|  |  |
| --- | --- |
| **К Г Э У** | МИНИСТЕРСТВО ВЫСШЕГО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ»**  (ФГБОУ ВО «КГЭУ») |

**Кафедра «Информатики и информационных управляющих систем»**

ОТЧЕТ

Дисциплина «Алгоритмизация и программирование»

Выполнил: Соловьёв Л.А.  
Группа: ПИ-1-22  
Преподаватель: Шорина Т.В.

Казань, 2024

***Постановка задачи***

1. Изучение работы функций библиотеки Matplotlib

Matplotlib - это библиотека для создания качественных графиков в Python. Она широко используется для визуализации данных в научных и инженерных приложениях, а также для создания графиков для публикаций и презентаций.

**Основные команды модуля Matplotlib**

|  |  |
| --- | --- |
| mpl.plot(x, y) | Создает линейный график на основе данных x и y. |
| mpl.scatter(x, y) | Создает диаграмму рассеяния на основе данных x и y. |
| mpl.bar(x, height) | Создает столбчатую диаграмму на основе данных x и height. |
| mpl.hist(x, bins) | Создает гистограмму распределения на основе данных x. |
| mpl.boxplot(data) | Создает ящик с усами на основе данных в data. |
| mpl.pie(x, labels) | Создает круговую диаграмму на основе данных x с метками labels. |
| mpl.hexbin() | Построение шестиугольного графика. |
| mpl.bar();  mpl.plot(); | Объединение двух графиков. |
| ax.bar(); | Сложные визуализации с накоплением |
| mpl.legend() | Добавляет легенду на график. |

**Примеры обработки данных с использованием модуля**

**Задача 1.** Написать программу, которая выводит график y = x2.

Алгоритм выполнения:

Для начала, импортируем библиотеку matplotlib с помощью команды import matplotlib.pyplot as mpl. Затем создаем переменные x и y c помощью итерируемой функции range() и list(), map(). После этого, используя функцию mpl.plot() создаем график функции, mpl.title() для создания заголовка графика, mpl.xlabel() и mpl.ylabel() для создания подписей для осей. Затем, выводим полученный график с помощью функции mpl.show().

Ожидаемый итог:

После выполнения кода ожидается, что на выходе мы получим график функции y = x2.

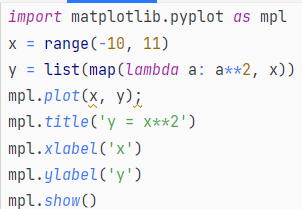


Рисунок 1.1 – Код к задаче №1

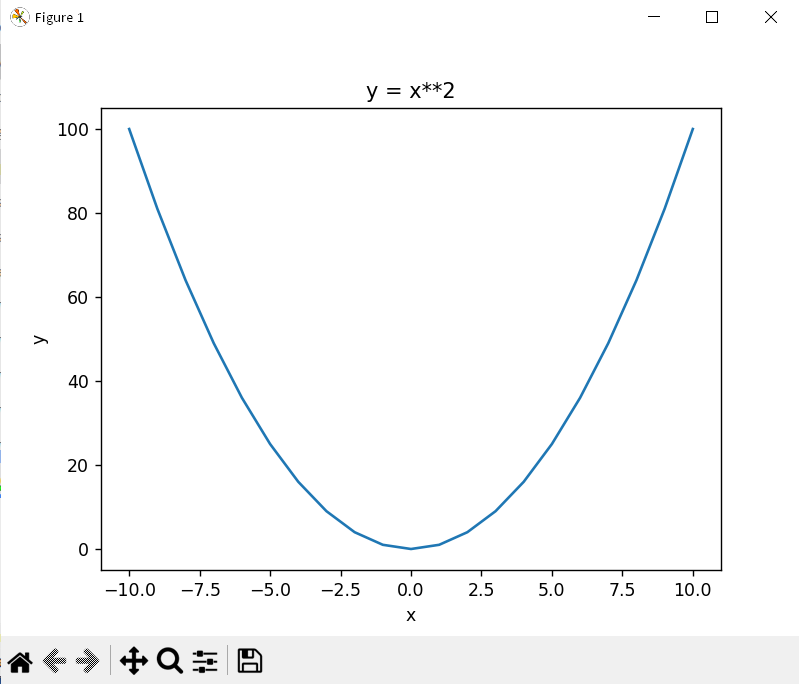


Рисунок 1.2 – Вывод к задаче №1

**Задача 2.** Написать программу, которая выводит столбчатую диаграмму

Алгоритм выполнения:

