|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

|  |  |
| --- | --- |
| **ФАКУЛЬТЕТ** | **ИУК «Информатика и управление»** |
| **КАФЕДРА** | **ИУК4 «Программное обеспечение ЭВМ,** |
| **информационные технологии»** | |

**ЛАБОРАТОРНАЯ РАБОТА №1**

**«Основные операции над множествами»**

**ДИСЦИПЛИНА: «Дискретная математика»**



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил: студент гр. ИУК4-31Б | |  |  | ( | Суриков Н. С. | ) |
|  |  |  | (подпись) |  | (Ф.И.О.) |  |
| Проверил: | |  |  | ( | Никитенко У. В. | ) |
|  |  |  | (подпись) |  | (Ф.И.О.) |  |

|  |  |
| --- | --- |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | |
|  | - Балльная оценка:  - Оценка: |

**Цель:** изучение способов задания множеств, приобретение практических навыков в выполнении операций над множествами, визуализация результатов.

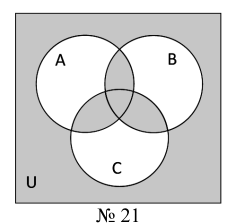
**Задачи:**

1. Разработать приложение визуализации операций над множествами

**Вариант 21**

***Задание:*** *Для конечных множеств, по заданному выражению строится диаграмма Эйлера-Венна, заштрихованная (выделенная определенным цветом) часть диаграммы должна соответствовать выражению.*

*В качестве тестового примера использовать Задание 1 индивидуального варианта ЛР\_1(I).*

**

**Этапы выполнения:**

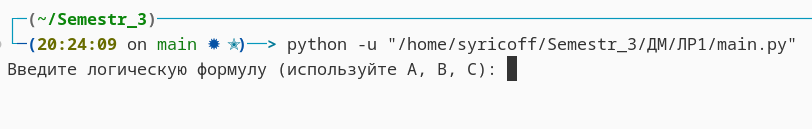
*Этап 1 — Выбор средств для решения задачи:*

Для решения задачи был выбран язык Python и библиотека matplotlib\_venn, так как синтаксис языка достаточно прост, а библиотека предоставляет готовые решения для быстрой разработки.

Установка: pip install matplotlib\_venn

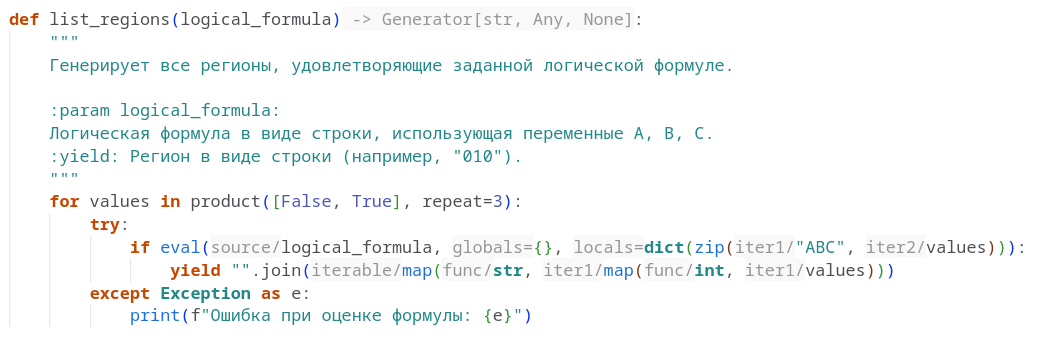
*Этап 2 — Формирование формулы:*

*Программа способна формировать диаграммы по введённой формуле, при условии что эта формула написана с использованием операторов и синтаксиса языка Python.*



*Этап 3 — Обработка формулы:*

Для обработки была написана функция обработчик которая генерирует регионы, которые должны быть закрашены.



*Этап 4 — Вывод диаграммы:*

Для вывода диаграммы используются средства библиотеки matplotlib\_venn, создание происходит в основной функции plot\_diagram.

