МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»

**Обнинский институт атомной энергетики –**

филиал федерального государственного автономного образовательного учреждения высшего образования
«Национальный исследовательский ядерный университет «МИФИ»

**(ИАТЭ НИЯУ МИФИ)**

Утверждено на заседании
УМС ИАТЭ НИЯУ МИФИ
Протокол №2-8/2024 От 30.08.2024

# РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Базовый Python + объектно-ориентированное программирование (ООП) в Python

*Шифр, название дисциплины*

01.04.02 «Прикладная математика и информатика»

*Шифр, название специальности/направления подготовки*

Математическое моделирование и прикладной анализ данных

*Название программы магистратуры*

магистр

*(Квалификация (степень) выпускника)*

Форма обучения: очная

**г. Обнинск 2024 г.**

Программа составлена в соответствии с требованиями образовательного стандарта высшего образования национального исследовательского ядерного университета «МИФИ» по направлению подготовки 01.04.02 – Прикладная математика и информатика. (квалификация (степень) магистр).

Программу составил:

_____ С.В. Ермаков, доцент, к.ф.-м.н, доцент

Рецензент:

_____ Г.Е. Деев, доцент, к.ф.-м.н, доцент

Программа рассмотрена на заседании ОИКС

(протокол № 5/7 от «30» июля от 2024 г.)

Руководитель направления подготовки 01.04.02
«Прикладная математика и информатика»

_____ Ермаков С.В.

«____»_____2024 г.

**1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы**

В результате освоения ООП магистратуры обучающийся должен овладеть следующими результатами обучения по дисциплине:

| Коды компетенций | Результаты освоения ООП *Содержание компетенций** | Перечень планируемых результатов обучения по дисциплине** |
|---|---|---|
| ОПК-1 | Способен решать актуальные задачи фундаментальной и прикладной математики | З-ОПК-1 Знать актуальные задачи фундаментальной и прикладной математики, методы математического моделирования.<br>У-ОПК-1 Уметь использовать методы математического моделирования для решения задач фундаментальной и прикладной математики.<br>В-ОПК-1 Владеть методами математического моделирования и основами их использования. |

**2. Место дисциплины в структуре ООП магистратуры**

Дисциплина реализуется в рамках общенаучного модуля.

Для освоения дисциплины необходимы компетенции, сформированные в рамках изучения следующих дисциплин: **Linux**
Дисциплина изучается на 1 курсе в 1 семестре.

**3. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам занятий) и на самостоятельную работу обучающихся**

Общая трудоемкость (объем) дисциплины составляет 5 зачетных единиц (з.е.), 180 академических часа.

***3.1. Объём дисциплины по видам учебных занятий (в часах)***

| | **Семестр** | | |
|---|---|---|---|
| | **№ 1** | **№ 2** | **Всего** |
| | **Количество часов на вид работы:** | | |
| **Контактная работа обучающихся с преподавателем** | | | |
| **Аудиторные занятия *(всего)*** | **64** | | **64** |
| В том числе: | | | |
| *лекции* | 32 | | 32 |
| *практические занятия* | 32 | | 32 |
| *лабораторные занятия* | | | |
| **Промежуточная аттестация** | | | |
| В том числе: | | | |
| *зачет* | | | |
| *экзамен* | 36 | | 36 |
| **Самостоятельная работа обучающихся** | **80** | | **80** |

| | | | |
|---|---|---|---|
| ***(всего)*** | | | |
| В том числе: | | | |
| *проработка учебного (теоретического) материала* | 20 | | 20 |
| *выполнение индивидуальных заданий* | 20 | | 20 |
| *подготовка ко всем видам контрольных испытаний текущего контроля успеваемости (в течение семестра)* | 20 | | 20 |
| *подготовка ко всем видам контрольных испытаний промежуточной аттестации (по окончании семестра)* | 20 | | 20 |
| *Всего (часы):* | **180** | | **180** |
| *Всего (зачетные единицы):* | **5** | | **5** |