Для начала, импортируем библиотеку matplotlib с помощью команды import matplotlib.pyplot as mpl. Затем создаем переменные cats и counts. После этого, используя функцию mpl.bar() создаем столбчатую диаграмму, mpl.title() для создания заголовка графика, mpl.xlabel() и mpl.ylabel() для создания подписей для осей. Затем, выводим полученный график с помощью функции mpl.show().

Ожидаемый итог:

После выполнения кода ожидается, что на выходе мы получим столбчатую диаграмму количества фруктов.

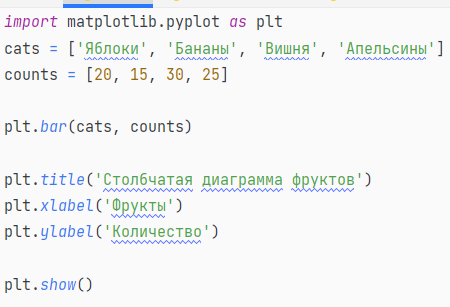


Рисунок 2.1 – Код к задаче №2

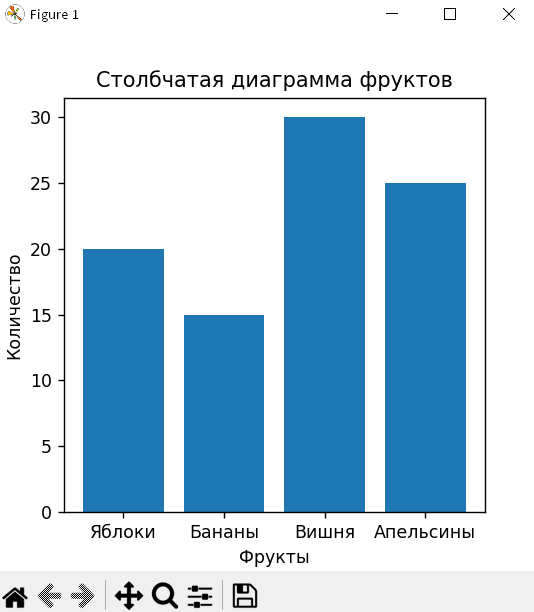


Рисунок 2.2 – Вывод к задаче №2

**Задача 3.** Написать программу, которая выводит точечную диаграмму

Алгоритм выполнения:

Для начала, импортируем библиотеку matplotlib с помощью команды import matplotlib.pyplot as mpl. Затем создаем переменные x и y. После этого, используя функцию mpl.plot(), необязательным параметрами marker и color, создаем диаграмму c отмеченными точками красного цвета, mpl.title() для создания заголовка графика, mpl.xlabel() и mpl.ylabel() для создания подписей для осей. Затем, выводим полученный график с помощью функции mpl.show().

Ожидаемый итог:

После выполнения кода ожидается, что на выходе мы получим график красного цвета с отмеченными точками и подписанной первой точкой.

