

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций
Институт цифрового развития
ОТЧЁТ**

по лабораторной работе №2.2

Дисциплина: «Основы кроссплатформенного программирования»

Тема: «Условные операторы и циклы в языке Python

»

Выполнил: студент 1 курса

группы ИВТ-б-о-21-1

Толубаев Рамиль Ахметович

Ставрополь 2022

Выполнение работы.

1. Создал репозиторий в GitHub «rep 2.2» в который добавил .gitignore, который дополнил правила для работы с IDE PyCharm с ЯП Python, выбрал лицензию MIT, клонировал его на лок. сервер и организовал в соответствии с моделью ветвления git-flow.

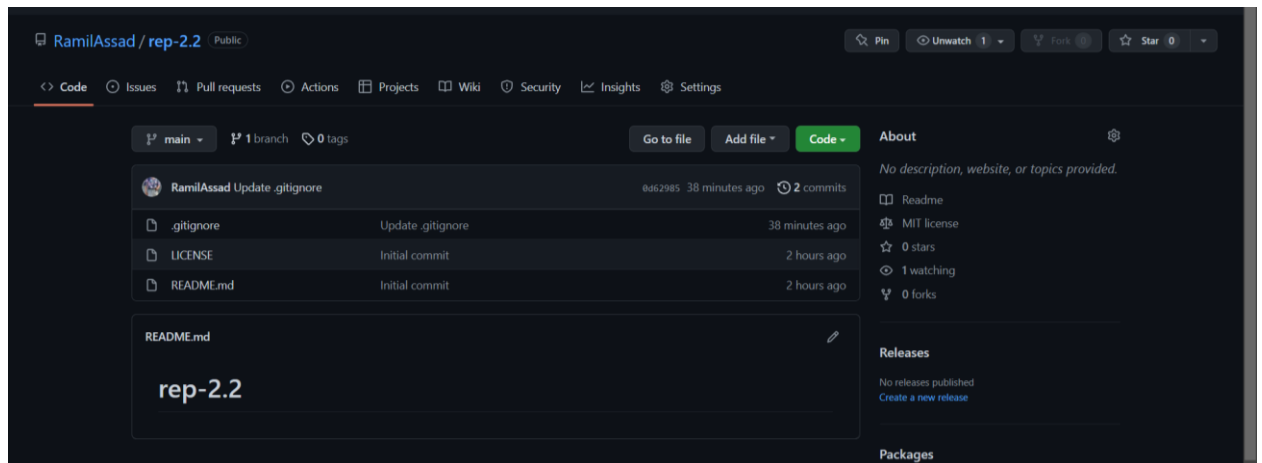


Рисунок 1.1 Создание репозитория

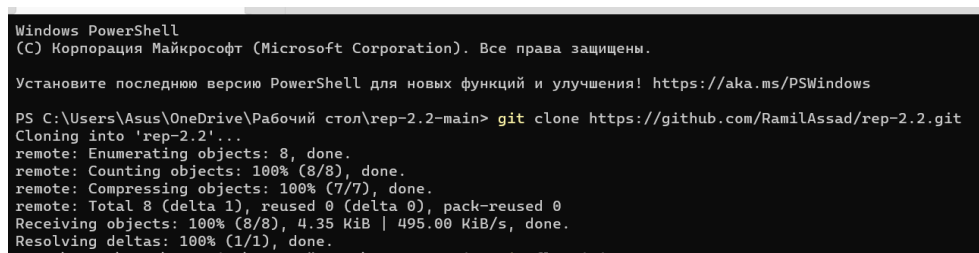


Рисунок 1.2 Клонирование и организация репозитория согласно модели ветвления git-flow

Пример 1.

