования считается первым в истории?

- A) Fortran
- B) Assembly
- C) COBOL
- D) Ada

Правильный ответ: A

- 1 2. Какой язык программирования был разработан для решения научных и инженерных задач в 1957 году?

- A) COBOL
- B) BASIC
- C) Fortran
- D) Lisp

Правильный ответ: C

- 1 3. Какой из следующих языков программирования был создан специально для работы с искусственным интеллектом?

- A) Pascal
- B) C
- C) Lisp
- D) Java

Правильный ответ: C

- 1 4. Какая концепция была впервые введена в язык программирования Algol в 1960-х годах?

- A) Объектно-ориентированное программирование
- B) Структурное программирование и блочная структура
- C) Логическое программирование
- D) Программирование параллельных процессов

Правильный ответ: B

- 1 5. Какой язык программирования был разработан для коммерческих и бизнес-приложений в 1960 году?

- A) COBOL
- B) Python
- C) Ruby
- D) C++

Правильный ответ: A

Тема 3. Принципы ООП: инкапсуляция, наследование, полиморфизм.

Тест №3

- 1 1. Что такое инкапсуляция в объектно-ориентированном программировании?

- A) Создание нового класса на основе существующего
- B) Скрытие внутренней реализации объекта и предоставление интерфейса для взаимодействия с ним
- C) Возможность одного метода выполнять разные задачи
- D) Использование наследования для создания новых классов

Правильный ответ: B

- 1 2. Какое из следующих утверждений правильно описывает наследование в ООП?

- A) Процесс создания новых объектов
- B) Механизм, позволяющий одному классу использовать методы и поля другого класса
- C) Использование интерфейсов для создания полиморфизма
- D) Объединение нескольких объектов в один

Правильный ответ: B

1 3. Что означает термин "полиморфизм" в контексте ООП?

- A) Способность функции обрабатывать данные различных типов
- B) Возможность создания объектов, которые не связаны с классами
- C) Использование нескольких классов в одном методе
- D) Скрытие данных от других классов

Правильный ответ: A

1 4. Каким образом наследование способствует повторному использованию кода?

- A) Позволяет объектам менять свои данные в зависимости от времени выполнения
- B) Позволяет создавать новые классы, которые автоматически получают свойства и методы существующих классов
- C) Позволяет скрыть реализацию метода от пользователя
- D) Обеспечивает возможность перегрузки операторов

Правильный ответ: B

1 5. Какой из следующих примеров лучше всего иллюстрирует полиморфизм?

- A) Класс "Птица" с методами "летать" и "петь"
- B) Метод "draw" для объектов типа "Круг" и "Квадрат", которые выполняют разные действия при вызове
- C) Класс "Животное" и подклассы "Кошка" и "Собака"
- D) Класс с приватными и публичными полями

Правильный ответ: B

Тема 4. Применение ООП в современных языках программирования (C++, Java, Python и др.).

Тест №4

1 1. Как в языке C++ определяется класс?

- A) `define ClassName {}`
- B) `class ClassName {};`
- C) `new class ClassName {}`
- D) `create ClassName {}`

Правильный ответ: B

1 2. Какой ключевое слово в Java используется для создания экземпляра класса?

- A) `class`
- B) `new`
- C) `instance`
- D) `object`

Правильный ответ: B

1 3. Какой метод в Python является конструктором класса?

- A) `__start__()`
- B) `__init__()`
- C) `__construct__()`
- D) `__new__()`

Правильный ответ: B

1 4. Какой принцип ООП позволяет методу с тем же именем выполнять разные функции в Java?

- A) Инкапсуляция
- B) Полиморфизм
- C) Наследование
- D) Абстракция

Правильный ответ: B

1 5. Что произойдет, если в C++ не реализован виртуальный деструктор в базовом классе при удалении объекта через указатель на базовый класс?