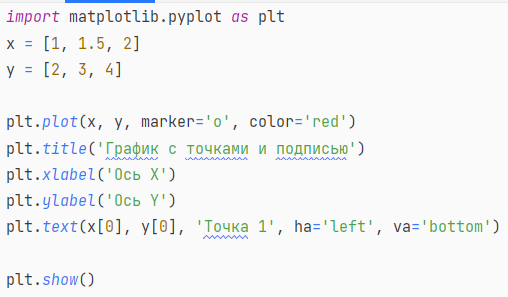


Рисунок 3.1 – Код к задаче №3

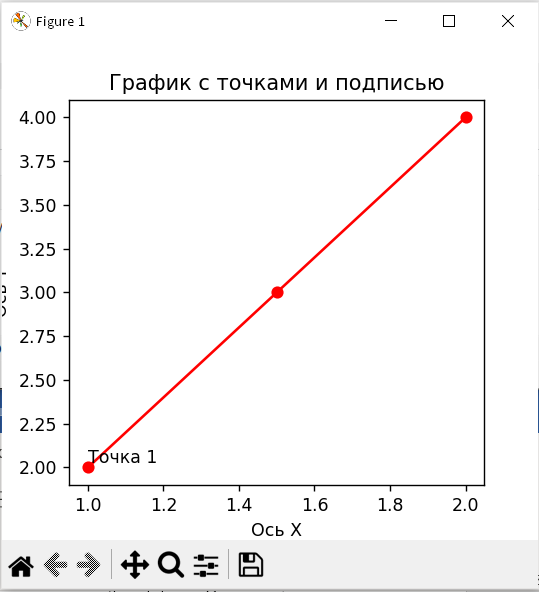


Рисунок 3.2 – Вывод к задаче №3

**Задача 4.** Написать программу, которая выводит точечную диаграмму

Алгоритм выполнения:

Для начала, импортируем библиотеку matplotlib с помощью команды import matplotlib.pyplot as mpl, а так же дополнительные библиотеки mpl\_toolkits.mplot3d и numpy. Затем создаем переменные fig(основная фигура) и ax(3d оси). После этого создаем переменные radius, u, v, x, y, z необходимые для математического описания сферы, затем используя функцию ax.plot\_surface() отображаем созданные переменные со значениями функций, mpl.title() для создания заголовка графика, ax.set\_xlabel(), ax.set\_ylabel() и ax.set\_zlabel() для создания подписей для осей. Затем, выводим полученный график с помощью функции mpl.show().

Ожидаемый итог:

После выполнения кода ожидается, что на выходе мы получим график красного цвета с отмеченными точками и подписанной первой точкой.