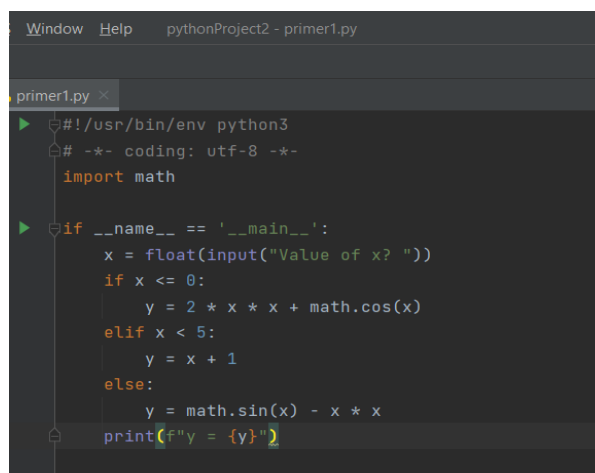


Рисунок 2.1 Код из методички

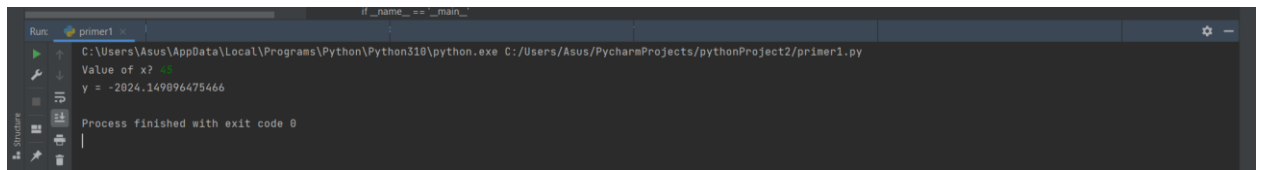


Рисунок 2.2 Результат кода сверху

Пример 2.

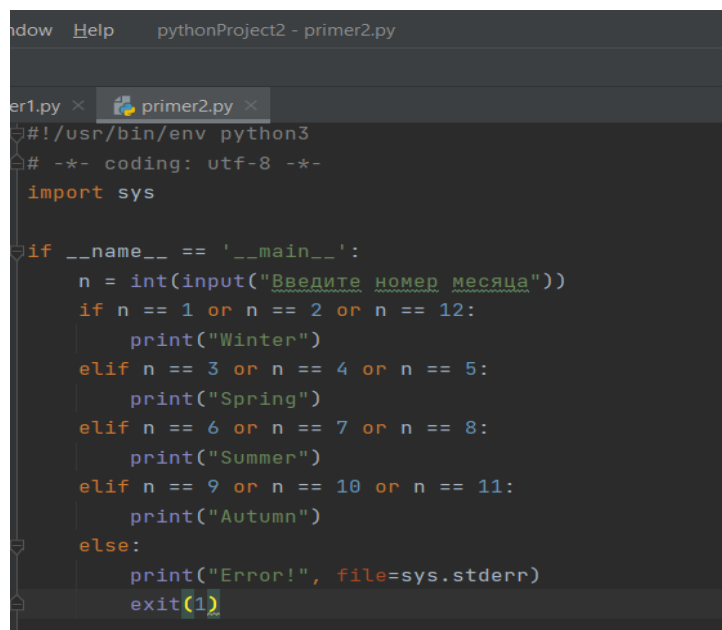


Рисунок 2.3 Код из методички

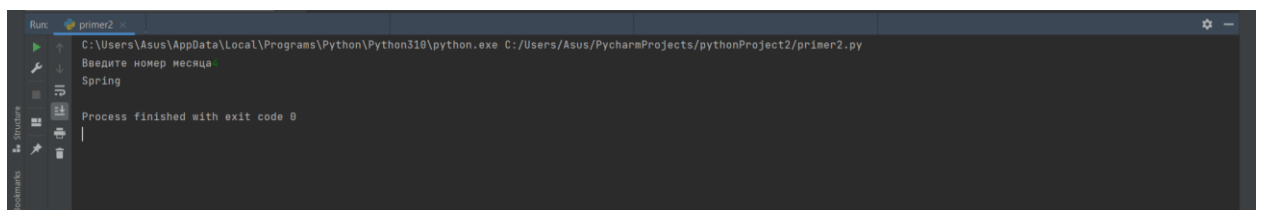


Рисунок 2.4 Результат кода сверху

Пример 3.

```
VCS Window Help pythonProject2 - primer3.py

primer3.py x
1  ▶  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  import math
4
5  ▶  if __name__ == '__main__':
6      n = int(input("Value of n? "))
7      x = float(input("Value of x? "))
8
9      S = 0.0
10     for k in range(1, n + 1):
11         a = math.log(k * x) / (k * k)
12         S += a
13
14     print(f"S = {S}")
```