**4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий**

***4.1. Разделы дисциплины и трудоемкость по видам учебных занятий (в академических часах)***

| **№ п/п** | **Наименование раздела /темы дисциплины** | **Общая трудоём-кость всего** *(в часах)* | **Виды учебных занятий, включая самостоятельную работу обучающихся и трудоемкость** *(в часах)* | | | | **Формы текущего контроля успевае-мости** |
|---|---|---|---|---|---|---|---|
| | | | **Аудиторные учебные занятия** | | | **СРО** | |
| | | | **Лек** | **Сем/Пр** | **Лаб** | | |
| 1. | | | 32 | 32 | - | 80 | |
| 1.1. | Введение | 4 | 1 | 1 | | 2 | |
| 1.2. | Переменные. Базовые типы данных | 6 | 2 | 2 | - | 2 | |
| 1.3. | Сложные типы данных | 6 | 2 | 2 | - | 2 | |
| 1.3. | Сложные типы данных | 6 | 2 | 2 | - | 2 | |
| 1.4. | Дополнительные типы данных | 4 | 1 | 1 | | 2 | |
| 1.5. | Функции | 4 | 1 | 1 | - | 2 | |
| 1.6. | Условия | 4 | 1 | 1 | - | 2 | |
| 1.7. | Циклы for и while | 4 | 1 | 1 | - | 2 | |
| 1.8. | Модули и библиотеки | 4 | 1 | 1 | | 2 | |
| 1.9. | Базовая работа с файлами | 4 | 1 | 1 | | 2 | |
| 1.10. | Промежуточные бизнес-кейсы | 4 | 1 | 1 | - | 2 | |
| 1.11 | лямбда-функции и функциональное программирование | 4 | 1 | 1 | | 2 | |
| 1.12 | Comprehensions | 4 | 1 | 1 | | 2 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.13 | Итерируемые объекты, итераторы, генераторы, генераторные выражения | 4 | 1 | 1 | | 2 | |
| 1.14 | Промежуточные бизнес-кейсы | 6 | 1 | 1 | | 4 | |
| 1.15 | Исключения | 4 | 1 | 1 | | 2 | |
| 1.16 | Оператор * | 4 | 1 | 1 | | 2 | |
| 1.17 | Работа с датой и временем | 4 | 1 | 1 | | 2 | |
| 1.18 | Промежуточные бизнес-кейсы | 6 | 1 | 1 | | 4 | |
| 1.19 | Регулярные выражения | 4 | 1 | 1 | | 2 | |
| 1.20 | Работа с базами данных через Python | 4 | 1 | 1 | | 2 | |
| 1.21 | Модуль requests | 4 | 1 | 1 | | 2 | |
| 1.22 | Итоговый проект | 6 | 1 | 1 | | 4 | |
| 1.23 | Основы ООП | 4 | 1 | 1 | | 2 | |
| 1.24 | Инкапсуляция | 4 | 1 | 1 | | 2 | |
| 1.25 | Наследование | 4 | 1 | 1 | | 2 | |
| 1.26 | Полиморфизм, абстрактные классы и метод | 4 | 1 | 1 | | 2 | |
| 1.27 | Дополнительные приемы - property, staticmethod, геттеры-сеттеры, декораторы | 4 | 1 | 1 | | 2 | |
| 1.28 | Магические методы, Singleton | 4 | 1 | 1 | | 2 | |
| 1.29 | Итоговые бизнес-кейсы: ООП | 6 | 1 | 1 | | 8 | |

### *4.2. Содержание дисциплины, структурированное по разделам (темам)*

*Лекционный курс*

| № | Наименование раздела /темы дисциплины | Содержание |
|---|---|---|
| 1.1. | Введение | истории языка Python<br>изучим инструменты для работы |
| 1.2. | Переменные. Базовые типы данных | рассмотрим основы: поговорим о переменных, типах данных и строковых функциях |
| 1.3. | Сложные типы данных | Изучим списки, кортежи, множества и словари. |
| 1.4. | Дополнительные типы данных | три менее известных, но невероятно полезных типа данных в Python: Counter, namedtuple и frozenset. |
| 1.5. | Функции | функции в python |
| 1.6. | Условия | условия в python |
| 1.7. | Циклы for и while | основы цикла while и цикла for |
| 1.8. | Модули и библиотеки | импортировать и использовать модули и библиотеки на Python |

