

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №2.2 (5)**  
**дисциплины**  
**«Программирование на Python»**

Выполнил:  
Ибрагимов Муса Айнудинович  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника»,  
направленность (профиль)  
«Программное обеспечение средств  
вычислительной техники и  
автоматизированных систем», очная  
форма обучения

---

(подпись)

Руководитель:  
Воронкин Роман Александрович

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

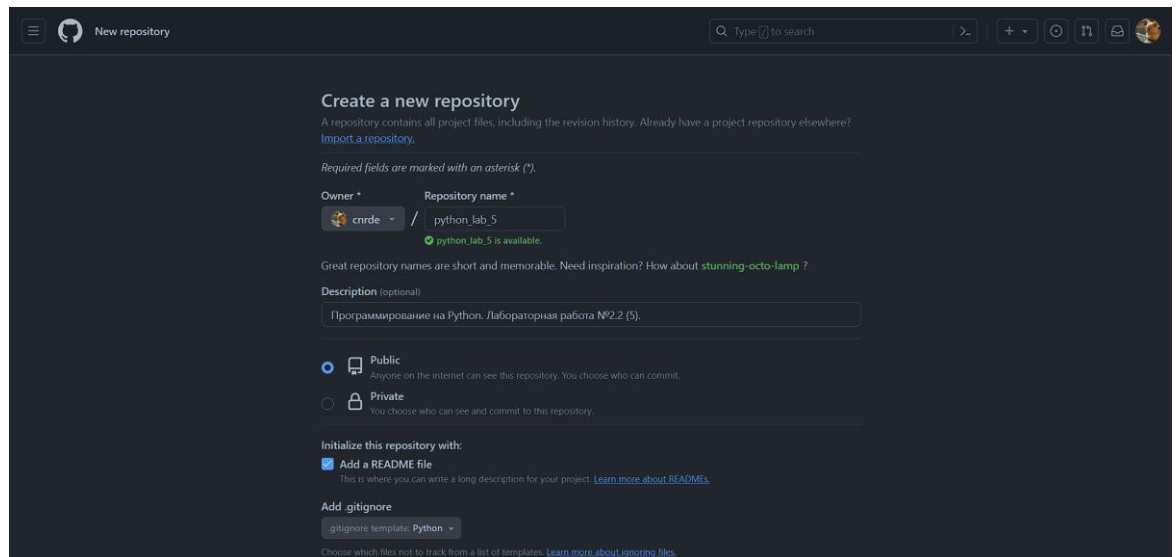
Ставрополь, 2023 г.

Тема: Условные операторы и циклы в языке Python.

Цель: Приобретение навыков программирования разветвляющихся алгоритмов и алгоритмов циклической структуры. Освоить операторы языка Python версии 3.x if, while, for, break и continue, позволяющих реализовывать разветвляющиеся алгоритмы и алгоритмы циклической структуры.

### Практическая часть:

1. Изучил теоретический материал работы.
2. Создал общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.



3. Проработал примеры из лабораторной работы.

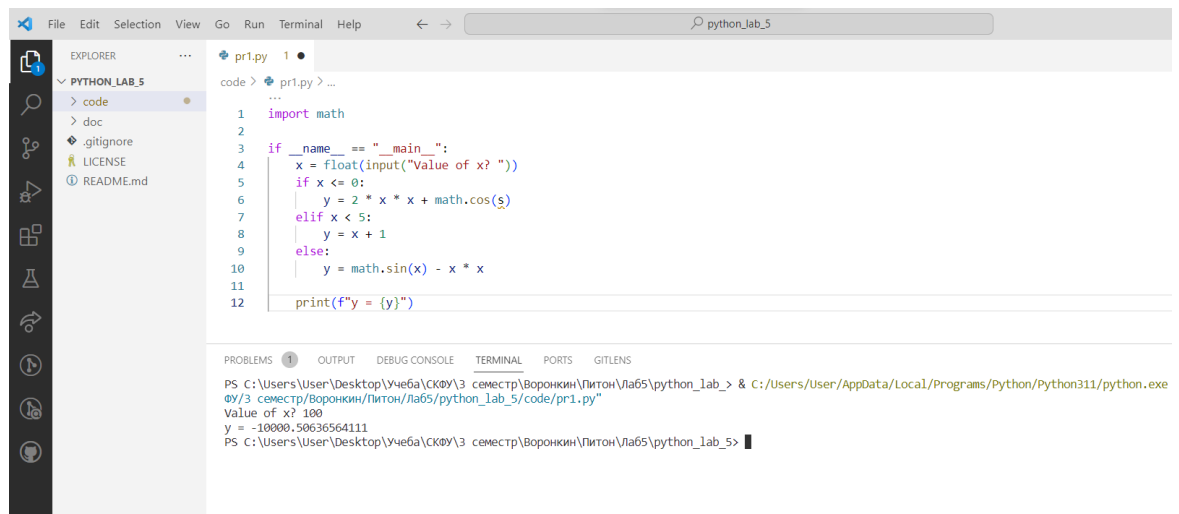


Рисунок 1. Первый пример.

