ГОУ ВПО Российско-Армянский (Славянский) университет

Утверждено
Директор Института _____
А.К. Агаронян

«30» апреля 2025г., протокол № 05

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ДИСЦИПЛИНЫ

Наименование дисциплины: Б1.В.11 Структуры данных и алгоритмы (Python)

Автор (ы) ***Э. А. Газазян, старший преподаватель, к.ф.-м.н.,***
Ф.И.О, ученое звание (при наличии), ученая степень (при наличии)

Направление подготовки: 11.03.04 Электроника и наноэлектроника
Наименование образовательной программы: Квантовая информатика

Согласовано:

Заведующий Кафедрой общей физики и квантовых наноструктур

Айрапетян Д.Б.

1. АННОТАЦИЯ

1.1. Краткое описание содержания данной дисциплины;

Дисциплина «Структуры данных и алгоритмы (Python)» представляет собой курс, направленный на изучение фундаментальных структур данных и алгоритмов, применяемых в программировании. Основной целью курса является развитие у студентов глубокого понимания концепций, методов и практических навыков, необходимых для эффективного решения задач в области программирования.

В ходе курса студенты познакомятся с различными типами структур данных, включая массивы, списки, стеки, очереди, связанные списки, деревья, графы и хеш-таблицы. Они изучат принципы их построения, работы и применения для хранения и обработки данных.
Особое внимание уделяется алгоритмам сортировки и поиска, включая как простые, так и более сложные и эффективные методы. Студенты изучат такие алгоритмы, как Bubble Sort, Selection Sort, Merge Sort, Quick Sort, линейный и бинарный поиск.

Курс также охватывает темы рекурсии и динамического программирования, которые являются важными инструментами для решения сложных задач, требующих разбивки на подзадачи и оптимизации. Студенты научатся понимать и применять принципы рекурсивного программирования, а также методы динамического программирования для оптимизации решений.

Кроме того, важной частью курса является изучение алгоритмической сложности, которая позволяет оценить эффективность алгоритмов по времени выполнения и потребляемым ресурсам. Студенты ознакомятся с методами анализа времени выполнения алгоритмов и научатся оценивать пространственную сложность.

Практическая составляющая курса включает выполнение различных заданий и проектов на языке программирования Python, что позволит студентам закрепить полученные теоретические знания на практике. Лабораторные работы и домашние задания будут направлены на развитие навыков программирования и понимания принципов работы структур данных и алгоритмов.

Таким образом, дисциплина «Структуры данных и алгоритмы (Python)» предоставляет студентам необходимые знания и навыки, которые являются основой для дальнейшего изучения и профессиональной деятельности в области программирования и разработки программного обеспечения.

1.2. Трудоемкость в академических кредитах и часах, формы итогового контроля (экзамен/зачет);

Трудоемкость в академических кредитах и часах

252 адемических часа / 7 кредитов; итоговый контроль: Экзамен

1.3. Взаимосвязь дисциплины с другими дисциплинами учебного плана специальности (направления)

Численные методы в физике, Квантовая информатика, Квантовое программирование (QISKET), Программирование в физике, Функциональное программирование (Python & Wolfram)

1.4. Результаты освоения программы дисциплины:

Код компетенции *(в соответствии рабочим с учебным планом)*	**Наименование компетенции** *(в соответствии рабочим с учебным планом)*	**Код индикатора достижения компетенций** *(в соответствии рабочим с учебным планом)*	**Наименование индикатора достижений компетенций***(в соответствии рабочим с учебным планом)*
ОПК-3	Способен применять методы поиска, хранения, обработки, анализа и представления в требуемом формате информации из различных источников и баз данных, соблюдая при этом основные требования информационной безопасности	ОПК-3.1	Знает, как использовать информационно-коммуникационные технологии при поиске необходимой информации и современные принципы поиска, хранения, обработки, анализа и представления в требуемом формате информации
		ОПК-3.2	Умеет решать задачи обработки данных с помощью современных средств автоматизации
		ОПК-3.3	Владеет навыками обеспечения информационной безопасности
ОПК-4;	Способен применять современные компьютерные технологии для подготовки текстовой и конструкторско- технологической документации с учетом требований нормативной документации	ОПК-4.1	Знает, как использовать компьютерные технологии для подготовки текстовой конструкторско-технологической документации; современные интерактивные программные комплексы для выполнения и