Рисунок 2.5 Код из методички

```
Run primer3 x
C:\Users\Asus\AppData\Local\Programs\Python\Python310\python.exe C:/Users/Asus/PycharmProjects/pythonProject2/primer3.py
Value of n? 5
Value of x? 2
S = 3.294431456662813
Process finished with exit code 0
```

Рисунок 2.6 Результат кода сверху

Пример 4.

```
if __name__ == '__main__':
    a = float(input("Value of a? "))
    if a < 0:
        print("Illegal value of a", file=sys.stderr)
        exit(1)

    x, eps = 1, 1e-10
    while True:
        xp = x
        x = (x + a / x) / 2
        if math.fabs(x - xp) < eps:
            break

    print(f"x = {x}\nX = {math.sqrt(a)}")
```

Рисунок 2.7 Код из методички

```
Run primer4 x
C:\Users\Asus\AppData\Local\Programs\Python\Python310\python.exe C:/Users/Asus/PycharmProjects/pythonProject2/primer4.py
Value of a? 5
x = 2.23606797749979
X = 2.23606797749979
Process finished with exit code 0
```

Рисунок 2.8 Результат кода сверху

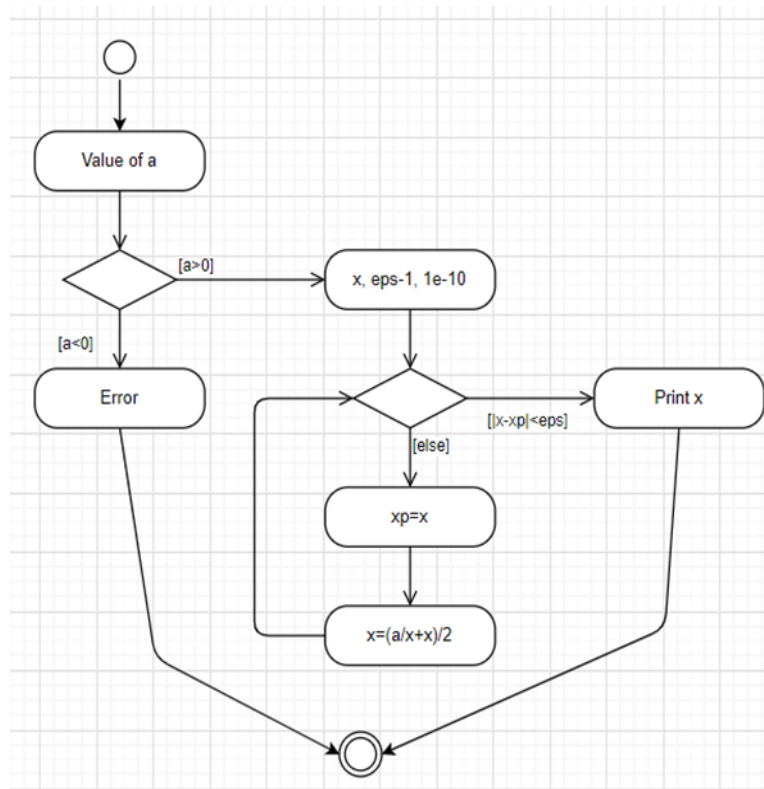


Рисунок 2.9 UML-диаграмма

Пример 5.

```

primer5.py
▶ #!/usr/bin/env python3
  # -*- coding: utf-8 -*-
  import math
  import sys

  EULER = 0.5772156649015328606
  EPS = 1e-10
  ▶ if __name__ == '__main__':
    x = float(input("Value of x?"))
    if x == 0:
      print("Illegal value of x", file=sys.stderr)
      exit(1)
    a = x
    S, k = a, 1
    while math.fabs(a) > EPS:
      a *= x * k / (k + 1) ** 2
      S += a
      k += 1
    print(f"Ei({x}) = {EULER + math.log(math.fabs(x)) + S}")
  