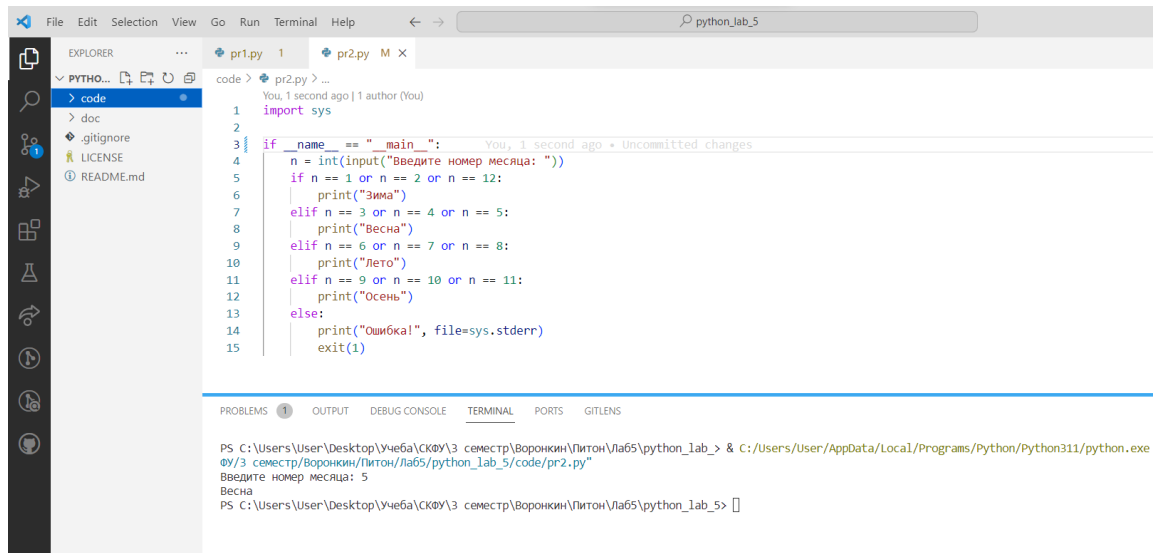


Рисунок 2. Второй пример.

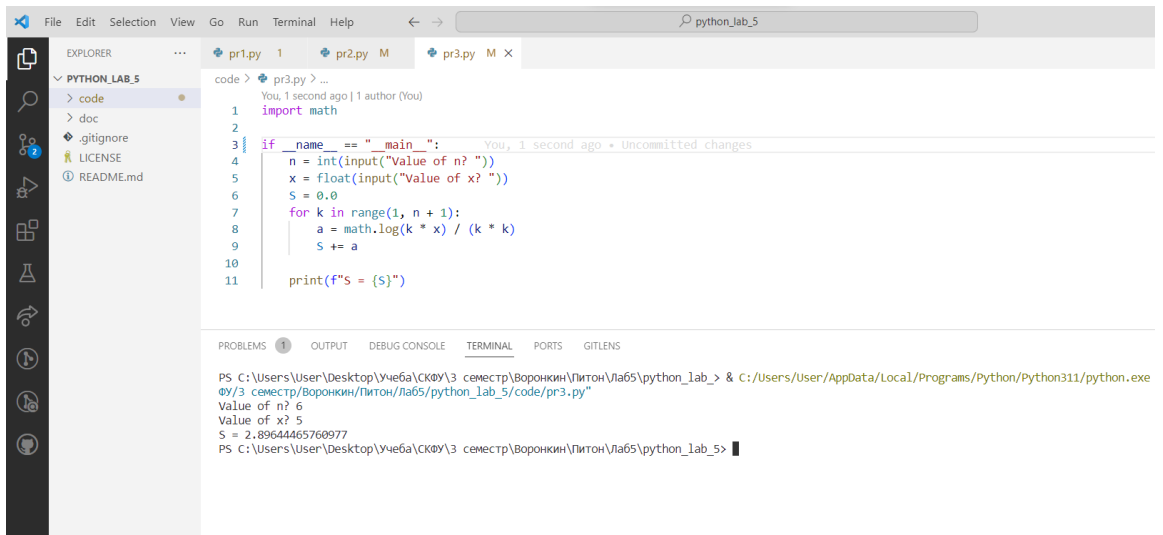


Рисунок 3. Третий пример.