			редактирования текстов, изображений и чертежей
		ОПК-4.2	Умеет проектировать решение конкретной задачи проекта, выбирая оптимальный способ ее решения, исходя из действующих правовых норм и имеющихся ресурсов и ограничений; использовать современные средства автоматизации разработки и выполнения конструкторской документации
		ОПК-4.3	Владеет современными программными средствами подготовки конструкторско-технологической документации
ОПК-5	Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК-5.1	Понимает принципы построения алгоритмов и компьютерных программ, пригодных для практического применения
		ОПК-5.2	Умеет на основе алгоритмов применять языки программирования для создания компьютерные программ
		ОПК-5.3	Владеет навыками программирования, отладки и тестирования компьютерных программ
ПК-2	Способен разрабатывать эффективные алгоритмы решения сформулированных задач с использованием современных языков программирования и обеспечивать их программную реализацию	ПК-2.1	Знает методы разработки эффективных алгоритмов решения научно-исследовательских задач
		ПК-2.2	Умеет использовать алгоритмы решения исследовательских задач с использованием современных языков программирования
		ПК-2.3	Владеет навыками разработки стратегии и методологии исследования изделий микро- и наноэлектроники

2. УЧЕБНАЯ ПРОГРАММА

2.1. Цели и задачи дисциплины

Основной целью курса «Структуры данных и алгоритмы (Python)» является обеспечение студентов фундаментальными знаниями и практическими навыками в области структур данных и алгоритмов, что является критически важным для разработки эффективных программных решений. Курс направлен на формирование у студентов глубокого понимания основных концепций и методик, используемых для организации, обработки и хранения данных, а также на развитие умений применять эти знания для решения конкретных задач программирования.

Курс преследует следующие основные цели:

1. Ознакомление с основными структурами данных и алгоритмами: Представить студентам различные типы структур данных (например, списки, стеки, очереди, деревья, графы) и алгоритмов (например, сортировки, поиска), а также показать их применение в различных контекстах.
2. Развитие навыков программирования на языке Python: Обучить студентов использовать язык программирования Python для реализации структур данных и алгоритмов, акцентируя внимание на практическом применении этих знаний.
3. Повышение алгоритмического мышления: Научить студентов анализировать задачи, выбирать подходящие алгоритмы и структуры данных, а также оценивать их эффективность с точки зрения времени и ресурсов.
4. Формирование практических навыков решения задач: Путем выполнения практических заданий и лабораторных работ помочь студентам освоить методы и приемы решения реальных задач программирования с использованием структур данных и алгоритмов.

2.2. Трудоемкость дисциплины и виды учебной работы (в академических часах и зачетных единицах) ***(удалить строки, которые не будут применены в рамках дисциплины)***

Виды учебной работы	**Всего, в акад. часах**	**Распределение по семестрам**
		4 сем
1	**2**	**3**
1. Общая трудоемкость изучения дисциплины по семестрам, в т. ч.:	**252**	**252**
1.1. Аудиторные занятия, в т. ч.:	**64**	**64**
1.1.1. Лекции	**32**	**32**
1.1.2. Практические занятия, в т. ч.	**32**	**32**
1.1.2.1. Контрольные работы		

1.2. Самостоятельная работа, в т. ч.:	**152**	**152**
1.2.1. Подготовка к экзаменам		
1.2.1.1. Письменные домашние задания		
1.3. Консультации		
Итоговый контроль (Экзамен/Зачет)	**Экзамен 36**	**Экзамен 36**

2.3. Содержание дисциплины

2.3.1. Тематический план и трудоемкость аудиторных занятий (модули, разделы дисциплины и виды занятий) по рабочему учебному плану

Разделы и темы дисциплины	**Всего (ак. часов)**	**Лекции (ак. часов)**	**Практ. Занятия (ак. часов)**
1	**2=3+4**	**3**	**4**
Тема 1 Введение в структуры данных и алгоритмы	**8**	4	4
Тема .2Списки и массивы	**12**	6	6
Тема 3 Алгоритмы сортировки	**16**	8	8
Тема 5 Алгоритмы поиска	**16**	8	8
Тема 6 Рекурсия	**12**	6	6
ИТОГО	**64**	**32**	**32**