```

Рисунок 2.10 Код из методички

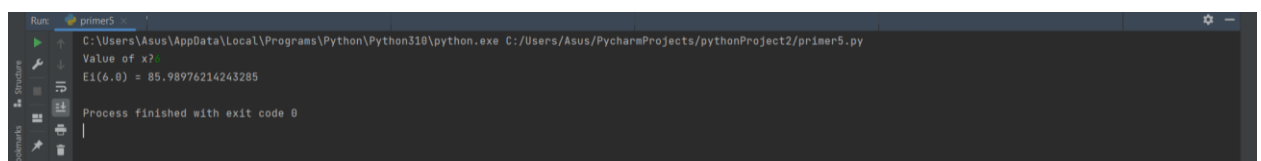


Рисунок 2.11 Результат кода сверху

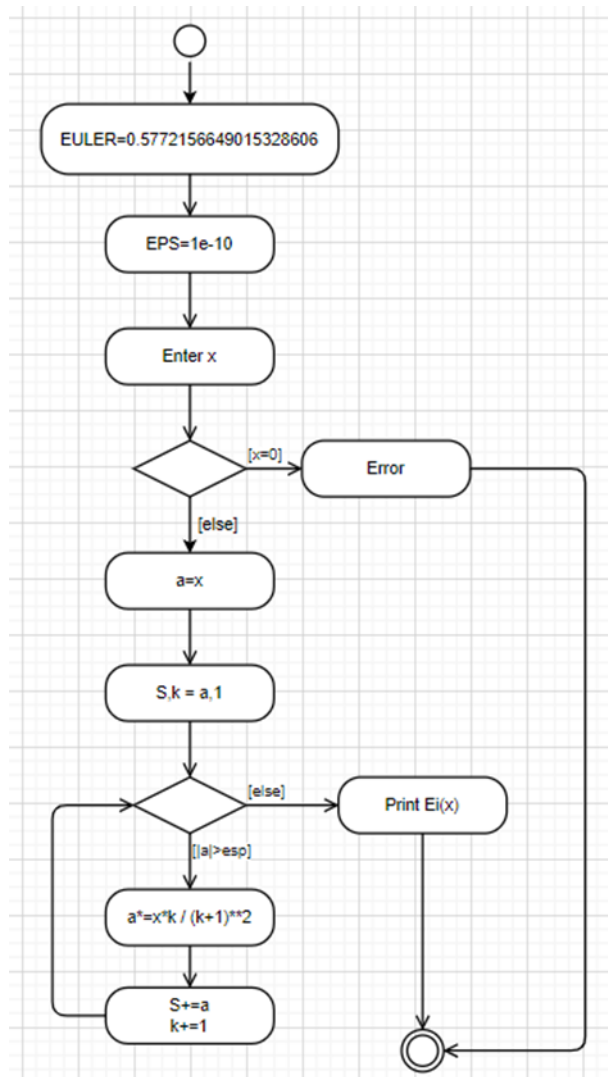


Рисунок 2.12 UML-диаграмма

Индивидуальные задания:

Задание 1. Вводится число карандашей. Вывести фразу Я купил N карандашей, согласовав слово "карандаш" с числом.

```

pythonProject2 - individual1.py
individual1.py
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  import sys
4
5  if __name__ == '__main__':
6      n = int(input("n = "))
7      if n > 10:
8          print("Ошибка", file=sys.stderr)
9          exit(1)
10
11     if n == 1:
12         b = "карандаш"
13     elif n <= 4:
14         b = "карандаша"
15     else:
16         b = "карандашей"
17     print("Я купил ", n, b)
  