| | | |
|---|---|---|
| 1.9. | Базовая работа с файлами | основы работы с файлами на Python |
| 1.10. | Промежуточные бизнес-кейсы | практический урок |
| 1.11 | лямбда-функции и функциональное программирование | использование лямбда-функций |
| 1.12 | Comprehensions | концепция языка программирования Python - Comprehensions |
| 1.13 | Итерируемые объекты, итераторы,генераторы, генераторные выражения | Итерируемые объекты и итераторы |
| 1.14 | Промежуточные бизнес-кейсы | Промежуточные бизнес-кейсы |
| 1.15 | Исключения | Исключения в языке Python |
| 1.16 | Оператор * | Оператор asterisk |
| 1.17 | Работа с датой и временем | Встроенный модуль datetime |
| 1.18 | Промежуточные бизнес-кейсы | Промежуточные бизнес-кейсы |
| 1.19 | Регулярные выражения | основы регулярных выражений в Python |
| 1.20 | Работа с базами данных через Python | Работа с базами данных через Python. |
| 1.21 | Модуль requests | основные возможности модуля requests |
| 1.22 | Итоговый проект | итоговый проект по базовому python |
| 1.23 | Основы ООП | объектно-ориентированное программирования в Python |
| 1.24 | Инкапсуляция | способы реализации инкапсуляции в Python |
| 1.25 | Наследование | как создавать иерархии классов, переопределять методы, вызывать методы родительских классов,понятия множественного наследования и абстрактных классов |
| 1.26 | Полиморфизм, абстрактные классы и метод | полиморфизм, абстрактные классы и методы |
| 1.27 | Дополнительные приемы - property, staticmethod, геттеры-сеттеры, декораторы | Дополнительные приемы ООП: property, staticmethod, геттеры-сеттеры, дополнительные декораторы |
| 1.28 | Магические методы, Singleton | магические методы и паттерн проектирования Singleton |
| 1.29 | Итоговые бизнес-кейсы: ООП | итоговая практика |

*Практические/семинарские занятия*

| **№** | **Наименование раздела /темы дисциплины** | **Содержание** |
|---|---|---|
| 1.1. | Введение | истории языка Python<br>изучим инструменты для работы |
| 1.2. | Переменные. Базовые типы данных | рассмотрим основы: поговорим о переменных, типах данных и строковых функциях |
| 1.3. | Сложные типы данных | Изучим списки, кортежи, множества и словари. |

| | | |
|---|---|---|
| 1.4. | Дополнительные типы данных | три менее известных, но невероятно полезных типа данных в Python: Counter, namedtuple и frozenset. |
| 1.5. | Функции | функции в python |
| 1.6. | Условия | условия в python |
| 1.7. | Циклы for и while | основы цикла while и цикла for |
| 1.8. | Модули и библиотеки | импортировать и использовать модули и библиотеки на Python |
| 1.9. | Базовая работа с файлами | основы работы с файлами на Python |
| 1.10. | Промежуточные бизнес-кейсы | практический урок |
| 1.11 | лямбда-функции и функциональное программирование | использование лямбда-функций |
| 1.12 | Comprehensions | концепция языка программирования Python - Comprehensions |
| 1.13 | Итерируемые объекты, итераторы,генераторы, генераторные выражения | Итерируемые объекты и итераторы |
| 1.14 | Промежуточные бизнес-кейсы | Промежуточные бизнес-кейсы |
| 1.15 | Исключения | Исключения в языке Python |
| 1.16 | Оператор * | Оператор asterisk |
| 1.17 | Работа с датой и временем | Встроенный модуль datetime |
| 1.18 | Промежуточные бизнес-кейсы | Промежуточные бизнес-кейсы |
| 1.19 | Регулярные выражения | основы регулярных выражений в Python |
| 1.20 | Работа с базами данных через Python | Работа с базами данных через Python. |
| 1.21 | Модуль requests | основные возможности модуля requests |
| 1.22 | Итоговый проект | итоговый проект по базовому python |
| 1.23 | Основы ООП | объектно-ориентированное программирования в Python |
| 1.24 | Инкапсуляция | способы реализации инкапсуляции в Python |
| 1.25 | Наследование | как создавать иерархии классов, переопределять методы, вызывать методы родительских классов,понятия множественного наследования и абстрактных классов |
| 1.26 | Полиморфизм, абстрактные классы и метод | полиморфизм, абстрактные классы и методы |
| 1.27 | Дополнительные приемы - property, staticmethod, геттеры-сеттеры, декораторы | Дополнительные приемы ООП: property, staticmethod, геттеры-сеттеры, дополнительные декораторы |
| 1.28 | Магические методы, Singleton | магические методы и паттерн проектирования Singleton |
| 1.29 | Итоговые бизнес-кейсы: ООП | итоговая практика |

