

**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

Богдашов Артём Владимирович

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.

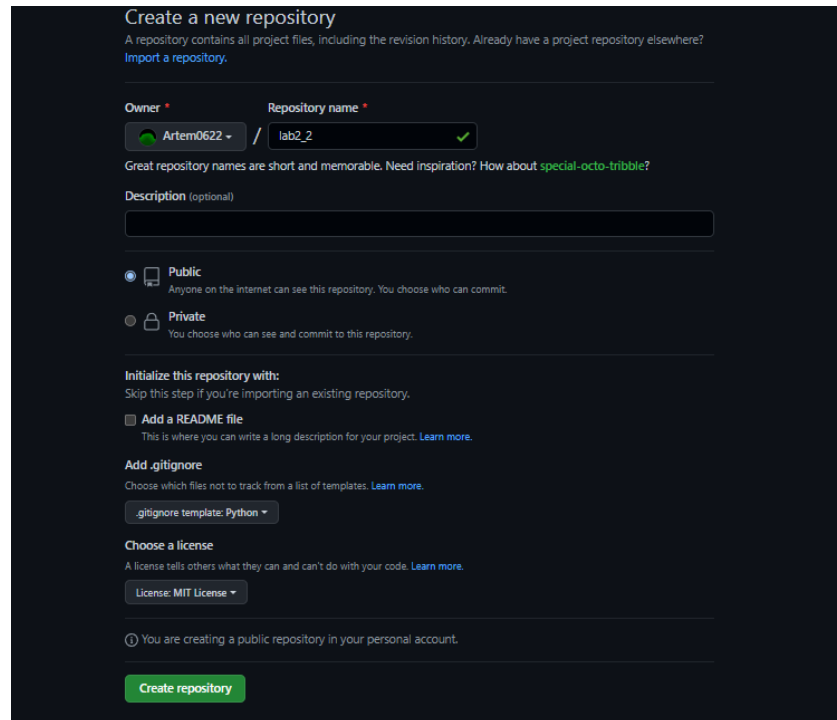


Рисунок 1.1 Создание репозитория

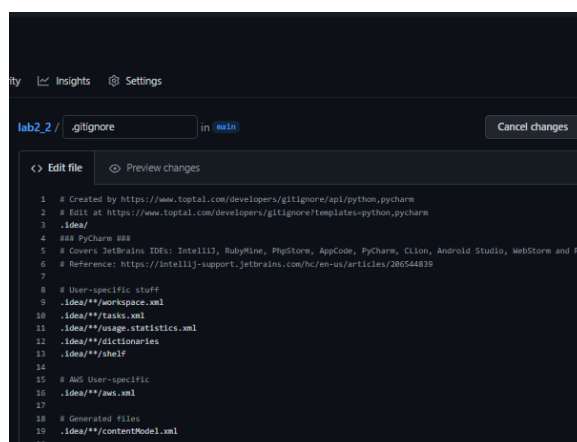


Рисунок 1.2 Добавление правил в .gitignore

```
C:\Games\Программы\СУ\1.2К\Программирование\2.2>git clone https://github.com/Artem0622/lab2_2.git
Cloning into 'lab2_2'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 7 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (7/7), 4.27 KiB | 4.27 MiB/s, done.
Resolving deltas: 100% (1/1), done.
```

Рисунок 1.3 Клонирование репозитория

```
C:\Games\Программы\СУ\1.2К\Программирование\2.2\lab2_2>git flow init
Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
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:/Games/Программы/СУ/1.2К/Программирование/2.2/lab2_2/.git/hooks]
```

Рисунок 1.4 Организация репозитория согласно модели ветвления git-flow

Пример 1. Составить UML-диаграмму деятельности и программу с использованием конструкции ветвления и вычислить значение функции