```

Рисунок 3.1 Код индивидуального задания

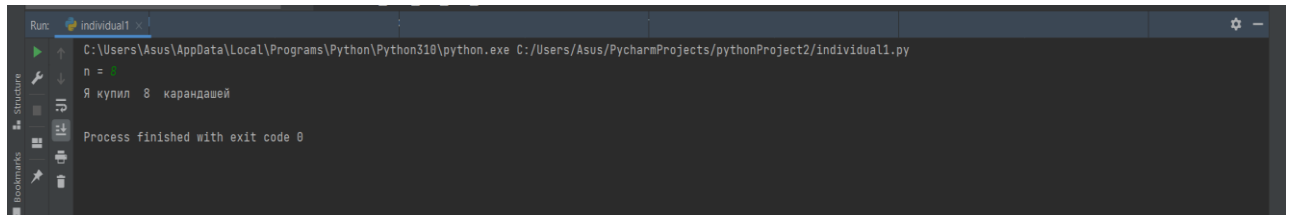


Рисунок 3.2 Результат выполнения программы

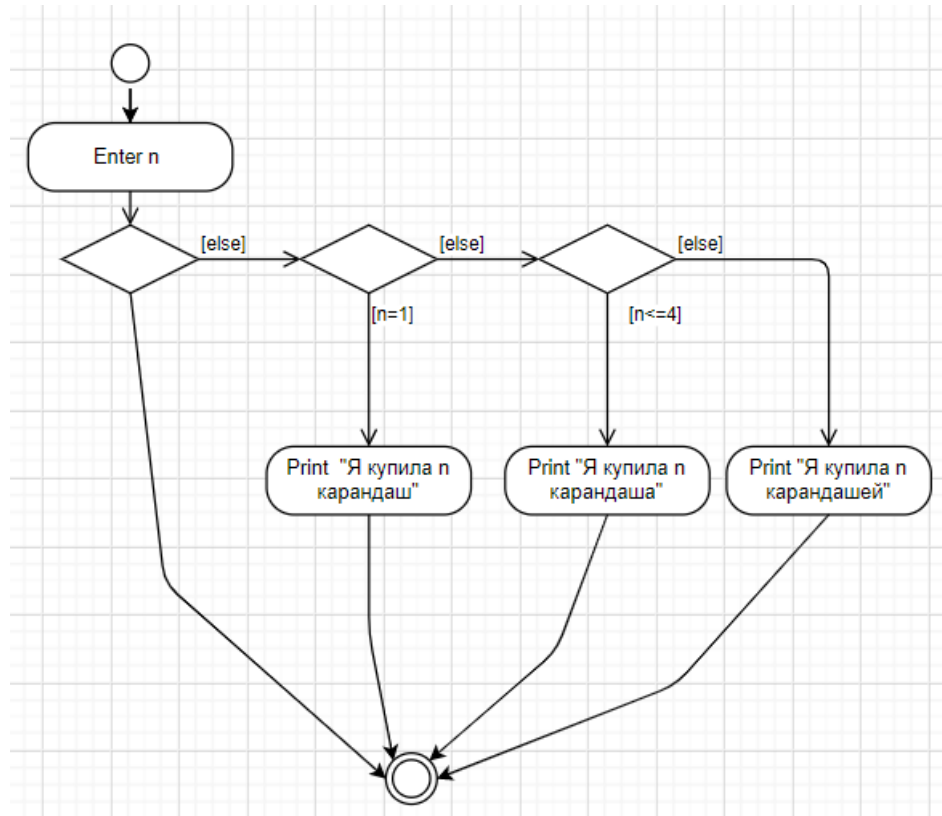


Рисунок 3.3 UML-диаграмма

Задание 2. Треугольник задан координатами своих вершин. Определить принадлежит ли данная точка треугольнику. Координаты вершин треугольника и координаты точки задать самостоятельно.