*Лабораторные занятия*

Не предусмотрены.

## 5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

В качестве учебно-методических материалов используется рекомендованная литература.

## 6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

### *6.1. Паспорт фонда оценочных средств по дисциплине*

| № п/п | Контролируемые разделы (темы) дисциплины (результаты по разделам) | Код контролируемой компетенции (или её части) / и ее формулировка | Наименование оценочного средства |
|---|---|---|---|
| 1.8. | Модули и библиотеки | ОПК-1 | Контрольная работа № 1 |
| 1.11. | лямбда-функции и функциональное программирование | ОПК-1 | Контрольная работа № 1 |
| 1.20 | Работа с базами данных через Python | ОПК-1 | Контрольная работа № 2 |
| 1.28. | Магические методы, Singleton | ОПК-1 | Контрольная работа № 2 |

### *6.2. Типовые контрольные задания или иные материалы*

#### *6.2.1. Экзамен*

В экзаменационном билете два теоретических вопроса и один практический

Теоретические вопросы билета:

1. Что такое переменные и базовые типы данных в Python? Приведите примеры.
2. Какие существуют сложные типы данных в Python? Приведите примеры и объясните их использование.
3. Что представляют собой дополнительные типы данных в Python? Чем они отличаются от базовых и сложных типов?
4. Как создаются и используются функции в Python? Объясните основные принципы работы с функциями.
5. Как реализуются условия в Python? Приведите примеры использования условных операторов.
6. Как работают циклы for и while в Python? Приведите примеры их использования.
7. Что такое модули и библиотеки в Python? Как происходит их подключение и использование?
8. Как осуществляется базовая работа с файлами в Python? Приведите примеры чтения и записи данных в файлы.
9. Как решаются промежуточные бизнес-кейсы с использованием Python? Приведите пример решения задачи с использованием языка.
10. Что такое лямбда-функции и функциональное программирование в Python? Приведите примеры использования лямбда-функций.

11. Что такое comprehension в Python? Как они используются для создания списков, множеств и словарей?
12. Как работают итерируемые объекты, итераторы, генераторы и генераторные выражения в Python? Приведите примеры.
13. Как работают исключения в Python? Какие типы исключений существуют и как с ними работать?
14. Что такое оператор в Python и как его можно использовать? Приведите примеры различных операторов.
15. Как работать с датой и временем в Python? Какие стандартные модули используются для этого?
16. Какие решения можно предложить для промежуточных бизнес-кейсов с использованием Python? Приведите примеры.
17. Что такое регулярные выражения в Python? Как их использовать для обработки строк?
18. Как работать с базами данных через Python? Какие библиотеки и модули для этого существуют?
19. Как использовать модуль requests для работы с HTTP-запросами в Python? Приведите примеры использования.
20. Что такое основы объектно-ориентированного программирования (ООП) в Python? Объясните основные принципы ООП.
21. Что такое инкапсуляция в ООП и как она реализуется в Python? Приведите примеры.
22. Как работает наследование в Python? Приведите пример наследования классов в Python.
23. Что такое полиморфизм, абстрактные классы и методы в Python? Объясните с примерами.
24. Как использовать дополнительные приемы в Python, такие как property, staticmethod, геттеры-сеттеры и декораторы?
25. Что такое магические методы в Python? Как реализовать паттерн Singleton с использованием магических методов?

