**ФГАОУ ВО «МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Лабораторная работа №3**

Организация циклов

**Вариант № 28**

По дисциплине:

Основы программирования

Выполнил студент 1-го курса группы 243-323

Онищенко А. А.

Проверил

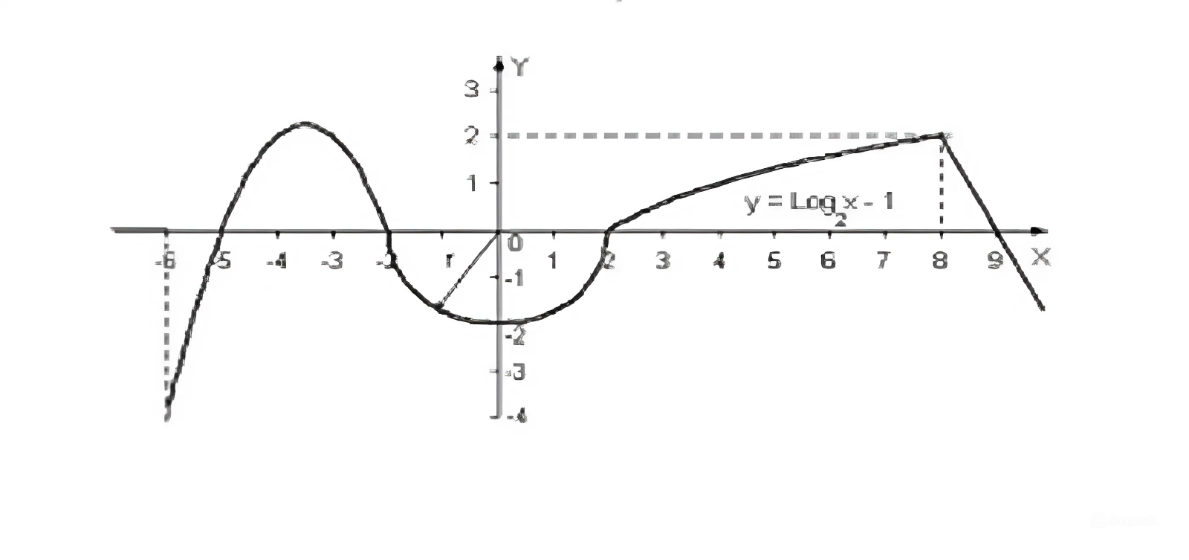
\_\_\_\_\_\_\_\_\_\_\_Никишина И. Н.

**Москва, 2025**

**Задание 1**

**Постановка задачи**

Вычислить и вывести на экран в виде таблицы значения функции, заданной графически, на интервале от *Xнач* до *Xкон* с шагом *dx*. Интервал и шаг задать таким образом, чтобы проверить все ветви программы. Таблицу снабдить заголовком и шапкой.



**Теоретическая часть**

Для решения задачи используется обновленная программа из первого задания второй лабораторной работы. В нее добавляется оператор цикла, проверяющий условие при котором программа продолжит выполняться по кругу.

**Описание программы**

Программа написана на алгоритмическом языке Python 3.6, реализована в среде ОС Windows 10 и состоит из частей, отвечающих за ввод данных, вычисление и представление данных на экране монитора.

**Описание алгоритма**

1. Ввести значения переменных *Xbeg, Xend, Dx*.
2. Присвоить текущему значению *Xt* начальное значение *Xbeg*.
3. Вычислить значение функции и вывести в виде строки таблицы.
4. Вычислить новое значение *Xt*.
5. Повторить с пункта 3, если *Xt < Xend.*
6. Завершить программу.

**Описание входных и выходных данных**

Входные данные поступают от пользователя с клавиатуры, выходные выводятся на экран. Данные имеют тип float.