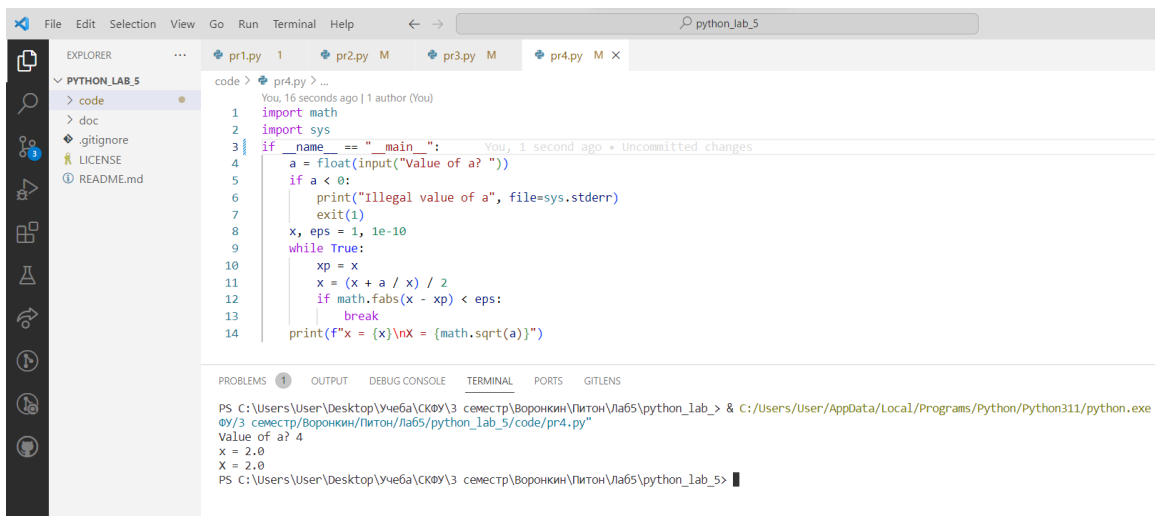


Рисунок 4. Четвертый пример.