- A) Ничего, программа будет работать нормально
- B) Произойдет утечка памяти, так как деструктор производного класса не будет вызван
- C) Компилятор выдаст ошибку
- D) Все деструкторы будут вызваны автоматически

Правильный ответ: B

Тема 5. Основные концепции функционального программирования.

Тест №5

1 1. Что такое чистая функция в функциональном программировании?

- A) Функция, которая вызывает побочные эффекты
- B) Функция, которая всегда возвращает одинаковый результат для одних и тех же входных значений и не имеет побочных эффектов
- C) Функция, которая изменяет глобальное состояние
- D) Функция, использующая циклы и условные операторы

Правильный ответ: B

1 2. Какую роль выполняет функция "map" в функциональном программировании?

- A) Создает копию списка
- B) Применяет данную функцию к каждому элементу списка и возвращает новый список
- C) Объединяет несколько списков в один
- D) Удаляет дубликаты из списка

Правильный ответ: B

1 3. Что такое "каррирование" в функциональном программировании?

- A) Способ объединения нескольких функций в одну
- B) Техника преобразования функции с множественными аргументами в последовательность функций с одним аргументом
- C) Процесс создания локальных переменных внутри функции
- D) Метод сортировки списка

Правильный ответ: B

1 4. Что означает термин "высшего порядка функция" в функциональном программировании?

- A) Функция, которая принимает другую функцию в качестве аргумента или возвращает функцию в качестве результата
- B) Функция, которая вызывается в конце программы
- C) Функция, которая выполняет сложные математические операции
- D) Функция, которая не принимает аргументы

Правильный ответ: A

1 5. Какую из следующих операций выполняет функция "reduce" в функциональном программировании?

- A) Разделяет список на подсписки
- B) Применяет функцию к парам элементов из списка для получения единого результата
- C) Удаляет последний элемент из списка
- D) Сортирует список в порядке убывания

Правильный ответ: B

Тема 6. Языки функционального программирования: Haskell, Lisp, F# и др.

Тест №6

1 1. Какую особенность имеет язык Haskell, отличающую его от других языков программирования?

- A) Императивный подход к программированию
- B) Ленивое вычисление (Lazy evaluation)
- C) Использование статических типов данных
- D) Сильная поддержка объектно-ориентированного программирования

Правильный ответ: B

1 2. Какая структура данных является центральной в языке Lisp?

- A) Массив
- B) Список
- C) Дерево
- D) Очередь

Правильный ответ: B

1 3. Каким образом в F# можно объявить функцию?

- A) `function myFunc() {...}`
- B) `def myFunc = {...}`
- C) `let myFunc x = ...`
- D) `sub myFunc {...}`

Правильный ответ: C

1 4. Какой язык функционального программирования является строго типизированным и поддерживает вывод типов?

- A) Lisp
- B) Haskell
- C) JavaScript
- D) Python

Правильный ответ: B

1 5. Что означает концепция "иммутабельность данных" в функциональных языках программирования, таких как Haskell и F#?

- A) Данные могут быть изменены в любой момент
- B) Данные никогда не могут быть изменены после их создания
- C) Данные автоматически сохраняются в базу данных
- D) Данные могут быть изменены только в глобальной области видимости

Правильный ответ: B

Тема 7. Принципы логического программирования.

Тест №7

1 1. Какой основной принцип лежит в основе логического программирования?

- A) Построение программ как последовательности инструкций
- B) Описание логических отношений и использование вывода для получения решений
- C) Определение функций и их применение к данным
- D) Использование классов и объектов для моделирования реальных сущностей

Правильный ответ: B

1 2. Какой язык программирования является наиболее известным примером логического программирования?

- A) Python
- B) Java
- C) Prolog
- D) C++

Правильный ответ: С

1 3. Что такое факт в логическом программировании?

- A) Утверждение, которое истинно и не требует дополнительного доказательства
- B) Функция, которая возвращает логическое значение
- C) Массив данных, используемый в программе
- D) Инструкция для выполнения циклов

Правильный ответ: А

1 4. Какой метод используется в Prolog для поиска решений, когда возможны несколько путей?