2.3.2. Краткое содержание разделов дисциплины в виде тематического плана

Тема 1. Введение в структуры данных и алгоритмы

Первая тема курса посвящена знакомству с основными понятиями и терминами, используемыми в области структур данных и алгоритмов. Студенты узнают, что такое структура данных, алгоритм, а также какие задачи решаются с их помощью. Будет проведен обзор языка программирования Python и его возможностей для работы со структурами данных и алгоритмами. Основное внимание уделяется важности выбора правильных структур данных и алгоритмов для эффективного решения задач. Также рассматриваются основные параметры оценки алгоритмов, такие как время выполнения и использование памяти. Эта тема задает базис для дальнейшего изучения и позволяет студентам осознать значимость структур данных и алгоритмов в программировании.

Тема 2. Списки и массивы

Эта тема посвящена изучению списков и массивов, которые являются одними из самых простых и широко используемых структур данных. Студенты узнают о внутренней организации массивов и списков, их преимуществах и недостатках. Рассматриваются базовые операции, такие как добавление, удаление и доступ к элементам. Особое внимание уделяется динамическим спискам Python, их методам и функциям. Примеры программирования включают сортировку списков и реализацию различных алгоритмов поиска. Также обсуждается использование списков для моделирования других структур данных, таких как стеки и очереди.

Тема 3. Связанные списки

Связанные списки являются более сложной структурой данных по сравнению с массивами и списками. Эта тема охватывает односвязные и двусвязные списки, их структуру и операции. Студенты научатся создавать и манипулировать связанными списками, включая операции вставки, удаления и поиска. Рассматриваются проблемы и преимущества использования связанных списков по сравнению с массивами. Также обсуждается реализация круговых связанных списков и их применение в различных алгоритмах и структурах данных.

Тема 4. Алгоритмы сортировки

Алгоритмы сортировки играют ключевую роль в организации данных для последующего быстрого доступа и обработки. Существуют различные алгоритмы сортировки, от простых (Bubble Sort, Selection Sort) до более сложных и эффективных (Merge Sort, Quick Sort). Для физиков, работающих с упорядоченными данными или анализирующих большие объемы информации, понимание алгоритмов сортировки важно для оптимизации вычислительных процессов. В этом разделе студенты изучат принципы работы различных алгоритмов сортировки, их сравнительные характеристики и примеры применения в решении реальных задач.

Тема 5. Алгоритмы поиска

Алгоритмы поиска используются для нахождения элементов в структурах данных. Линейный поиск и бинарный поиск являются основными методами, применяемыми для последовательных и отсортированных данных соответственно. Поиск в графах, включая

алгоритмы глубинного (DFS) и ширинного (BFS) обхода, используется для анализа сложных систем связей. Для физиков, занимающихся анализом данных или моделированием систем, эффективные методы поиска необходимы для быстрого и точного извлечения информации. В этом разделе студенты изучат различные алгоритмы поиска, их реализацию и применение, а также методы оптимизации поиска в различных структурах данных.

Тема 6. Рекурсия

Рекурсия – это мощный инструмент программирования, который позволяет функции вызывать саму себя для решения задачи. В этой теме студенты познакомятся с основными концепциями и принципами рекурсивного подхода, его преимуществами и недостатками, а также научатся применять рекурсию для решения различных задач.

2.3.3. Краткое содержание семинарских/практических занятий/лабораторного практикума

Практические занятия

1. **Основы работы с Python**

- Знакомство с основными конструкциями языка
- Выполнение простых программ на Python

2. Работа с массивами и списками

- Практические задачи на создание и модификацию списков и массивов

3. Связанные списки

- Выполнение задач с использованием связанных списков

4. Алгоритмы сортировки и поиска

- Практическое применение и сравнение алгоритмов

5. Рекурсия

- Решение задач с использованием рекурсивных методов

Эти семинарские, практические занятия и лабораторные работы направлены на укрепление теоретических знаний и развитие практических навыков студентов в области структур данных и алгоритмов с использованием языка Python.