**Листинг программы**

*from* math *import* \*  
  
*print*("Enter Xbeg, Xend, Dx")  
  
xb = *float*(*input*("Xbeg = "))  
xe = *float*(*input*("Xend = "))  
dx = *float*(*input*("Dx = "))  
  
*print*(f"Xbeg = {xb:7.2f} Xend = {xe:7.2f} Dx = {dx:7.2f}\n")  
  
xt = xb  
  
*print*("+---------+---------+")  
*print*("| X | Y |")  
*print*("+---------+---------+")  
  
*while* xt < xe:  
 *if* -6 <= xt <= -2:  
 y = (-8/9) \* ((xt + 3.5) \*\* 2) + 2  
 *elif* -2 <= xt < 2:  
 y = -sqrt(4 - xt \*\* 2)  
 *elif* 2 <= xt < 8:  
 y = log(xt, 2) - 1  
 *else*:  
 y = -2 \* xt + 18  
 *print*(f"| {xt:7.2f} | {y:7.2f} |")  
 xt += dx  
*print*("+---------+---------+")

**Результаты и тестовые кейсы**

Xbeg = -6

Xend = 10

Dx = 0.5

Xbeg = -6.00 Xend = 10.00 Dx = 0.50

+---------+---------+

| X | Y |

+---------+---------+

| -6.00 | -3.56 |

| -5.50 | -1.56 |

| -5.00 | 0.00 |

| -4.50 | 1.11 |

| -4.00 | 1.78 |

| -3.50 | 2.00 |

| -3.00 | 1.78 |

| -2.50 | 1.11 |

| -2.00 | 0.00 |

| -1.50 | -1.32 |

| -1.00 | -1.73 |

| -0.50 | -1.94 |

| 0.00 | -2.00 |

| 0.50 | -1.94 |

| 1.00 | -1.73 |

| 1.50 | -1.32 |

| 2.00 | 0.00 |

| 2.50 | 0.32 |

| 3.00 | 0.58 |

| 3.50 | 0.81 |

| 4.00 | 1.00 |

| 4.50 | 1.17 |

| 5.00 | 1.32 |

| 5.50 | 1.46 |

| 6.00 | 1.58 |

| 6.50 | 1.70 |

| 7.00 | 1.81 |

| 7.50 | 1.91 |

| 8.00 | 2.00 |

| 8.50 | 1.00 |

| 9.00 | 0.00 |

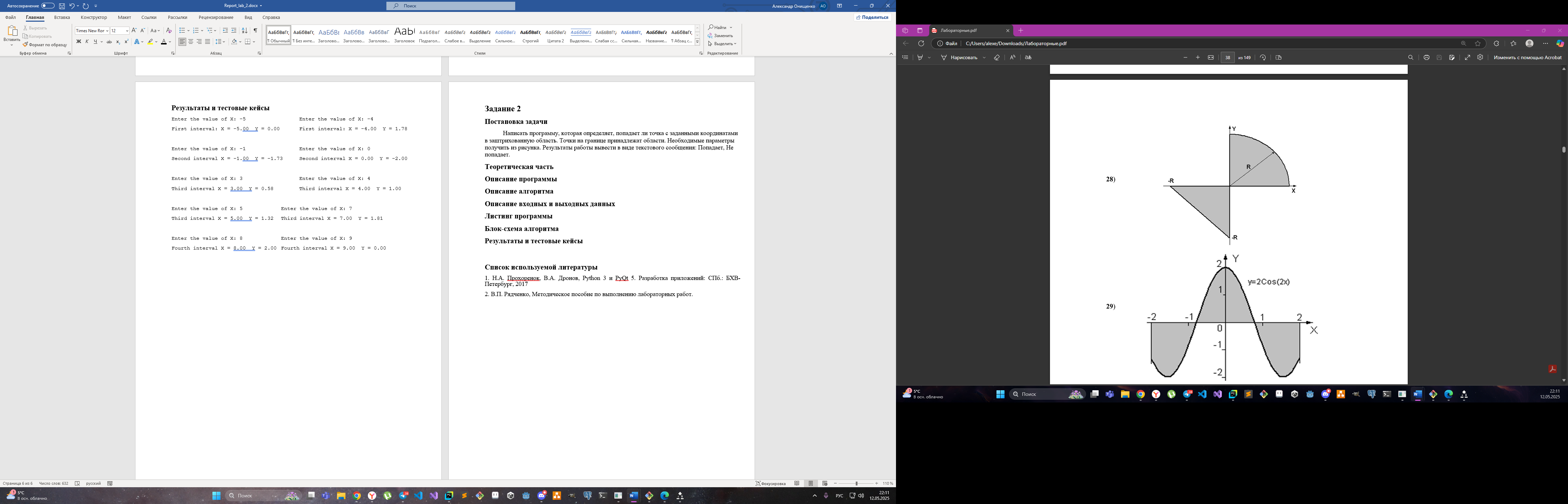
| 9.50 | -1.00 |

+---------+---------+

**Задание 2**

**Постановка задачи**

Для 10 выстрелов, координаты которых задаются генератором случайных чисел, вывести текстовые сообщения о попадании в мишень.