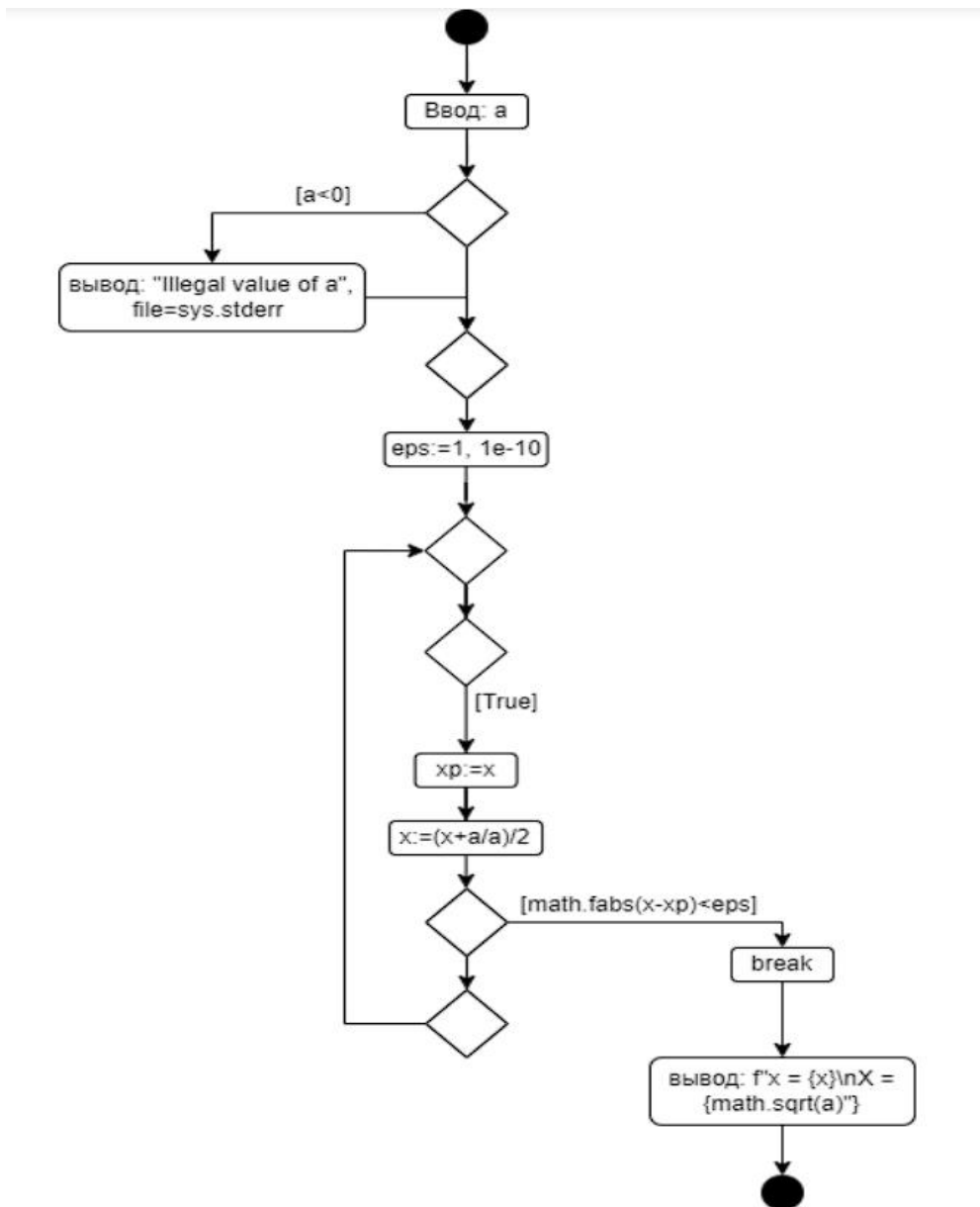


Рисунок 5. UML-диаграмма для программы четвертого примера.

```

code > pr5.py > ...
You, 1 second ago | 1 author (You)
1 import math
2 import sys
3 EULER = 0.5772156649015328606
4 EPS = 1e-10
5 if __name__ == "__main__":
6     x = float(input("Value of x? "))
7     if x == 0:
8         print("Illegal value of x", file=sys.stderr)
9         exit(1)
10    a = x
11    S, k = a, 1
12    while math.fabs(a) > EPS:
13        a = x * k / (k + 1) ** 2
14        S += a
15        k += 1
16    print(f"Ei({x}) = {EULER + math.log(math.fabs(x)) + S}")
  
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