Критерий оценки – правильность и полнота ответа на вопросы. Оценка выставляется по шкале от 0 до 40 баллов: теоретические вопросы –30 баллов, 10 баллов– дополнительные вопросы. Экзамен считается сданным при оценке не ниже 25 баллов.

### *6.2.2. Контрольная работа № 1*

На вход функции flat_tuple подается кортеж tup, состоящий из произвольного количества списков. Эти списки, в свою очередь, состоят из произвольных элементов - это могут быть и числа, и строки, и None, и другие коллекции.

Задача состоит в том, чтобы максимально простым способом сделать такой кортеж «плоским». Другими словами, нужно вернуть список, состоящий из элементов всех подсписков исходного кортежа.

### *6.2.2. Контрольная работа № 2*

Напишите функцию find_overlap, которая принимает на вход список из двух временных интервалов (например, 9:00-13:30 и 12:00-13:30) и возвращает перекрывающий временной

интервал (например, 12:00-13:30 в данном случае). Если перекрывающего интервала нет - верните строку "Совместного интервала нет".

Входные данные представлены списком строк в формате "ЧЧ:ММ-ЧЧ:ММ".

Если имеется более двух временных интервалов, необходимо найти максимальное перекрытие между всеми парами интервалов.

б) критерии оценивания компетенций (результатов) – правильная работа кода программы, понимание алгоритма метода оптимизации, умение вывести необходимые для алгоритма формулы.

в) описание шкалы оценивания:
Каждая задача оценивается по шкале от 0 до 10 баллов.
Контрольная работа считается выполненной успешно при суммарной оценке не ниже 18 баллов.

***6.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций***

| Форма аттестации | Наименование оценочного средства | Баллы |
|---|---|---|
| Экзамен (100 баллов) | Контрольная работа № 1 | 30 |
| | Контрольная работа № 2 | 30 |
| | Ответы на экзаменационный билет | 40 |

**7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины**

***а) основная учебная литература:***


1. Горелик, Йен: Высокопроизводительные Python-приложения. Практическое руководство по эффективному программированию. 2022
2. Марк Лутц: Изучаем Python. 2020 * Дэн Бейдер: Чистый Python. Тонкости программирования для профи. 2021


***б) дополнительная учебная литература:***

-

**8. Перечень ресурсов* информационно-телекоммуникационной сети «Интернет» (далее - сеть «Интернет»), необходимых для освоения дисциплины**

-

**9. Методические указания для обучающихся по освоению дисциплины**

| Вид учебного занятия | Организация деятельности студента |
|---|---|
| Лекция | Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка |

| | |
|---|---|
| | терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначить вопросы, термины, материал, который вызывает трудности, пометить и попытаться найти ответ в рекомендуемой литературе. Если самостоятельно не удается разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на консультации, на практическом занятии. |
| Практические занятия | Проработка рабочей программы, уделяя особое внимание целям и задачам, структуре и содержанию дисциплины. Работа с конспектом лекций, просмотр рекомендуемой литературы. Изучение выбранной предметной области на примерах решения задач семинарских занятий, индивидуальных домашних заданий. |
| Курсовая работа | Не предусмотрена |
| Контрольная работа | Ознакомиться с основной и дополнительной литературой, включая справочные издания, зарубежные источники, основополагающие термины. Попрактиковаться в решении аналогичных домашних задач по всем темам контрольных работ. |
| Лабораторная работа | Не предусмотрена. |
| Подготовка к экзамену | При подготовке к экзамену необходимо ориентироваться на конспекты лекций и рекомендуемую литературу. |

## 10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем (при необходимости)

Издательская система LaTeX для подготовки докладов, презентаций и учебного материала.