- A) Поиск в глубину
- B) Поиск с возвратом (backtracking)
- C) Жадный алгоритм
- D) Динамическое программирование

Правильный ответ: В

1 5. Что такое правило в логическом программировании?

- A) Логическое условие, которое при выполнении приводит к определенному результату
- B) Инструкция для выполнения арифметических операций
- C) Определение переменной с конкретным значением
- D) Метод, который вычисляет результат на основе входных данных

Правильный ответ: А

Тема 8. Языки логического программирования: Prolog, Datalog и др.

Тест №8

1 1. Какой язык логического программирования позволяет описывать знания и делать выводы на основе этих знаний, используя факты и правила?

- A) Java
- B) Prolog
- C) Python
- D) C#

Правильный ответ: В

1 2. Какой из следующих языков логического программирования предназначен для обработки и анализа данных с использованием декларативных запросов?

- A) Haskell
- B) SQL
- C) Datalog
- D) Lisp

Правильный ответ: С

1 3. Что такое "квантор существования" в контексте Prolog?

- A) Квантор, который утверждает, что существуют некоторые объекты, удовлетворяющие определенным условиям
- B) Квантор, который применяется к переменным для определения их значений
- C) Квантор, который определяет количество объектов в базе данных
- D) Квантор, который гарантирует отсутствие ошибок в коде

Правильный ответ: А

1 4. Какой из следующих элементов является базовой конструкцией в языке Prolog для представления фактов и правил?

- A) Функции
- B) Классы

С) Предикаты

D) Массивы

Правильный ответ: С

- 1 5. Какой механизм используется в Datalog для выполнения запросов к базе данных, чтобы найти данные, соответствующие заданным условиям?

A) Линейный поиск

B) Деревья поиска

С) Рекурсивные запросы

D) Хеширование

Правильный ответ: С

4.3 Промежуточная аттестация по дисциплине проводится в форме экзамена

Типовые вопросы экзамена (ПК-7)

Типовые вопросы для экзамена

- 1 1. Определите и сравните основные парадигмы программирования: процедурную, объектно-ориентированную, функциональную и логическую.
- 2 2. Какие преимущества и недостатки имеет процедурное программирование по сравнению с объектно-ориентированным программированием?
- 3 3. Приведите примеры языков программирования, поддерживающих каждую из перечисленных парадигм. Каковы особенности этих языков?
- 4 4. Что такое парадигма программирования и как она влияет на структуру и организацию кода?
- 5 5. Объясните, как использование различных парадигм программирования может повлиять на процесс разработки программного обеспечения.
- 6 6. Опишите ключевые этапы в развитии языков программирования с момента появления первых языков до современных языков.
- 7 7. Какое значение для истории языков программирования имели такие языки, как Fortran, COBOL и C?
- 8 8. Как эволюция языков программирования от процедурного к объектно-ориентированному и функциональному программированию повлияла на разработку ПО?
- 9 9. Что такое "новые" и "старые" языки программирования и как они дополняют друг друга?
- 10 10. Какие события или достижения в истории программирования оказали наибольшее влияние на современные технологии?
- 11 11. Объясните принцип инкапсуляции и его важность в объектно-ориентированном программировании. Приведите примеры.
- 12 12. Как наследование помогает в повторном использовании кода? Приведите пример использования наследования в конкретном языке программирования.
- 13 13. Что такое полиморфизм и как он реализуется в объектно-ориентированных языках? Приведите примеры.
- 14 14. Как инкапсуляция, наследование и полиморфизм способствуют созданию масштабируемого и поддерживаемого кода?
- 15 15. Сравните реализацию инкапсуляции, наследования и полиморфизма в двух различных объектно-ориентированных языках (например, C++ и Java).
- 16 16. Каковы основные особенности ООП в C++? Как C++ реализует наследование и полиморфизм?
- 17 17. Какие отличия и сходства в реализации ООП существуют между Java и Python?