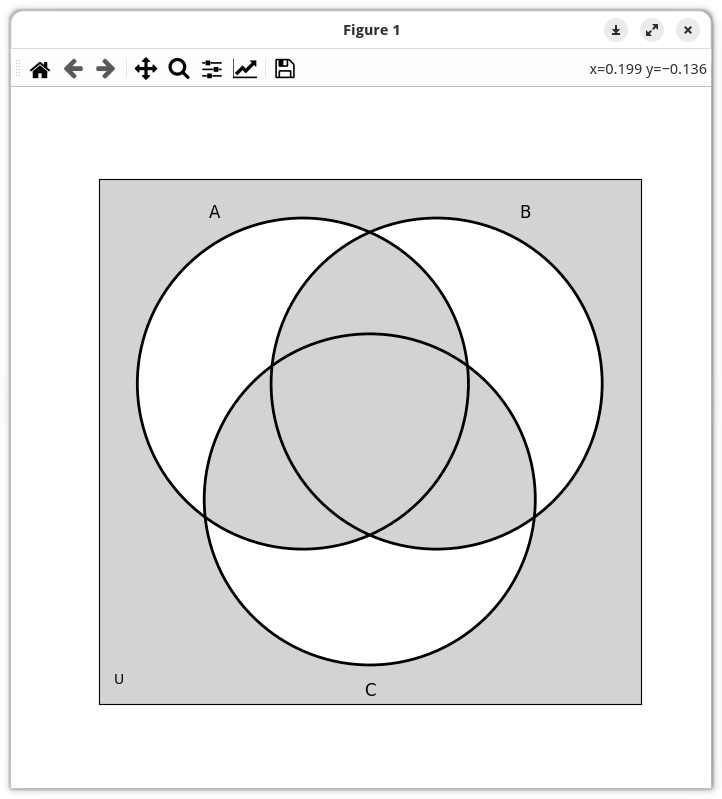


***Результат работы программы:***

*Ввод:*

Введите логическую формулу (используйте A, B, C): (A and B) or (A and C) or (B and C) or not(A or B or C)

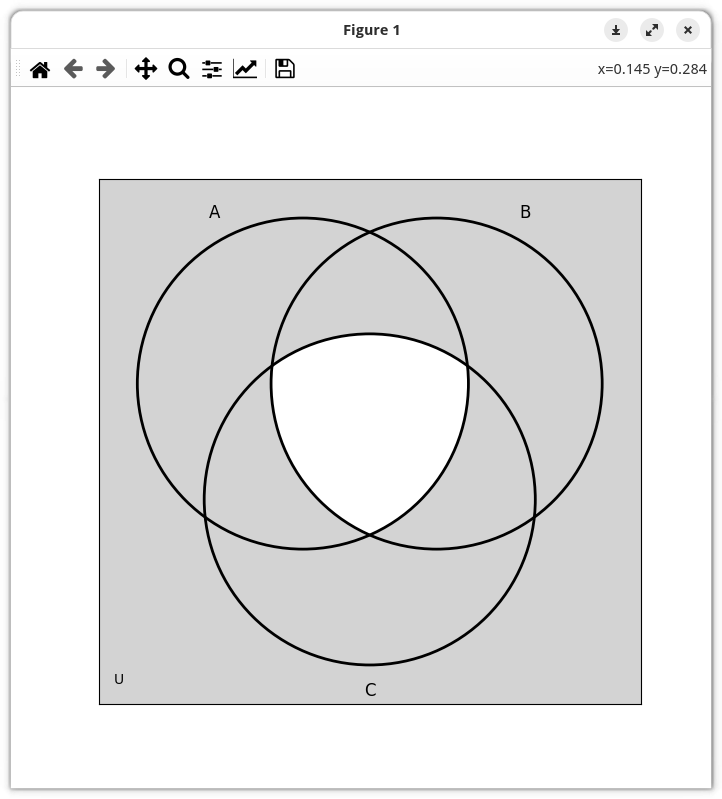
*Вывод:*

**

*Ввод:*

Введите логическую формулу (используйте A, B, C): not(A and B and C)

*Вывод:*

**

**Листинг программы:**

1 from matplotlib import pyplot as plt

2 from matplotlib\_venn import venn3, venn3\_circles

3 from itertools import product

4

5

6 def get\_logical\_formula():

7 """

8 Запрашивает у пользователя ввод логической формулы.

9

10 :return: Логическая формула в виде строки.

11 """

12 formula = input("Введите логическую формулу (используйте A, B, C): ")

13 return formula

14

15

16 def list\_regions(logical\_formula):

17 """

18 Генерирует все регионы, удовлетворяющие заданной логической формуле.

19

20 :param logical\_formula:

21 Логическая формула в виде строки, использующая переменные A, B, C.

22 :yield: Регион в виде строки (например, "010").

23 """

24 for values in product([False, True], repeat=3):

25 try:

26 if eval(logical\_formula, {}, dict(zip("ABC", values))):

27 yield "".join(map(str, map(int, values)))

28 except Exception as e:

29 print(f"Ошибка при оценке формулы: {e}")

30

31

32 def plot\_diagram(logical\_formula):

33 """

34 Создает диаграмму Венна по заданной логической формуле.

35

36 :param logical\_formula:

37 Логическая формула в виде строки, использующая переменные A, B, C.

38 """

39 figure, ax = plt.subplots(figsize=(7, 7))

40

41 venn\_diagram = venn3(

42 alpha=1.0,

43 subsets=(1,) \* 7,

44 set\_colors=("white",) \* 3,

45 subset\_label\_formatter=lambda x: "",

46 )

47

48 venn3\_circles(subsets=[1] \* 7, linestyle="solid")

49

50 regions\_to\_color = list(list\_regions(logical\_formula))

51

52 for region in regions\_to\_color:

53 if region == "000":

54 ax.set\_facecolor("lightgrey")

55 else:

56 patch = venn\_diagram.get\_patch\_by\_id(region)

57 if patch:

58 patch.set\_color("lightgrey")

59 else:

60 print(f"Регион {region} не найден")

61

62 plt.annotate("U", (-0.660, -0.675))

63 plt.axis("on")

64 plt.show()

65

66

67 if \_\_name\_\_ == "\_\_main\_\_":

68 logical\_formula = get\_logical\_formula()

69 plot\_diagram(logical\_formula)

**Вывод:** В ходе работы были изучены способы задания множеств, приобретены практические навыки в выполнении операций над множествами.