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
дисциплины
«Основы программной инженерии»

Выполнил:
Разворотников Денис Сергеевич
2 курс, группа ПИЖ-б-о-21-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

_____
(подпись)

Проверил:
_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь 2022

Цель: работы: 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.

1. Был создан общедоступного репозиторий в GitHub в котором были добавлены gitignore, правила для работы с IDE PyCharm с ЯП Python и лицензия MIT, репозиторий был клонировал на локальный сервер и организован в соответствие с моделью ветвления git-flow.

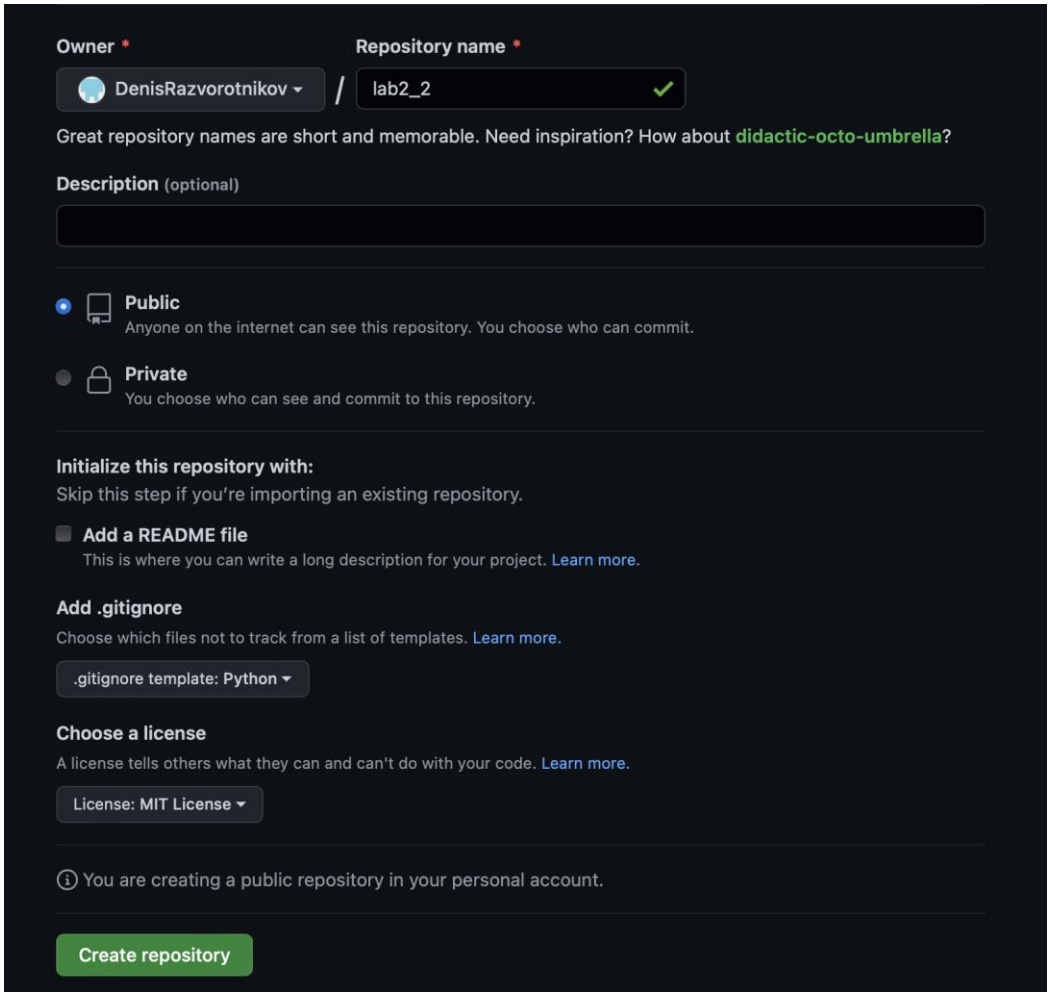

Рисунок 1.1 – Создание общедоступного репозитория

```
denisrazvorotnikov@Air-Denis desktop % git clone https://github.com/DenisRazvoro
tnikov/lab2_2.git
Cloning into 'lab2_2'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
denisrazvorotnikov@Air-Denis desktop %
```

Рисунок 1.2 – Клонирование созданного репозитория

```
C:\GIt_rep>git flow init
No branches exist yet. Base branches must be created now.
Branch name for production releases: [master]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/GIt_rep/.git/hooks]

C:\GIt_rep>_
```