Для этого воспользуемся формулой нахождения площади треугольника по координатам вершин:

$$S = \frac{1}{2} [x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)]$$

```

VCS Window Help pythonProject2 - individual2.py

individual2.py
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 if __name__ == '__main__':
4     tr1_1 = int(input("x1 точка треугольника"))
5     tr1_2 = int(input("y1 точка треугольника"))
6     tr2_1 = int(input("x2 точка треугольника"))
7     tr2_2 = int(input("y2 точка треугольника"))
8     tr3_1 = int(input("x3 точка треугольника"))
9     tr3_2 = int(input("y3 точка треугольника"))
10    t1 = int(input("x точки"))
11    t2 = int(input("y точки"))
12    S = 0.5 * abs((tr1_1 - tr2_1) * (tr3_2 - tr2_2) - (tr3_1 - tr2_1) * (tr1_2 - tr2_2))
13    S1 = 0.5 * abs((tr1_1 - t1) * (tr3_2 - t2) - (tr3_1 - t1) * (tr1_2 - t2))
14    S2 = 0.5 * abs((tr1_1 - t1) * (tr2_2 - t2) - (tr2_1 - t1) * (tr1_2 - t2))
15    S3 = 0.5 * abs((tr2_1 - t1) * (tr3_2 - t2) - (tr3_1 - t1) * (tr2_2 - t2))
16    if S == S1 + S2 + S3:
17        print("Да")
18    else:
19        print("Нет")

```

Рисунок 3.4 Код индивидуального задания

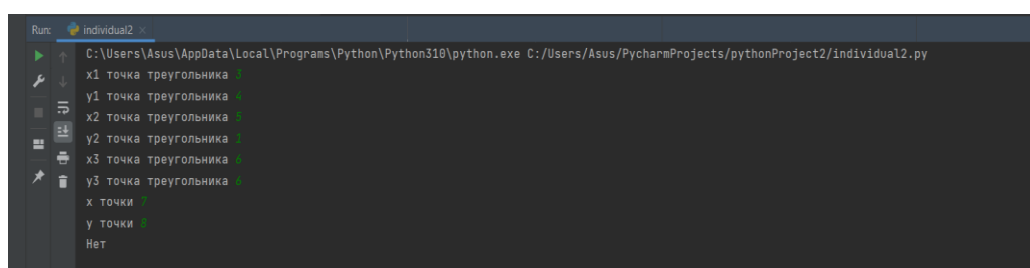


Рисунок 3.5 Результат выполнения программы

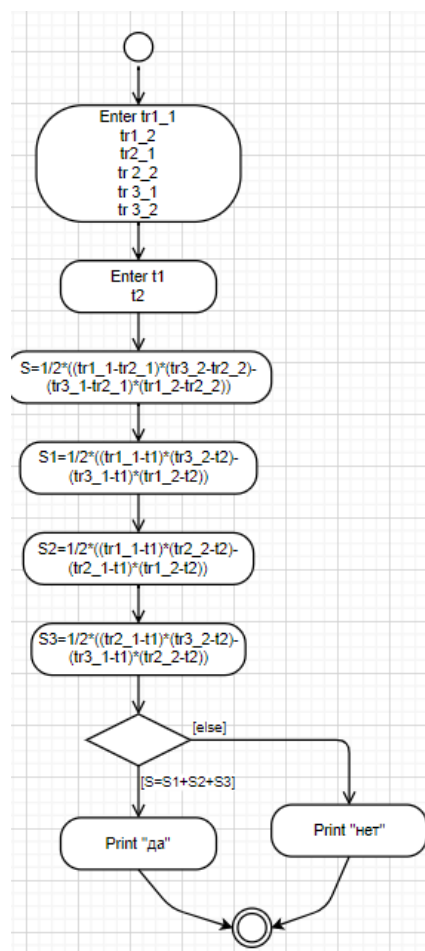
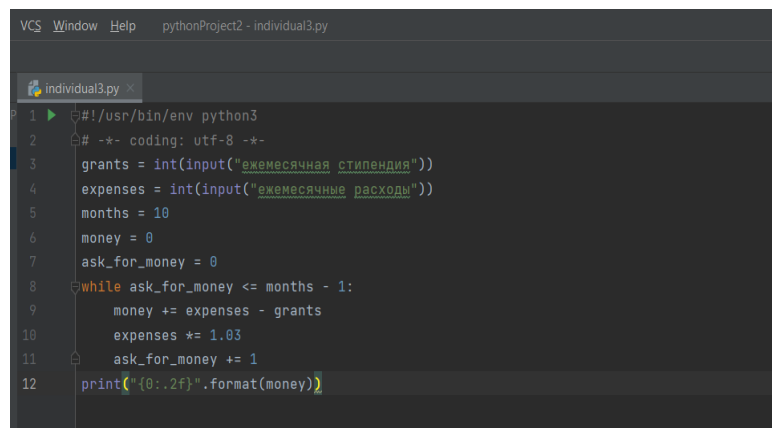