2.3.4. Материально-техническое обеспечение дисциплины

Для успешного освоения дисциплины "Структуры данных и алгоритмы (Python)" необходимо соответствующее материально-техническое обеспечение. Это включает в себя следующие компоненты:

1. **Компьютерные классы:**

- Оснащенные современными персональными компьютерами или ноутбуками с установленной операционной системой (Windows или Linux).
- Доступ к сети интернет для скачивания необходимых библиотек и инструментов.
- Среда разработки (IDE) для программирования на Python, такие как PyCharm, Visual Studio Code или Jupyter Notebook.

2. Программное обеспечение:

- Установленный интерпретатор Python последней версии (рекомендуется Python 3.x).
- Библиотеки для численных вычислений и символьных расчётов (NumPy, SymPy, SciPy).
- Пакеты для визуализации данных (Matplotlib).
- Менеджер пакетов pip для установки и управления библиотеками.

3. Учебные материалы:

- Учебные пособия и методические указания по каждой теме курса.
- Доступ к онлайн-ресурсам и документации по Python и его библиотекам.
- Практические задания и проекты для самостоятельной работы.

4. **Дополнительные ресурсы:**

- Электронная библиотека с доступом к книгам по "Структуры данных и алгоритмы (Python)".

Обеспечение всеми вышеперечисленными компонентами является необходимым для полного и эффективного усвоения материала дисциплины "Структуры данных и алгоритмы (Python)".

2.4. Модульная структура дисциплины с распределением весов по формам контролей

Формы контролей	Вес формы (форм) текущего контроля в результирующей оценке текущего контроля (по модулям)		Вес формы промежуточного контроля в итоговой оценке промежуточного контроля		Вес итоговой оценки промежуточного контроля в результирующей оценке промежуточных контролей		Вес итоговой оценки промежуточного контроля в результирующей оценке промежуточных контролей (семестровой оценке)	Веса результирующей оценки промежуточных контролей и оценки итогового контроля в результирующей оценке итогового контроля
Вид учебной работы/контроля	**М1**[1]	**М2**	**М1**	**М2**	**М1**	**М2**		
Контрольная работа *(при наличии)*			0.5	0.5				
Устный опрос *(при наличии)*								
Тест *(при наличии)*								
Лабораторные работы *(при наличии)*	0.5	0.5						
Письменные домашние задания *(при наличии)*								
Реферат *(при наличии)*								
Эссе *(при наличии)*								
Проект *(при наличии)*								
Решение задач	0.5	0.5						
Веса результирующих оценок текущих контролей в итоговых оценках промежуточных контролей					0.5	0.5		
Веса оценок промежуточных контролей в итоговых оценках промежуточных контролей								
Вес итоговой оценки 1-го промежуточного контроля в результирующей оценке промежуточных контролей							0.5	
Вес итоговой оценки 2-го промежуточного контроля в результирующей оценке промежуточных контролей							0.5	
Вес результирующей оценки промежуточных контролей в результирующей оценке итогового контроля								0.5
Вес итогового контроля (Экзамен/зачет) в результирующей оценке итогового контроля								0.5
	∑ = 1	**∑ = 1**	**∑ = 1**	**∑ = 1**	**∑ = 1**	**∑ = 1**	**∑ = 1**	**∑ = 1**

3. Теоретический блок *(указываются материалы, необходимые для освоения учебной программы дисциплины)*

[1] Учебный Модуль

3.1. Материалы по теоретической части курса

3.1.1. Учебник(и);

1. «Структуры данных и алгоритмы на Python» автор: Никлаус Вирт

2. «Алгоритмы. Построение и анализ» авторы: Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клифорд Штайн

3.1.2. Учебное(ые) пособие(я);

Python Documentation (docs.python.org)

4. Фонды оценочных средств ***(указываются материалы, необходимые для проверки уровня знаний в соответствии с содержанием учебной программы дисциплины).***