- 18 18. Как в современных языках программирования реализуются абстрактные классы и интерфейсы? Приведите примеры на Java и Python.
- 19 19. Какие методы и практики используются для обеспечения инкапсуляции и абстракции в C++ и Java?
- 20 20. Как управление памятью в объектно-ориентированных языках влияет на разработку программного обеспечения?
- 21 21. Опишите основные концепции функционального программирования, такие как функции высшего порядка, неизменяемость данных и чистые функции.
- 22 22. Какие преимущества предоставляет использование функционального программирования в сравнении с процедурным программированием?
- 23 23. Как рекурсия используется в функциональном программировании и какие преимущества она имеет по сравнению с итерацией?
- 24 24. Что такое ленивая оценка (lazy evaluation) и как она применяется в функциональных языках программирования?
- 25 25. Объясните, как функциональное программирование решает проблемы побочных эффектов.
- 26 26. Какие ключевые особенности языка Haskell отличают его от других языков программирования? Приведите примеры его применения.
- 27 27. Как язык Lisp поддерживает концепции функционального программирования? Приведите примеры кода на Lisp.
- 28 28. В чем особенности и преимущества языка F# по сравнению с другими функциональными языками?
- 29 29. Объясните, как различные функциональные языки реализуют концепции, такие как карринг и частичное применение функций.
- 30 30. Сравните возможности и особенности двух языков функционального программирования (например, Haskell и F#).
- 31 31. Объясните, что такое логическое программирование и как оно отличается от других парадигм программирования.
- 32 32. Каковы основные компоненты логического программирования, такие как факты, правила и вывод?
- 33 33. Опишите, как решаются задачи с использованием логического программирования. Приведите пример.
- 34 34. Как логическое программирование используется для решения задач, связанных с базами данных и поиском информации?
- 35 35. Как логическое программирование справляется с неопределенностью и неполнотой данных?
- 36 36. Опишите основные концепции и синтаксис языка Prolog. Приведите примеры фактов, правил и запросов.
- 37 37. Как язык Datalog отличается от Prolog? Какие задачи лучше решать с помощью Datalog?
- 38 38. Как язык Prolog используется для построения систем экспертных знаний? Приведите пример.
- 39 39. Что такое резольвенты и как они используются в логическом программировании?
- 40 40. Как логическое программирование применяется в области искусственного интеллекта и обработки естественного языка?

Типовые задания для экзамена (ПК-7)

Типовые задания для экзамена

- 1 1. Реализуйте простую программу на процедурном языке (например, C) и на функциональном языке (например, Haskell) для вычисления факториала числа. Объясните, как различаются подходы к решению одной и той же задачи в разных парадигмах.