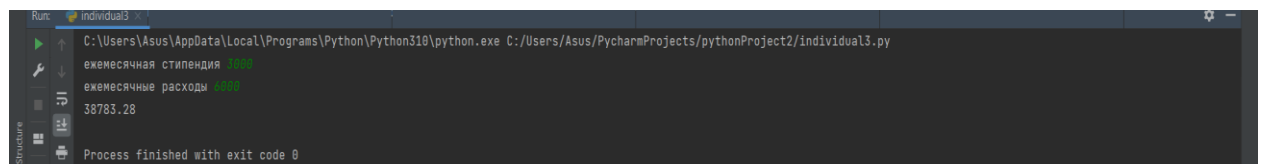
Рисунок 3.6 UML-диаграмма

Задание 3. Ежемесячная стипендия студента составляет p ., а расходы на проживание превышают стипендию и составляют p в месяц. Рост цен ежемесячно увеличивает расходы на 3%. Составьте программу расчета необходимой суммы денег, которую надо единовременно просить у родителей, чтобы можно было прожить учебный год (10 месяцев), используя только эти деньги и стипендию.



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  grants = int(input("ежемесячная стипендия"))
4  expenses = int(input("ежемесячные расходы"))
5  months = 10
6  money = 0
7  ask_for_money = 0
8  while ask_for_money <= months - 1:
9      money += expenses - grants
10     expenses *= 1.03
11     ask_for_money += 1
12 print("{0:.2f}".format(money))
```

Рисунок 3.7 Код индивидуального задания



```
Run: individual3
C:\Users\Asus\AppData\Local\Programs\Python\Python310\python.exe C:/Users/Asus/PycharmProjects/pythonProject2/individual3.py
ежемесячная стипендия 1000
ежемесячные расходы 1000
38783.28
Process finished with exit code 0
```

Рисунок 3.8 Результат выполнения программы

Ответы на вопросы

1. Для чего нужны диаграммы деятельности UML?

Позволяет наглядно визуализировать алгоритм программы.

2. Что такое состояние действия и состояние деятельности?

Состояние действия - частный вид состояния деятельности, а конкретнее – такое состояние, которое не может быть подвергнуто дальнейшей декомпозиции.

Состояние деятельности можно представлять себе как составное состояние, поток управления которого включает только другие состояния деятельности и действий.

3. Какие нотации существуют для обозначения переходов и ветвлений в диаграммах деятельности?

Переходы, ветвление, алгоритм разветвляющейся структуры, алгоритм циклической структуры.

4. Какой алгоритм является алгоритмом разветвляющейся структуры?

Алгоритм разветвляющейся структуры - это алгоритм, в котором вычислительный процесс осуществляется либо по одной, либо по другой ветви, в зависимости от выполнения некоторого условия.

5. Чем отличается разветвляющийся алгоритм от линейного?

Линейный алгоритм - алгоритм, все этапы которого выполняются однократно и строго последовательно.

Разветвляющийся алгоритм - алгоритм, содержащий хотя бы одно условие, в результате проверки которого ЭВМ обеспечивает переход на один из нескольких возможных шагов.

6. Что такое условный оператор? Какие существуют его формы?

Оператор, конструкция языка программирования, обеспечивающая выполнение определённой команды (набора команд) только при условии истинности некоторого логического выражения, либо выполнение одной из нескольких команд.

Условный оператор имеет полную и краткую формы.

7. Какие операторы сравнения используются в Python?

If, elif, else

8. Что называется простым условием? Приведите примеры.

Простым условием называется выражение, составленное из двух арифметических выражений или двух текстовых величин.

Пример: `a == b`

9. Что такое составное условие? Приведите примеры.

Составное условие – логическое выражение, содержащее несколько простых условий объединённых логическими операциями. Это операции `not`, `and`, `or`.

Пример: `(a == b or a == c)`

10. Какие логические операторы допускаются при составлении сложных условий?

not, and, or.

11. Может ли оператор ветвления содержать внутри себя другие ветвления?

Может.

12. Какой алгоритм является алгоритмом циклической структуры?

Циклический алгоритм — это вид алгоритма, в процессе выполнения которого одно или несколько действий нужно повторить.

13. Типы циклов в языке Python.

В Python есть 2 типа циклов: - цикл while, - цикл for.

14. Назовите назначение и способы применения функции range.

Функция range генерирует серию целых чисел, от значения start до stop, указанного пользователем. Мы можем использовать его для цикла for и обходить весь диапазон как список.

15. Как с помощью функции range организовать перебор значений от 15 до 0 с шагом 2?