4.1. Планы практических и семинарских занятий

- **Практическое занятие 1: Введение в Python и его встроенные структуры данных**
  План: Обзор синтаксиса Python, работа со списками, кортежами, словарями и множествами.
- **Практическое занятие 2: Массивы и списки**
  План: Создание и инициализация массивов и списков, операции добавления, удаления и изменения элементов.
- **Практическое занятие 3: Стек и очередь**
  План: Реализация стека и очереди на основе списка, примеры задач с использованием стека и очереди.
- **Практическое занятие 4: Связанные списки**
  План: Реализация односвязных и двусвязных списков, операции вставки, удаления и поиска.
- **Практическое занятие 5: Бинарные деревья поиска**
  План: Вставка, удаление и поиск элементов в бинарных деревьях поиска, алгоритмы обхода дерева.
- **Практическое занятие 6: Алгоритмы сортировки**
  План: Реализация пузырьковой сортировки, сортировки вставками и выбором, быстрой сортировки и сортировки слиянием.
- **Практическое занятие 7: Алгоритмы поиска**
  План: Линейный и бинарный поиск, примеры задач и оптимизация поиска.

- **Практическое занятие 8: Рекурсия**

  План: Реализация рекурсивных функций, примеры задач с использованием рекурсии.

4.2. Планы практикумов

- Практикум 1: Основы Python и встроенные структуры данных
- Темы: Списки, кортежи, словари, множества. Основные операции и методы.
- Практикум 2: Массивы и списки
- Темы: Создание и модификация массивов и списков, их применение.
- Практикум 3: Стек и очередь
- Темы: Реализация и использование стека и очереди.
- Практикум 4: Связанные списки
- Темы: Односвязные и двусвязные списки, основные операции.
- Практикум 5: Деревья
- Темы: Реализация бинарных деревьев и основные операции.
- Практикум 6: Графы и их алгоритмы
- Темы: Представление графов, алгоритмы поиска (DFS, BFS).
- Практикум 7: Алгоритмы сортировки
- Темы: Реализация различных алгоритмов сортировки, сравнение их эффективности.
- Практикум 8: Алгоритмы поиска
- Темы: Линейный и бинарный поиск, оптимизация поиска.

4.3. Материалы по практической части курса

4.3.1. Учебно-методические пособия

- **«**Практическое руководство по структурам данных и алгоритмам на Python» автор: Иванов И.И.
- «Алгоритмы на Python для начинающих» автор: Петров П.П.

4.3.2. Учебные справочники

- «Справочник по Python» автор: Смит Дж.
- «Структуры данных и алгоритмы: справочник программиста» автор: Кларк А.

4.3.3. Задачники (практикумы)

- «Задачник по структурам данных и алгоритмам на Python» автор: Браун К.
- «Практикум по алгоритмам и структурам данных» автор: Джонс М.

4.3.4. Наглядно-иллюстративные материалы

- Презентации и схемы по структурам данных и алгоритмам.

4.3.5. Другие виды материалов

- Интерактивные учебные материалы на платформе Codeacademy, Coursera и т.д.

4.4. Вопросы и задания для самостоятельной работы студентов

- Реализовать и протестировать функции для основных операций с массивами и списками.
- Написать рекурсивную функцию для вычисления факториала числа.
- Разработать алгоритм поиска элемента в связном списке.
- Написать программу для сортировки массива методом быстрой сортировки.

4.5. Тематика рефератов, эссе и других форм самостоятельных работ

- Сравнительный анализ алгоритмов сортировки.
- Применение рекурсии в решении комбинаторных задач.
- Оптимизация алгоритмов поиска.
- Реализация и применение графов в реальных задачах.

4.6. Образцы вариантов контрольных работ, тестов и/или других форм текущих и промежуточных контролей

- Тесты на знание теоретических основ структур данных и алгоритмов.
- Контрольные работы с задачами на реализацию структур данных и алгоритмов.
- Задания на анализ и оценку эффективности алгоритмов.

4.7. Перечень экзаменационных вопросов

- Объясните основные концепции и виды структур данных.
- Опишите принципы работы рекурсии и приведите примеры.
- Каковы основные алгоритмы сортировки и их характеристика?
- В чем заключается различие между стеком и очередью?