## 11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Видеопроектор, компьютер, издательская система LaTeX для подготовки докладов, презентаций и учебного материала.

## 12. Иные сведения и (или) материалы

### *12.1. Перечень образовательных технологий, используемых при осуществлении образовательного процесса по дисциплине*

Часов в интерактивной форме – 8.

В ходе практических занятий происходит публичное обсуждение каждой решаемой задачи. При этом студенты высказывают свои мнения по выбору наиболее простого способа поиска оптимального решения.

После решения домашних работ на консультациях проводится разбор допущенных студентами ошибок.

### *12.2. Формы организации самостоятельной работы обучающихся (темы, выносимые для самостоятельного изучения; вопросы для самоконтроля; типовые задания для самопроверки*

Некоторые темы изучаются студентами самостоятельно. Для изучения используется приведённая в списке основная и дополнительная литература. Контроль освоения материала осуществляется при проверке контрольных работ, домашнего задания и на экзамене.

| № | Тема и часть, изучаемая (осваиваемая) самостоятельно |
|---|---|
| 1.1 | Асинхронное программирование (asyncio) |
| 1.2 | Логгирование и модуль logging |
| 1.3 | Многопоточность и многопроцессорность (threading, multiprocessing) |
| 1.4 | Работа с графическими интерфейсами (GUI) в Python (Tkinter, PyQt) |
| 1.5 | Типизация и аннотации типов (typing) |

Вопросы и задания для самоконтроля по всем темам:

1. Что такое переменная в Python и как объявить её?
2. Какие базовые типы данных существуют в Python? Приведите примеры.
3. Чем отличаются списки и кортежи в Python?
4. Что такое множества и словари в Python, и когда их следует использовать?
5. Как определить функцию в Python и какие у неё параметры?
6. Как работают условные операторы (if, elif, else) в Python?
7. В чем разница между циклами for и while? Приведите примеры их использования.
8. Что такое модуль в Python и как подключить библиотеку с помощью import?
9. Как открыть и закрыть файл в Python, а также прочитать из него данные?
10. Что такое лямбда-функции, и как их использовать в Python?
11. Что такое comprehensions (list, set, dictionary) и в чем их преимущества?
12. Чем отличаются итераторы и генераторы в Python? Приведите примеры.
13. Как обрабатывать исключения в Python с помощью блоков try, except?
14. Для чего используется оператор * в Python, и как он работает в функциях?
15. Как работать с датами и временем в Python с использованием модуля datetime?
16. Что такое регулярные выражения и как использовать их в Python?
17. Как отправить HTTP-запрос с помощью модуля requests в Python?
18. Что такое наследование в Python, и как оно применяется в ООП?
19. Что такое инкапсуляция и как она реализуется в Python?
20. В чем суть полиморфизма и как его реализовать в Python?
21. Что такое декораторы в Python и как они могут быть использованы?
22. Какие магические методы существуют в Python и как они помогают в программировании?
23. Что такое паттерн Singleton и как его реализовать в Python?

***12.3. Краткий терминологический словарь***

| | |
|---|---|
| Лямбда-функции | это анонимные функции, которые могут быть определены с использованием ключевого слова lambda. Они обычно используются для кратких функций, которые могут быть переданы в качестве аргумента другим функциям. Пример: lambda x: x + 1. |
| Comprehensions | это конструкция в Python для создания новых коллекций (списков, множеств, словарей) с помощью компактного синтаксиса. Например, list |

| | |
|---|---|
| | comprehension позволяет создать список на основе другого списка: `[x * 2 for x in range(5)]` создаст список `[0, 2, 4, 6, 8]`. |
| Итераторы | это объекты, которые позволяют поочередно получать элементы коллекции. Итератор реализует методы `__iter__()` и `__next__()`. Примером итератора может служить объект типа списка или множества, который можно использовать в цикле `for`. |
| Инкапсуляция | это принцип объектно-ориентированного программирования, заключающийся в сокрытии внутренней реализации объекта и предоставлении доступа к данным только через публичные методы. Это позволяет защищать данные от ненадежных изменений и контролировать доступ к ним. |

1.