**Теоретическая часть**

Для решения задачи используется обновленная программа из второго задания второй лабораторной работы. В нее добавляется оператор цикла с параметром для прохождения по значениям точек заданных с помощью функции *uniform* модуля random, в качестве параметров минимального и максимального значения *uniform* принимается R для X и для Y.

**Описание программы**

Программа написана на алгоритмическом языке Python 3.6, реализована в среде ОС Windows 10 и состоит из частей, отвечающих за ввод данных, вычисление и представление данных на экране монитора.

**Описание алгоритма**

1. Ввести значение радиуса R в качестве ограничения.
2. Задать количество прохождений цикла и переменную счетчик.
3. В каждом цикле генерировать X и Y .
4. Проверить на принадлежность к области.
5. Вывести результат проверки принадлежности точки.
6. Вернуться к пункту 3.
7. Завершить программу когда счетчик вышел за пределы.

**Описание входных и выходных данных**

Входные данные поступают с клавиатуры, а выходные выводятся на монитор для просмотра. Входные и выходные данные имеют тип float.

**Листинг программы**

*from* math *import* \*  
*from* random *import* \*  
  
r = *float*(*input*("Enter R: "))  
flag = 0  
  
*print*("+---------+---------+---------+")  
*print*("| X | Y | Result |")  
*print*("+---------+---------+---------+")  
  
*for* n *in range*(10):  
 x = uniform(-r, r)  
 y = uniform(-r, r)  
 *if* x >= -r *and* x <= 0 *and* y >= -x - r *and* y <= 0 \  
 *or* x >= 0 *and* x <= r *and* y <= sqrt(r \*\* 2 - x \*\* 2) *and* y >= 0:  
 flag = 1  
 result = "Yes"  
 *else*:  
 flag = 0  
 result = "No"  
  
 *print*(f"| {x:7.2f} | {y:7.2f} | {result:^7} |")  
  
*print*("+---------+---------+---------+")

**Результаты и тестовые кейсы**

Enter R: 5

+---------+---------+---------+

| X | Y | Result |

+---------+---------+---------+

| 0.10 | 3.48 | Yes |

| 1.26 | -2.94 | No |

| -0.19 | -2.32 | Yes |

| -4.48 | -4.02 | No |

| -2.06 | 2.25 | No |

| -0.06 | -0.05 | Yes |

| 0.35 | -2.88 | No |

| -1.70 | 3.14 | No |

| -1.84 | -3.54 | No |

| 0.16 | 1.22 | Yes |

+---------+---------+---------+

Enter R: 7

+---------+---------+---------+

| X | Y | Result |

+---------+---------+---------+

| -0.60 | -1.14 | Yes |

| -3.11 | 2.79 | No |

| 6.07 | 1.91 | Yes |

| -6.41 | -5.99 | No |

| -1.63 | -3.36 | Yes |

| -4.39 | -1.15 | Yes |

| 0.84 | -1.07 | No |

| 4.58 | -0.66 | No |

| 0.24 | 4.06 | Yes |

| -5.51 | 2.01 | No |

+---------+---------+---------+

**Задание 3**

**Постановка задачи**

Вычислить и вывести на экран в виде таблицы значения функции, заданной с помощью ряда Тейлора, на интервале от *Xbeg* до *Xend* с шагом *Dx* с точностью ε.

Таблицу снабдить заголовком и шапкой. Каждая строка таблицы должны содержать значение аргумента, значение функции и количество просуммированных членов ряда.

**Теоретическая часть**

**Описание программы**

Программа написана на алгоритмическом языке Python 3.6, реализована в среде ОС Windows 10 и состоит из частей, отвечающих за ввод данных, вычисление и представление данных на экране монитора.

**Описание алгоритма**

**Описание входных и выходных данных**

Входные данные поступают от пользователя с клавиатуры, выходные выводятся на экран. Данные имеют тип float.

**Листинг программы**

**Блок-схема алгоритма**

**Результаты и тестовые кейсы**

**Список используемой литературы**

1. Н.А. Прохоренок, В.А. Дронов, Python 3 и PyQt 5. Разработка приложений: СПб.: БХВ- Петербург, 2017

2. В.П. Рядченко, Методическое пособие по выполнению лабораторных работ.