Рисунок 1.3 – Организование репозитория по модели ветвления git flow

2. Была создана папка (common_files), содержащая примеры из лабораторной работы

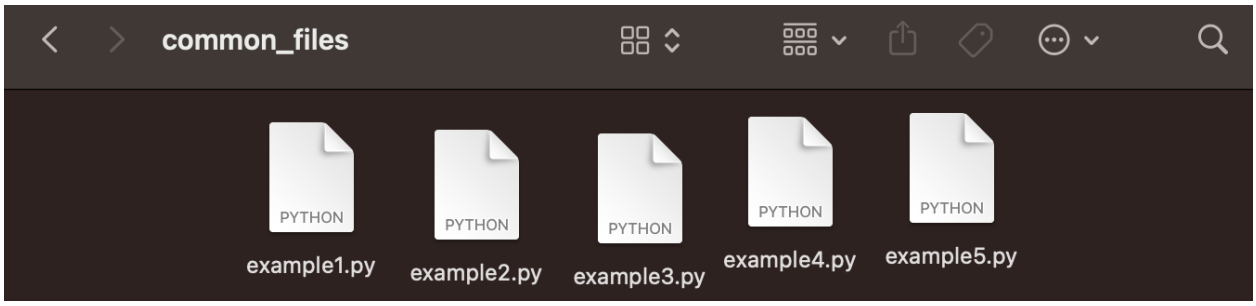

Рисунок 2.1 – Содержание папки common_files

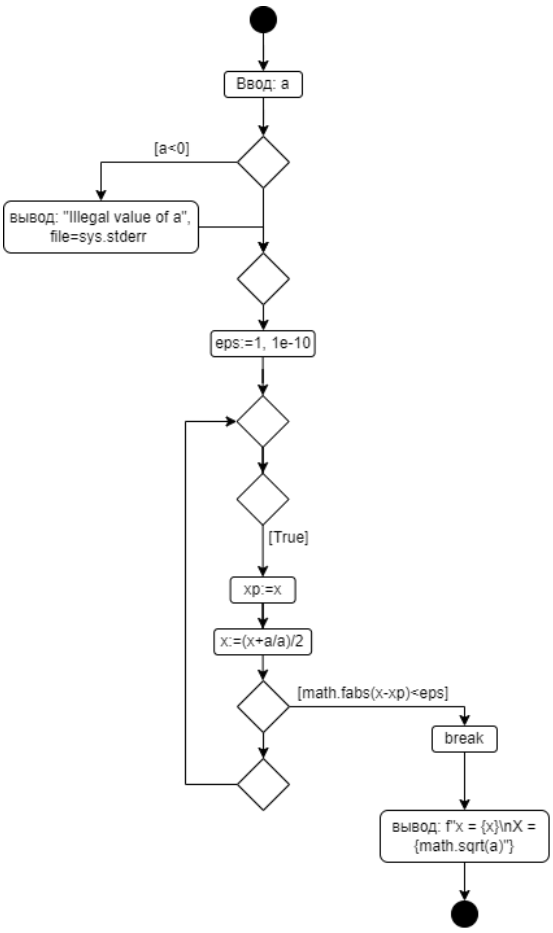

Рисунок 2.2 – UML-диаграмма программы для 4 примеры

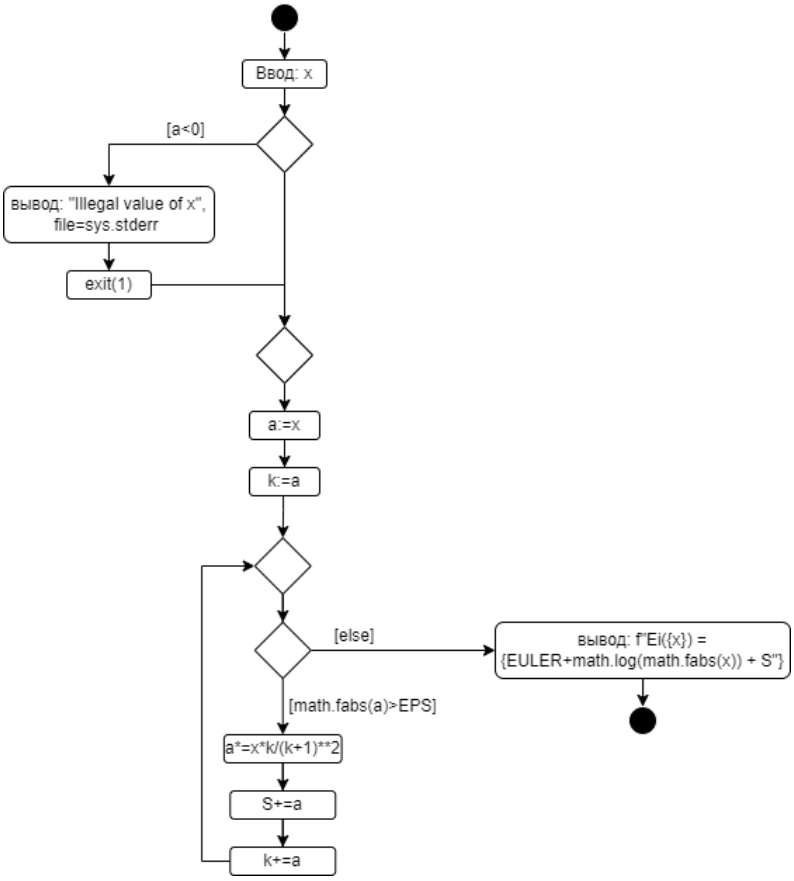

Рисунок 2.3 – UML-диаграмма для программы 5 примера

Индивидуальное задание

3 Было сделано индивидуальное задание Была построена UML диаграмма.

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

Код программы:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import sys

if __name__ == '__main__':
    a = int(input('Enter nomer month (the rage is from 1 to 12): '))
    if a==1:
        print('Это первое полугодие. Месяц Январь - 31 день')
    elif a==2:
        print('Это первое полугодие. Месяц Февраль - 29 день')
    elif a == 3:
        print('Это первое полугодие. Месяц Март - 31 день')
    elif a == 4:
```

```
        print('Это первое полугодие. Месяц Апрель - 30 день')
    elif a == 5:
        print('Это первое полугодие. Месяц Май - 31 день')