```
range(15, 0, 2)
```

16. Могут ли быть циклы вложенными?

Могут.

17. Как образуется бесконечный цикл и как выйти из него?

Бесконечный цикл в программировании — цикл, написанный таким образом, что условие выхода из него никогда не выполняется.

18. Для чего нужен оператор break?

Используется для выхода из цикла.

19. Где употребляется оператор continue и для чего он используется?

Оператор continue используется только в циклах. В операторах for , while , do while , оператор continue выполняет пропуск оставшейся части кода тела цикла и переходит к следующей итерации цикла.

20. Для чего нужны стандартные потоки stdout и stderr?

Ввод и вывод распределяется между тремя стандартными потоками: stdin — стандартный ввод (клавиатура), stdout — стандартный вывод (экран), stderr — стандартная ошибка (вывод ошибок на экран)

21. Как в Python организовать вывод в стандартный поток stderr?

Указать в `print(..., file=sys.stderr)`.

22. Каково назначение функции `exit`?

Функция `exit()` модуля `sys` - выход из Python.

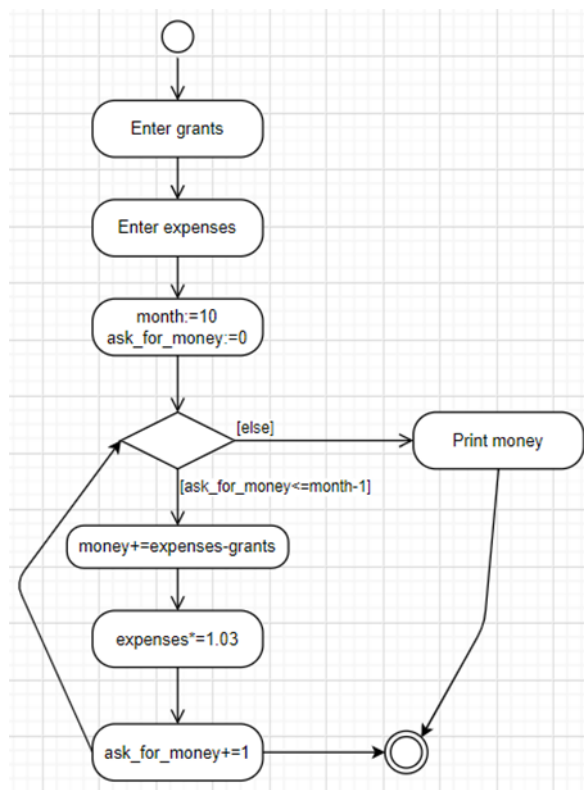


Рисунок 3.9 UML-диаграмма