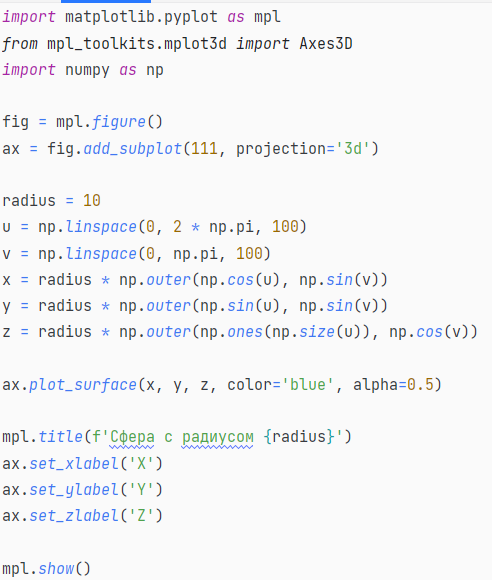


Рисунок 4.1 – Код к задаче №4

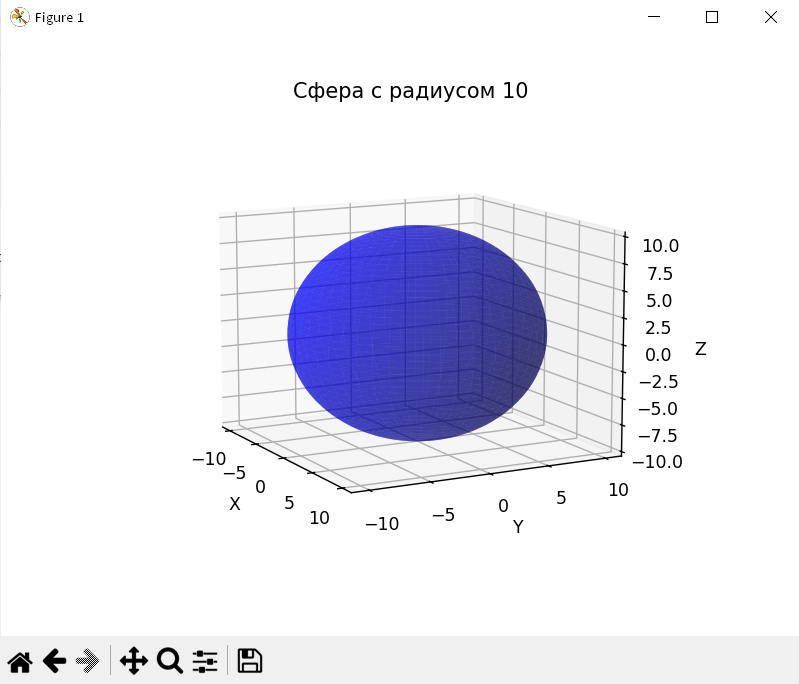


Рисунок 4.2 – Вывод к задаче №4

**Задача 4.** Написать программу, которая выводит круговую диаграмму

Алгоритм выполнения:

Для начала, импортируем библиотеку matplotlib с помощью команды import matplotlib.pyplot as mpl. Затем создаем переменные languages и popularity. Затем используя функцию mpl.pie() отображаем созданные переменные со значениями функций, mpl.title() для создания заголовка графика и mpl.axis() для задания круговой диаграммы, а не овальной. Затем, выводим полученный график с помощью функции mpl.show().

Ожидаемый итог:

После выполнения кода ожидается, что на выходе мы получим график красного цвета с отмеченными точками и подписанной первой точкой.

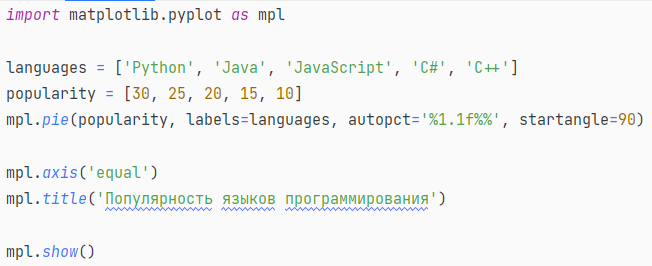


Рисунок 5.1 – Код к задаче №5

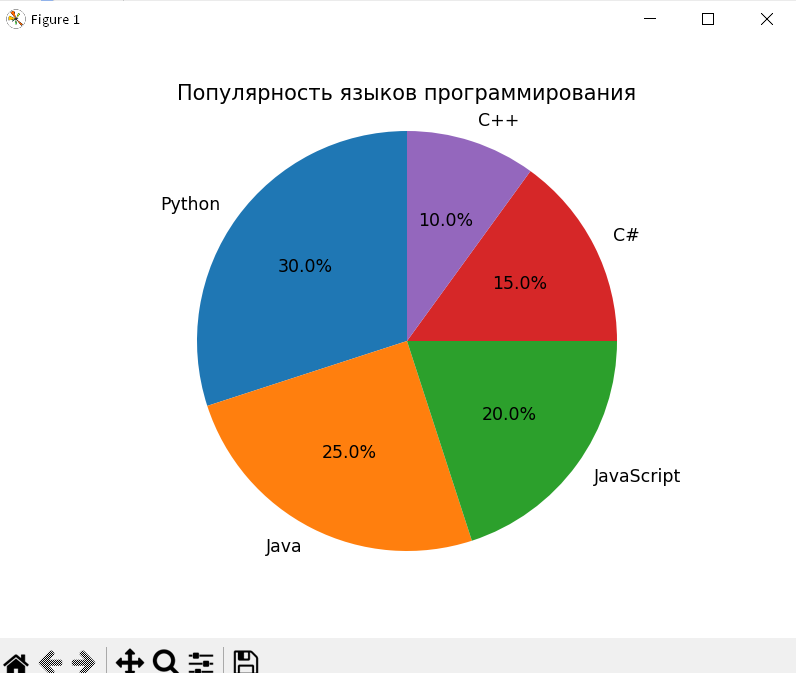


Рисунок 5.2 – Вывод к задаче №5

**Вывод**

Библиотека Matplotlib предоставляет широкий спектр возможностей для визуализации данных в Python. С ее помощью можно создавать различные типы графиков, настраивать их внешний вид и добавлять различные элементы управления, такие как подписи к осям, заголовки и легенды. Это делает Matplotlib мощным инструментом для исследования данных, анализа и визуализации результатов.

**Список использованных источников**

1. <https://matplotlib.org/cheatsheets/>