```

PS C:\Users\User\Desktop\Учеба\СкФ\3 семестр\Воронкин\Питон\лаб5\python_lab_5> & C:\Users\User\AppData\Local\Programs\Python\Python311\python.exe
ФУ/3 семестр\Воронкин\Питон\лаб5\python_lab_5/code/pr5.py
Value of x? 19
Ei(19.0) = 9950907.251046844
PS C:\Users\User\Desktop\Учеба\СкФ\3 семестр\Воронкин\Питон\лаб5\python_lab_5>
  
```

Рисунок 6. Пятый пример.

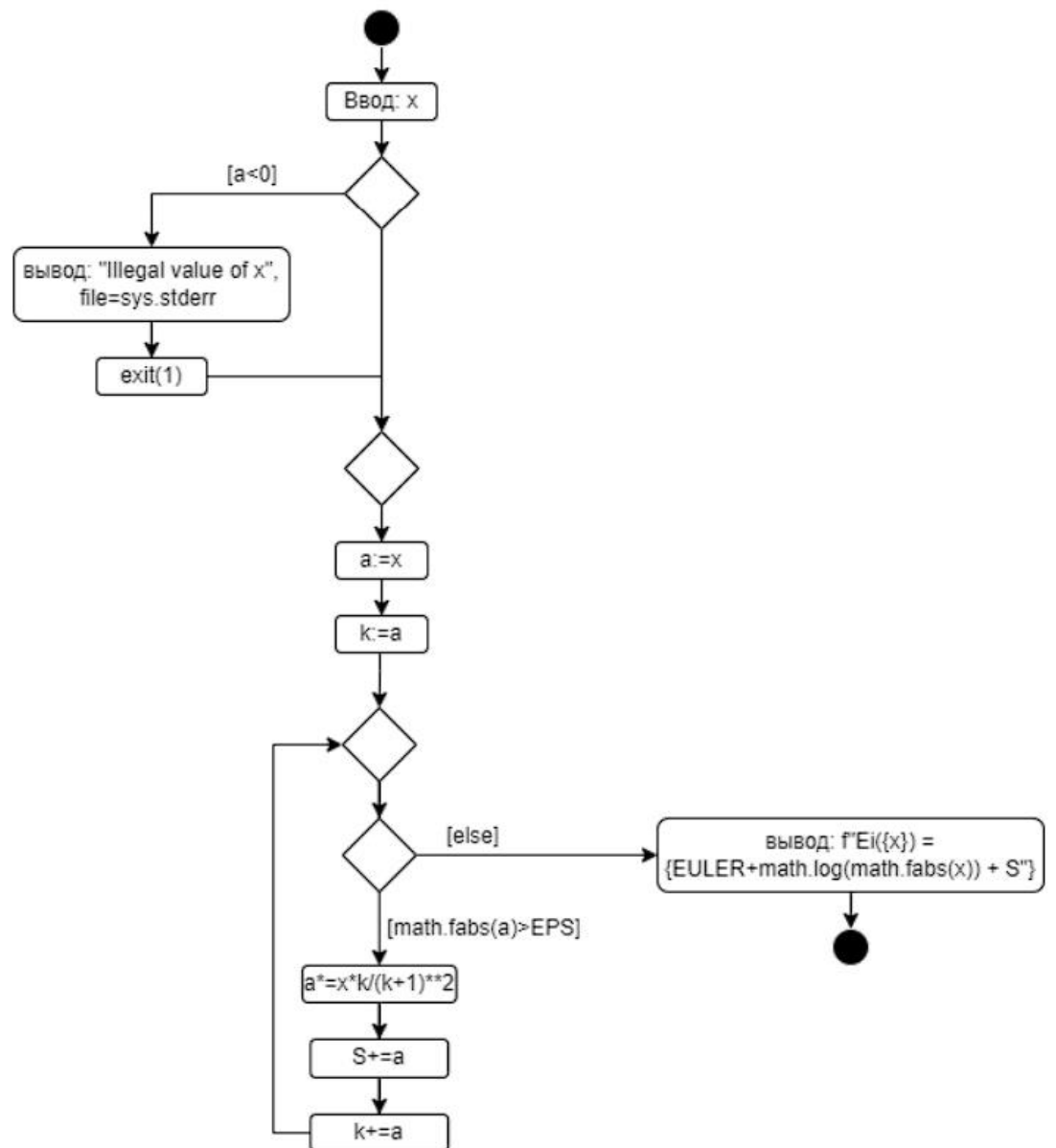
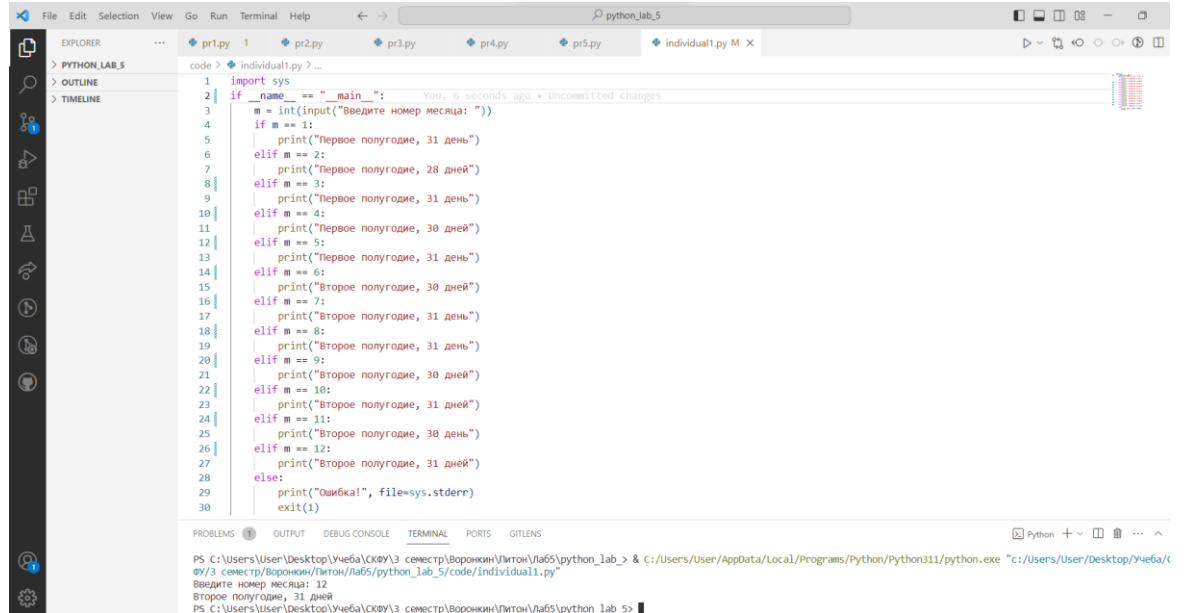


Рисунок 7. UML-диаграмма к программе пятого примера.

#### 4. Индивидуальное задание, вариант 8, задание 1:

Дано число  $m(1 \leq m \leq 12)$ . Определить полугодие, на которое приходится месяц с номером и количество дней в том месяце (год не високосный).