    elif a == 6:
         print('Это первое полугодие. Месяц Июнь - 30 день')
    elif a == 7:
        print('Это второе полугодие. Месяц Июль - 31 день')
    elif a == 8:
        print('Это второе полугодие. Месяц Август - 31 день')
    elif a == 9:
        print('Это второе полугодие. Месяц Сентябрь - 30 день')
    elif a == 10:
        print('Это второе полугодие. Месяц Октябрь - 31 день')
    elif a == 11:
        print('Это второе полугодие. Месяц Ноябрь - 30 день')
    elif a == 12:
        print('Это второе полугодие. Месяц Декабрь - 31 день')
    else:
        print(
            'Error',
            file=sys.stderr
        )
```

```
Enter nomer month (the rage is from 1 to 12): 6
Это первое полугодие. Месяц Июнь - 30 день

Process finished with exit code 0
```

```
Enter nomer month (the rage is from 1 to 12): 13
Error

Process finished with exit code 0
```

Рисунок 3.1 – Результат работы программы

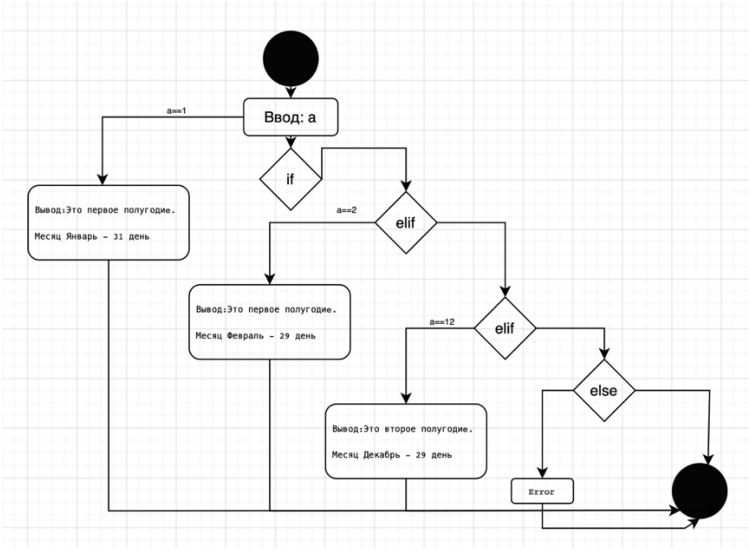

Рисунок 3.2 – UML диаграмма программы

Задание 2. Определить принадлежит ли точка A (a, b) кольцу определяемому окружностями x^2 + y^2 = 1 и x^2 + y^2 = 0.25.

Код программы:

```python
#!/usr/bin/env python3
# -- coding: utf-8 --

import math

if __name__ == '__main__':
    x = float(input())
    y = float(input())
    if 1 >= math.sqrt(x ** 2 + y ** 2) >= 0.25:
        print("Входит в кольцо")
    else:
        print("Не входит в кольцо")
```

```
/Users/denisrazvorotnikov/Desktop/lab2_2
10
100
Не входит в кольцо

Process finished with exit code 0
```

```
/Users/denisrazvorotnikov/Desktop/lab2_2
0
0.25
Входит в кольцо

Process finished with exit code 0
```

Рисунок 3.3 – Результат работы программы

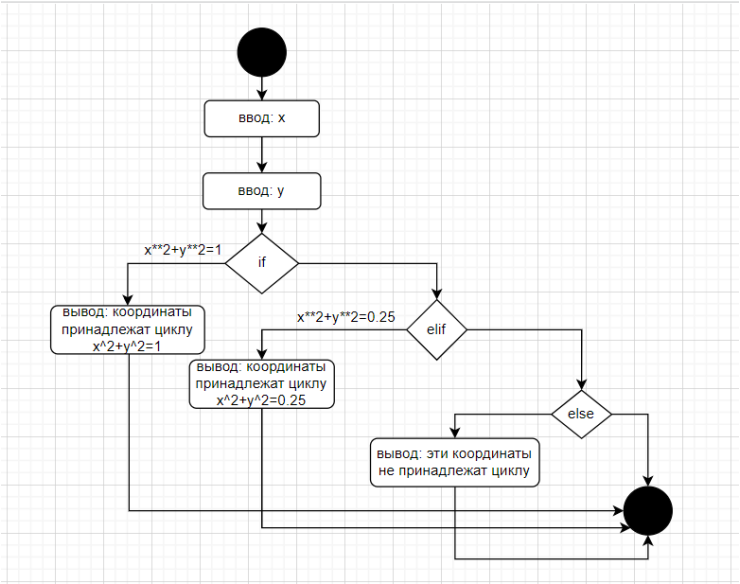

Рисунок 3.4 – UML диаграмма программы

Задание 3. Сумма цифр трехзначного числа кратна 7. Само число также делится на 7. Найти все такие числа.

Код программы:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

for i in range(105, 999, 7):
    h = i//100
    d = (i%100)//10
    u = i%10
    if h + d + u == 7:
        print(i)
```

```
133
322
511
700
```

Рисунок 3.5 – Результат работы программы

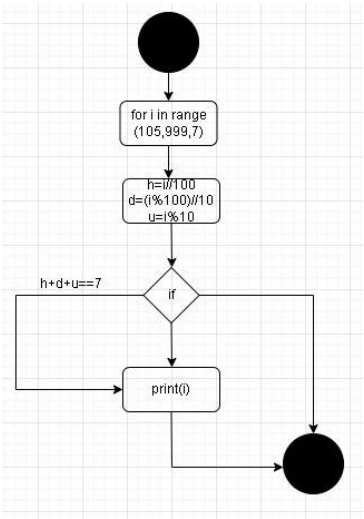

Рисунок 3.6 – UML диаграмма программы

Задание повышенной сложности

6. Функция Бесселя первого рода $J_n(x)$, значение $n = 0, 1, 2, \ldots$ также должно вводиться с клавиатуры

$$J_n(x) = \left(\frac{x}{2}\right)^n \sum_{k=0}^{\infty} \frac{(-x^2/4)^k}{k!(k+n)!}. \quad (3.6)$$

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import math

EPS = 1e-10
if __name__ == '__main__':
    x = float(input("Input x: "))
    n = float(input("Input n: "))
    a = x
    s = 0
    k = 0
    while math.fabs(a) > EPS:
        a = (((-x ** 2) / 4) * (k + 1)) / ((k + 1 ) * (k +n))
        s = s + a
        k = k + 1
    print(f"J({x}) = {((x / 2)**2) * s}")
```

```
C:\Users\UESR\Desktop\2.1\proj\venv\Scripts\python.exe
Input x: 2
Input n: 1
J(2.0) = -1.2642411176556763

Process finished with exit code 0
```

Рисунок 3.7 – Результат работы программы

Вывод: в результате лабораторной работы были приобретены навыки программирования разветвляющихся алгоритмов и алгоритмов циклической структуры. Были освоены операторы языка Python версии 3.x if, while, for, break и continue, позволяющих реализовывать разветвляющиеся алгоритмы и алгоритмы циклической структуры.

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, объединенных 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.