Тема 1. Введение в структуры данных и алгоритмы

1. Что такое структура данных и зачем она используется?
2. Каковы основные типы структур данных?
3. Объясните разницу между линейными и нелинейными структурами данных.
4. Что такое абстрактные типы данных (ADT)?
5. Приведите примеры абстрактных типов данных.
6. Какие основные характеристики алгоритма?

7. Что такое рекурсия и итерация?
8. Приведите примеры задач, решаемых с помощью рекурсии.
9. Объясните принцип работы стековой машины.
10. Что такое очереди и где они применяются?

Тема 2. Списки и массивы

1. Что такое массив и как он организован в памяти?
2. Каковы основные операции с массивами?
3. Объясните процесс добавления и удаления элементов в массиве.
4. Что такое динамический массив?
5. Опишите отличие между массивами и списками.
6. Что такое связанный список?
7. Опишите процесс реализации очереди на основе массива.
8. Опишите процесс реализации очереди на основе связанного списка.

Тема 3. Алгоритмы сортировки

1. Что такое алгоритм сортировки?
2. Какие существуют типы алгоритмов сортировки?
3. Объясните принцип работы пузырьковой сортировки.
4. Какова временная сложность сортировки слиянием?
5. Что такое быстрая сортировка?
6. Как работает пирамидальная сортировка?
7. В чем преимущества и недостатки сортировки вставками?
8. В каких случаях целесообразно использовать сортировку пузырьком?
9. Какие факторы влияют на выбор алгоритма сортировки для конкретной задачи?

Тема 5. Алгоритмы поиска

1. Что такое алгоритм поиска?
2. Какие основные виды алгоритмов поиска существуют?
3. Объясните принцип работы линейного поиска.
4. Какова временная сложность линейного поиска?
5. Что такое бинарный поиск?
6. В чем разница между линейным и бинарным поиском?
7. Как реализовать поиск в несортированном массиве?
8. Что такое поиск по строке?

Тема 6. Рекурсия

1. Что такое рекурсия?
2. Объясните принцип работы рекурсивных функций.
3. Какова структура рекурсивной функции?
4. Что такое базовый случай в рекурсии?
5. Объясните рекурсивный случай.
6. В чем преимущества и недостатки рекурсии?
7. Приведите примеры задач, решаемых с помощью рекурсии.
8. Что такое рекурсивное переполнение стека?
9. Объясните процесс вычисления факториала числа с помощью рекурсии.
10. Как с помощью рекурсии вычислить числа Фибоначчи?
11. Что такое хвостовая рекурсия?
12. В чем отличие хвостовой рекурсии от обычной рекурсии?

4.8. Образцы экзаменационных билетов

Тема 1. Введение в структуры данных и алгоритмы

1. Что такое структура данных и зачем она используется?
2. Каковы основные типы структур данных?
3. Объясните разницу между линейными и нелинейными структурами данных.
4. Что такое абстрактные типы данных (ADT)?
5. Приведите примеры абстрактных типов данных.
6. Какие основные характеристики алгоритма?
7. Что такое рекурсия и итерация?
8. Приведите примеры задач, решаемых с помощью рекурсии.
9. Объясните принцип работы стековой машины.
10. Что такое очереди и где они применяются?

Тема 2. Списки и массивы

1. Что такое массив и как он организован в памяти?
2. Каковы основные операции с массивами?
3. Объясните процесс добавления и удаления элементов в массиве.
4. Что такое динамический массив?
5. Опишите отличие между массивами и списками.

6. Что такое связанный список?
7. Опишите процесс реализации очереди на основе массива.
8. Опишите процесс реализации очереди на основе связанного списка.

Тема 3. Алгоритмы сортировки

1. Что такое алгоритм сортировки?
2. Какие существуют типы алгоритмов сортировки?
3. Объясните принцип работы пузырьковой сортировки.
4. Какова временная сложность сортировки слиянием?
5. Что такое быстрая сортировка?
6. Как работает пирамидальная сортировка?
7. В чем преимущества и недостатки сортировки вставками?
8. В каких случаях целесообразно использовать сортировку пузырьком?
9. Какие факторы влияют на выбор алгоритма сортировки для конкретной задачи?