```
1 import sys
2 if __name__ == "__main__":
3     m = int(input("Введите номер месяца: "))
4     if m == 1:
5         print("Первое полугодие, 31 день")
6     elif m == 2:
7         print("Первое полугодие, 28 дней")
8     elif m == 3:
9         print("Первое полугодие, 31 день")
10    elif m == 4:
11        print("Первое полугодие, 30 дней")
12    elif m == 5:
13        print("Первое полугодие, 31 день")
14    elif m == 6:
15        print("Второе полугодие, 30 дней")
16    elif m == 7:
17        print("Второе полугодие, 31 день")
18    elif m == 8:
19        print("Второе полугодие, 31 день")
20    elif m == 9:
21        print("Второе полугодие, 30 дней")
22    elif m == 10:
23        print("Второе полугодие, 31 день")
24    elif m == 11:
25        print("Второе полугодие, 30 дней")
26    elif m == 12:
27        print("Второе полугодие, 31 день")
28    else:
29        print("Ошибка!", file=sys.stderr)
30    exit(1)
```

Terminal output:

```
PS C:\Users\User\Desktop\Учеба\СКД\У3 семестр\Воронкин\Литон\lab5\python_lab_5> & C:/Users/User/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/User/Desktop/Учеба/У3 семестр/Воронкин/Литон/lab5/python_lab_5/code/individual1.py"
Введите номер месяца: 12
Второе полугодие, 31 день
PS C:\Users\User\Desktop\Учеба\СКД\У3 семестр\Воронкин\Литон\lab5\python_lab_5>
```

Рисунок 8. Первое индивидуальное задание.

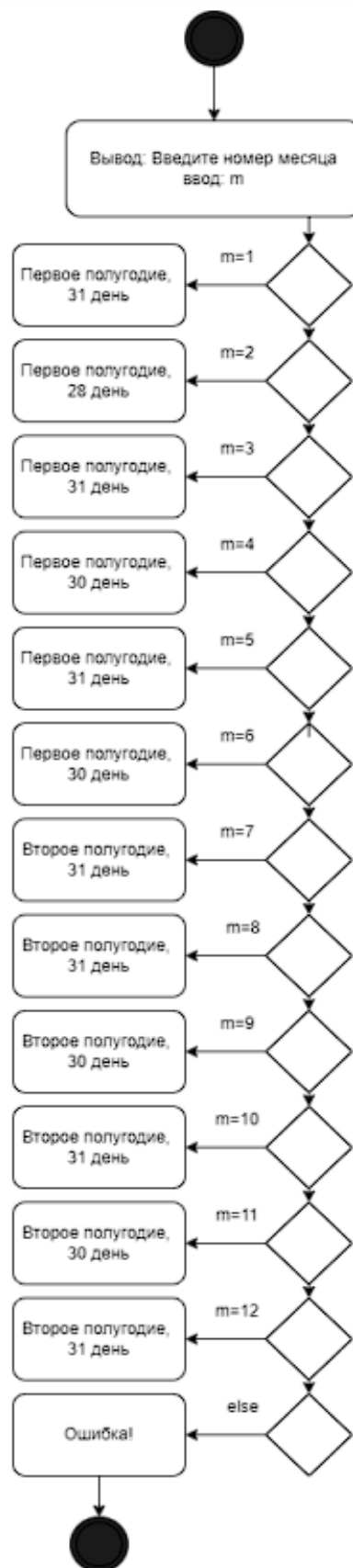


Рисунок 9. UML-диаграмма к первому индивидуальному заданию.

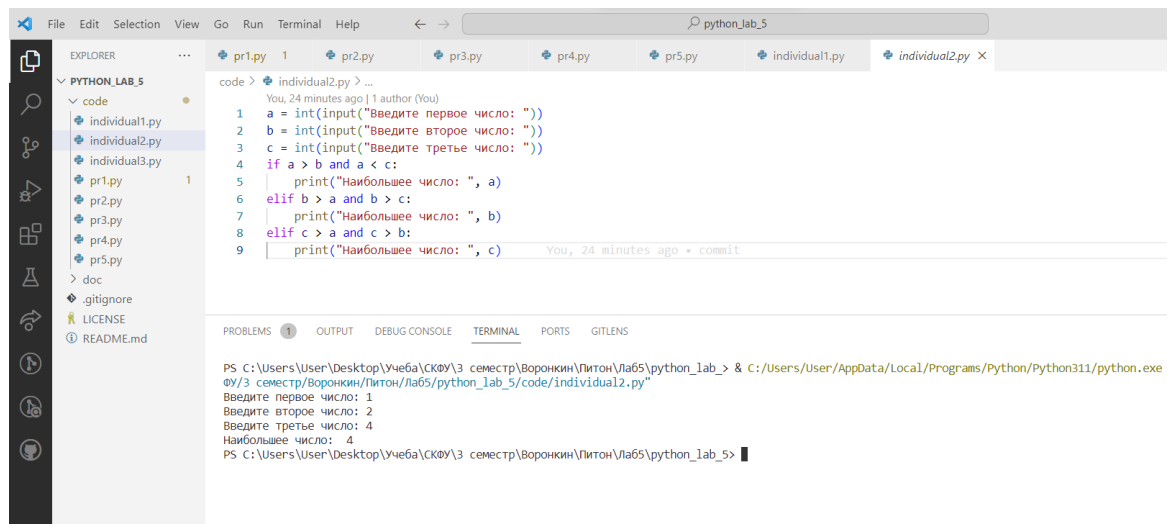


Рисунок 10. Второе индивидуальное задание.