- 2 2. Напишите программу на объектно-ориентированном языке (например, Java), которая моделирует простую библиотеку. В программе должны быть классы для книг и читателей, а также методы для добавления книг, регистрации читателей и выдачи книг. Опишите, как принципы ООП (инкапсуляция, наследование, полиморфизм) применяются в вашей реализации.
- 3 3. Создайте временную шкалу, на которой отображены ключевые события в истории развития языков программирования. Включите информацию о важных языках, таких как Fortran, COBOL, C, C++, Java, Python, и их влиянии на развитие технологий.
- 4 4. Напишите программу на языке программирования первой волны (например, Fortran), решающую задачу нахождения простых чисел в диапазоне. Затем перепишите эту программу на современном языке (например, Python) и объясните, какие улучшения и упрощения вы смогли сделать.
- 5 5. Реализуйте систему для управления автомобилями, включая базовый класс Vehicle и его подклассы Car и Truck. Класс Vehicle должен иметь методы для получения информации о транспортном средстве, а подклассы должны расширять функциональность и добавлять специфичные методы. Продемонстрируйте использование инкапсуляции, наследования и полиморфизма в вашей реализации.
- 6 6. Разработайте программу для учета сотрудников в компании, где используются принципы ООП. Создайте базовый класс Employee и два подкласса Manager и Engineer. Реализуйте методы для расчета зарплаты и вывода информации о сотрудниках, применяя полиморфизм.
- 7 7. Реализуйте класс для управления задачами (To-Do List) на C++ и Python. Класс должен поддерживать операции добавления, удаления и отображения задач. Объясните, как особенности языка программирования влияют на реализацию классов и методов.
- 8 8. Создайте приложение для учета книг в библиотеке на Java. Реализуйте классы Book, Author, и Library, используя принципы ООП. Продемонстрируйте, как использование Java влияет на организацию кода и работу с объектами.
- 9 9. Напишите функцию на языке функционального программирования (например, Haskell), которая принимает список чисел и возвращает список их квадратов. Объясните использование функций высшего порядка и рекурсии в вашем решении.
- 10 10. Реализуйте программу на языке, поддерживающем функциональное программирование (например, F#), для вычисления чисел Фибоначчи с использованием мемоизации. Опишите, как это улучшает производительность по сравнению с традиционным рекурсивным решением.
- 11 11. Напишите программу на Haskell, которая реализует функцию для вычисления факториала числа с использованием хвостовой рекурсии. Объясните, как хвостовая рекурсия улучшает производительность.
- 12 12. Реализуйте программу на Lisp, которая определяет и использует несколько функций для обработки списков (например, функции для фильтрации и трансформации списков). Продемонстрируйте использование особенностей Lisp для решения задач.
- 13 13. Напишите программу на Prolog, которая моделирует семейные отношения, включая факты и правила для определения родства (например, parent, sibling, ancestor). Продемонстрируйте, как Prolog решает запросы на основе логических правил.
- 14 14. Реализуйте простую программу на Datalog, которая выполняет запросы к базе данных о студентах и курсах. Опишите, как Datalog позволяет решать задачи, связанные с базами данных и логическим выводом.
- 15 15. Создайте Prolog-программу, которая решает задачу о шахматных фигурах, таких как проверки на допустимость движений. Объясните, как логическое программирование применяется для решения задач, связанных с правилами и условиями.

Напишите программу на Datalog для управления задачами в проекте. Реализуйте факты и правила для планирования задач и поиска зависимостей. Опишите, как логическое программирование помогает в управлении проектами и планировании.

4.4. Шкала оценивания промежуточной аттестации

Оценка	Компетенции	Дескрипторы (уровни) – основные признаки освоения (показатели достижения результата)
«отлично» (85 - 100 баллов)	ПК-7	
«хорошо» (70 - 84 баллов)	ПК-7	
«удовлетворительно» (50 - 69 баллов)	ПК-7	
«неудовлетворительно» (менее 50 баллов)	ПК-7	

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. Гниденко И. Г., Павлов Ф. Ф., Федоров Д. Ю. Технологии и методы программирования : учебное пособие для вузов. - пер. и доп; 2-е изд.. - Москва: Юрайт, 2024. - 248 с. - Текст : электронный // ЭБС «ЮРАЙТ» [сайт]. - URL: <https://urait.ru/bcode/536903>
2. Зыков С. В. Объектно-ориентированное программирование : учебник и практикум для вузов. - 2-е изд.. - Москва: Юрайт, 2024. - 151 с. - Текст : электронный // ЭБС «ЮРАЙТ» [сайт]. - URL: <https://urait.ru/bcode/537385>
3. Зыков С. В. Программирование. Функциональный подход : учебник и практикум для вузов. - 2-е изд.. - Москва: Юрайт, 2024. - 150 с. - Текст : электронный // ЭБС «ЮРАЙТ» [сайт]. - URL: <https://urait.ru/bcode/537721>
4. Малов А. В., Родионов С. В. Концепции современного программирования : учебное пособие для вузов. - Москва: Юрайт, 2024. - 96 с. - Текст : электронный // ЭБС «ЮРАЙТ» [сайт]. - URL: <https://urait.ru/bcode/544556>

6.2 Дополнительная литература:

1. Юрина Т. А. Объектно-ориентированное программирование : методические указания к курсовому проекту. - Омск: СиБАДИ, 2023. - 27 с. - URL: <https://e.lanbook.com/book/338579>
2. Соломонов, Д. В. Объектно-ориентированное программирование : лабораторный практикум. - Весь срок охраны авторского права; Объектно-ориентированное программирование. - Ставрополь: Северо-Кавказский федеральный университет, 2018. - 111 с. - Текст : электронный // IPR BOOKS [сайт]. - URL: <http://www.iprbookshop.ru/92712.html>
3. Долгинцев А. П. Объектно-ориентированное программирование : конспект лекций. - Самара: СамГУПС, 2011. - 31 с. - URL: <https://e.lanbook.com/book/130277>

6.3 Иные источники:

1. «Журнал математической физики, анализа, геометрии» - <https://vestnik.susu.ru/cmi>
2. «КомпьютерПресс». - www.compress.ru
3. Библиотека научной и учебной литературы - <http://sbiblio.com> - <http://sbiblio.com>
4. Единое окно доступа к образовательным интернет-ресурсам Федерального портала «Российское образование» - http://window.edu.ru/catalog/?p_rubr=2.1.21%2F
5. Компьютерные технологии в туризме - <http://www.arimsoft.ru/>
6. Консультант студента. Гуманитарные науки: электронно-библиотечная система - <http://www.studentlibrary.ru>
7. Консультант студента: электронно-библиотечная система - <http://www.studentlibrary.ru>
8. Национальная электронная библиотека <http://нэб.рф> - <http://нэб.рф>
9. Облачное решение от Microsoft - <https://azure.microsoft.com>
10. Образовательные ресурсы - <http://sibrc.tsu.ru/modules.php?m=1>
11. Образовательный портал "Учёба" - www.Ucheba.com
12. Образовательный портал для студентов – <http://www.alleng.ru> - <http://www.alleng.ru>
13. Российская национальная библиотека - <http://www.nlr.ru/>
14. Русская виртуальная библиотека - <https://rvb.ru/>
15. Учебный портал - www.tgspa.ru
16. Федеральный портал "Российское образование" - <http://www.edu.ru/>
17. ЭБС «Znaniy.com» - <http://www.znaniy.com/index.php?item=main>
18. Электронная библиотека учебников для вузов - <http://4du.ru/>
19. Электронный справочник «Информо» - www.informio.ru

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:

Операционная система Microsoft Windows 10

Операционная система "Альт Образование"

Microsoft Windows 10

Профессиональные базы данных и информационные справочные системы:

1. ЭБС «Университетская библиотека онлайн» . – URL: <http://www.biblioclub.ru>
2. Математические журналы МИАН - полнотекстовая коллекция журналов Математического института им. В.А. Стеклова РАН (МИАН) на платформе общероссийского портала Math-Net.Ru. – URL: <http://www.mathnet.ru>
3. Тамбовская областная универсальная научная библиотека им. А.С. Пушкина: официальный сайт. – URL: <http://www.tambovlib.ru>
4. Электронная библиотека. Образовательная платформа «Юрайт». – URL: <https://biblio-online.ru/book/sud-prisyazhnyh-442275>
5. Университетская библиотека онлайн: электронно-библиотечная система. – URL: <https://biblioclub.ru>
6. Научная электронная библиотека «КиберЛенинка». – URL: <https://cyberleninka.ru>
7. Научная электронная библиотека eLIBRARY.ru. – URL: <https://elibrary.ru>
8. Электронная библиотека ТГУ. – URL: <https://elibrary.tsutmb.ru/>
9. Юрайт: образовательная платформа, электронно-библиотечная система. – URL: <https://urait.ru>
10. Электронная библиотека РФФИ. – URL: <https://www.rfbr.ru/rffi/ru/library>
11. Российская государственная библиотека: официальный сайт. – URL: <https://www.rsl.ru>

Электронная информационно-образовательная среда

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.