Тема 5. Алгоритмы поиска

1. Что такое алгоритм поиска?
2. Какие основные виды алгоритмов поиска существуют?
3. Объясните принцип работы линейного поиска.
4. Какова временная сложность линейного поиска?
5. Что такое бинарный поиск?
6. В чем разница между линейным и бинарным поиском?
7. Как реализовать поиск в несортированном массиве?
8. Что такое поиск по строке?

Тема 6. Рекурсия

1. Что такое рекурсия?
2. Объясните принцип работы рекурсивных функций.
3. Какова структура рекурсивной функции?
4. Что такое базовый случай в рекурсии?
5. Объясните рекурсивный случай.
6. В чем преимущества и недостатки рекурсии?
7. Приведите примеры задач, решаемых с помощью рекурсии.
8. Что такое рекурсивное переполнение стека?
9. Объясните процесс вычисления факториала числа с помощью рекурсии.

10. Как с помощью рекурсии вычислить числа Фибоначчи?
11. Что такое хвостовая рекурсия?
12. В чем отличие хвостовой рекурсии от обычной рекурсии?

-

4.9. Образцы экзаменационных практических заданий

- Реализовать бинарное дерево поиска и написать функции для вставки и поиска элементов.
- Написать программу для сортировки массива методом слияния.

4.10. Банк тестовых заданий для самоконтроля

- Множественный выбор, правда/ложь и задачи на написание кода по темам курса.

4.11. Методики решения и ответы к образцам тестовых заданий

- Подробные решения и объяснения к тестам и практическим заданиям.

5. Методический блок

5.1. Методика преподавания

5.1.1. Методические рекомендации для студентов по подготовке к семинарским, практическим или лабораторным занятиям, по организации самостоятельной работы студентов при изучении конкретной дисциплины.

Подготовка к семинарским занятиям

1. **Изучение теоретического материала**: Перед каждым семинаром внимательно изучите соответствующий раздел учебника и дополнительные материалы, предоставленные преподавателем. Обратите внимание на основные концепции, определения и примеры.
2. **Конспектирование**: Составьте конспект ключевых моментов и вопросов, которые необходимо обсудить на семинаре. Это поможет лучше понять материал и подготовиться к активному участию в дискуссии.
3. **Просмотр видеолекций**: Если доступны видеолекции по теме семинара, обязательно посмотрите их. Видеолекции помогут лучше усвоить материал и подготовиться к обсуждению.
4. **Подготовка вопросов**: Сформулируйте вопросы по теме, которые остались непонятными или требуют дополнительного пояснения. Это позволит вам активнее участвовать в обсуждениях и получать разъяснения от преподавателя.

Подготовка к практическим занятиям

1. **Программирование на Python**: Убедитесь, что знакомы с основами языка Python. Выполните несколько простых упражнений, чтобы освежить знания.
2. **Решение задач**: Попробуйте решить несколько задач по теме предстоящего занятия самостоятельно. Это могут быть задачи из учебника или дополнительных источников.
3. **Анализ примеров кода**: Ознакомьтесь с примерами кода, которые будут обсуждаться на занятии. Попробуйте понять их работу и, если возможно, выполните их на своем компьютере.
4. **Подготовка к обсуждению**: Обсудить свои решения и подходы к задачам с преподавателем и одногруппниками. Запишите вопросы или проблемы, с которыми вы столкнулись при выполнении упражнений.

Организация самостоятельной работы

1. **Планирование времени**: Составьте расписание для самостоятельной работы. Определите время для чтения теоретического материала, выполнения практических заданий и повторения изученного материала.
2. **Использование дополнительных ресурсов**: Воспользуйтесь дополнительными ресурсами, такими как онлайн-курсы, видеолекции, статьи и блоги, чтобы углубить свои знания по теме.
3. **Регулярная практика**: Регулярно выполняйте упражнения и задачи, чтобы закрепить изученный материал. Постепенно увеличивайте сложность задач.
4. **Обратная связь**: Обсуждайте свои решения с одногруппниками и преподавателем. Не стесняйтесь задавать вопросы и просить разъяснений.
5. **Работа в группе**: Сотрудничайте с другими студентами. Обсуждайте решения задач, делитесь опытом и помогайте друг другу.