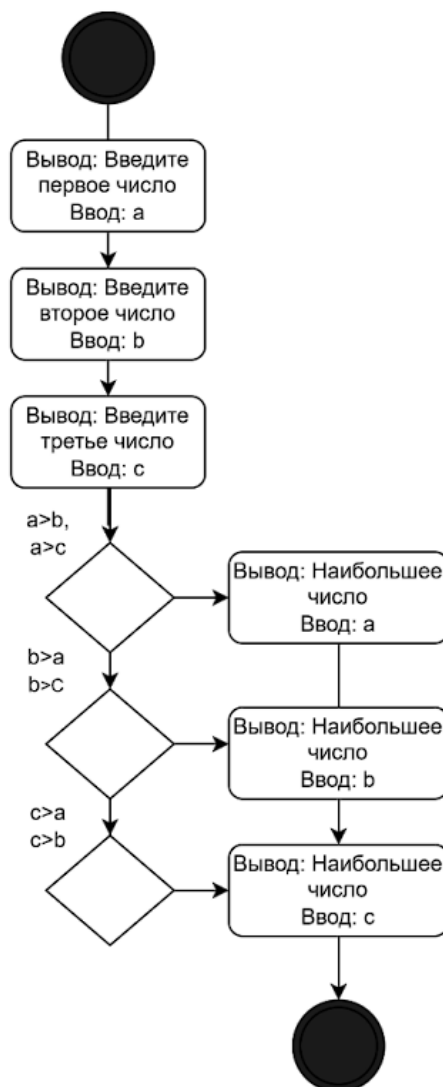


Рисунок 11. UML-диаграмма ко второму индивидуальному заданию.



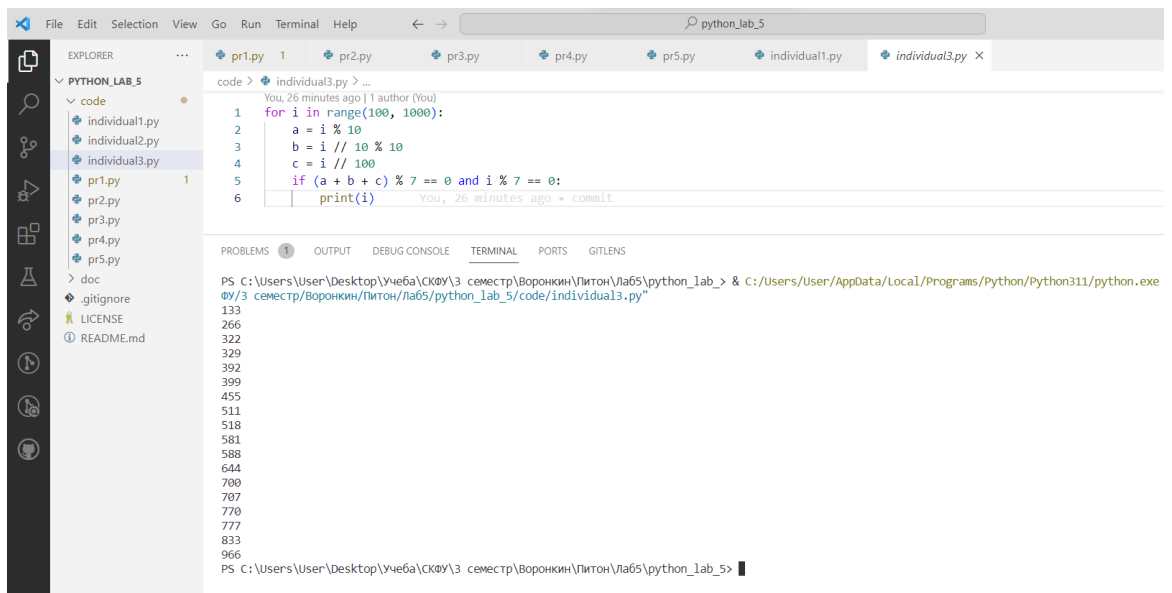


Рисунок 12. Третье индивидуальное заданиею.

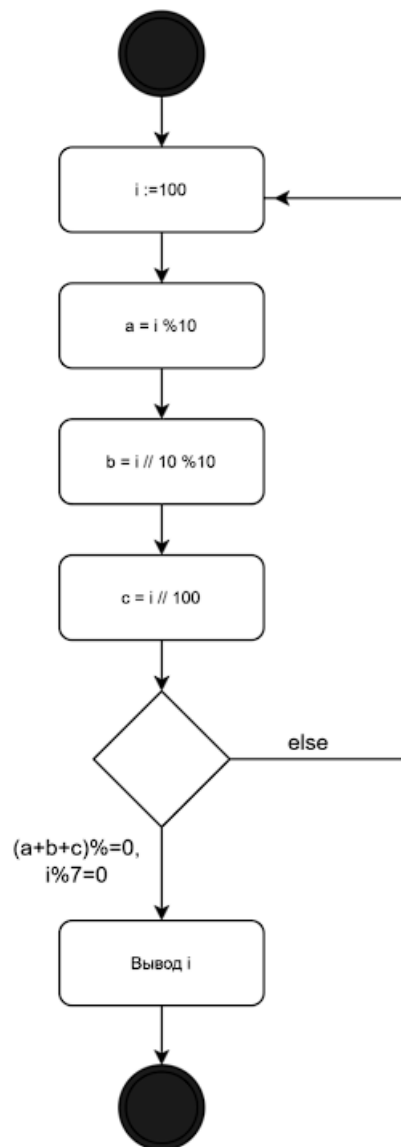
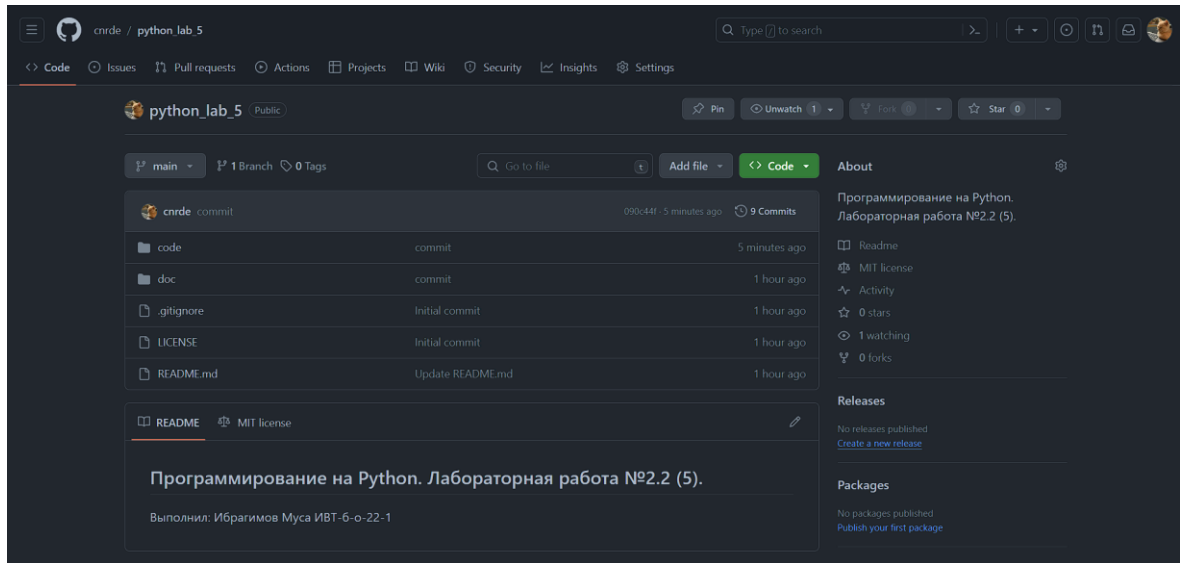


Рисунок 13. UML-диаграмма к третьему индивидуальному заданию.

5. Зафиксировал сделанные изменения репозитория и отправил их на сервер GitHub.



### Ответы на контрольные 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.

**Вывод:** Приобрел навыков программирования разветвляющихся алгоритмов и алгоритмов циклической структуры. Освоил операторы языка Python версии 3.x if, while, for, break и continue, позволяющих реализовывать разветвляющиеся алгоритмы и алгоритмы циклической структуры.