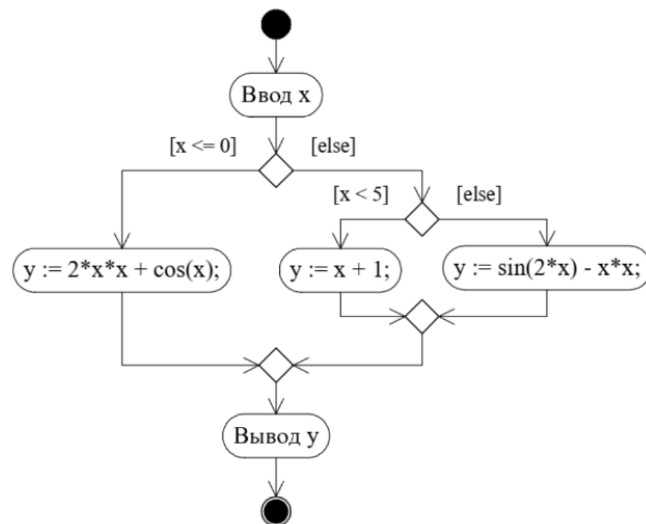


Рисунок 2.1 UML-диаграмма программы 1 примера

```
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Games\Программы\CV\1.2KC\Программирование\Коды\2.2\program\primer1.py
Value of x? 0
y = 1.0
>>>
```

Рисунок 2.2 Результат выполнения программы

Пример 2. Составить UML-диаграмму деятельности и программу для решения задачи: с клавиатуры вводится номер месяца от 1 до 12, необходимо для этого номера месяца вывести наименование времени года.

```
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Games\Программы\CV\1.2KC\Программирование\Коды\2.2\program\primer1.py
Value of x? 0
y = 1.0
>>>
= RESTART: C:\Games\Программы\CV\1.2KC\Программирование\Коды\2.2\program\primer2.py
Введите номер месяца: 5
Весна
>>>
```

Рисунок 2.3 Результат выполнения программы

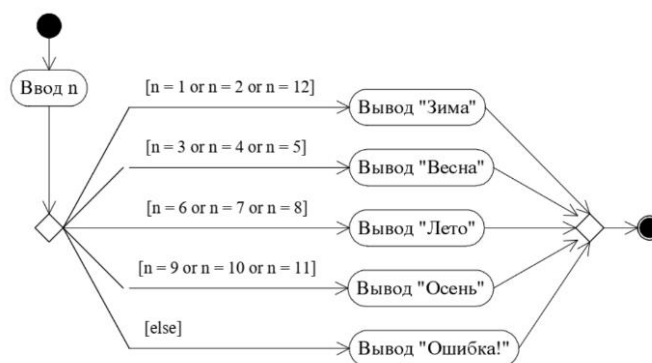


Рисунок 2.4 UML-диаграмма программы 2 примера

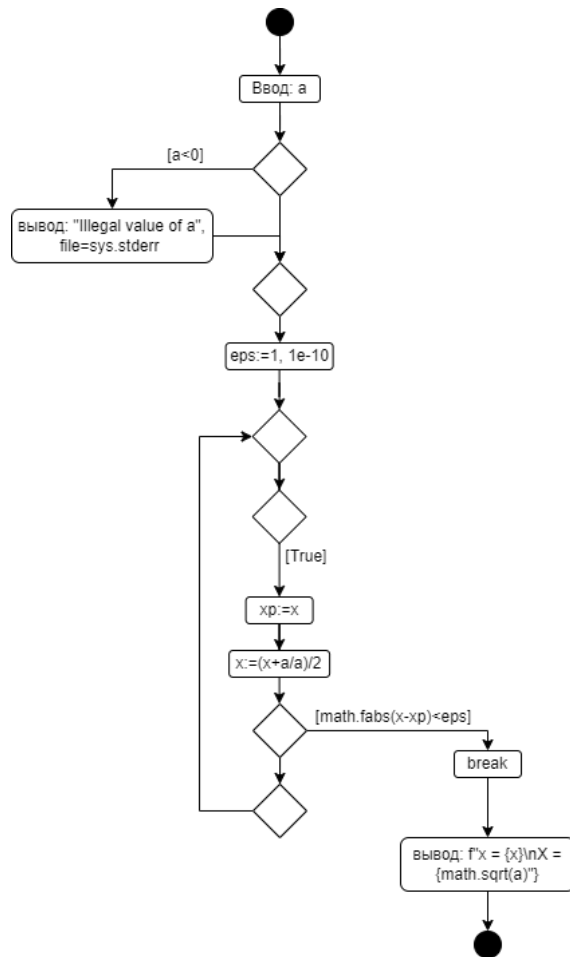


Рисунок 2.5 UML-диаграмма программы 4 примера

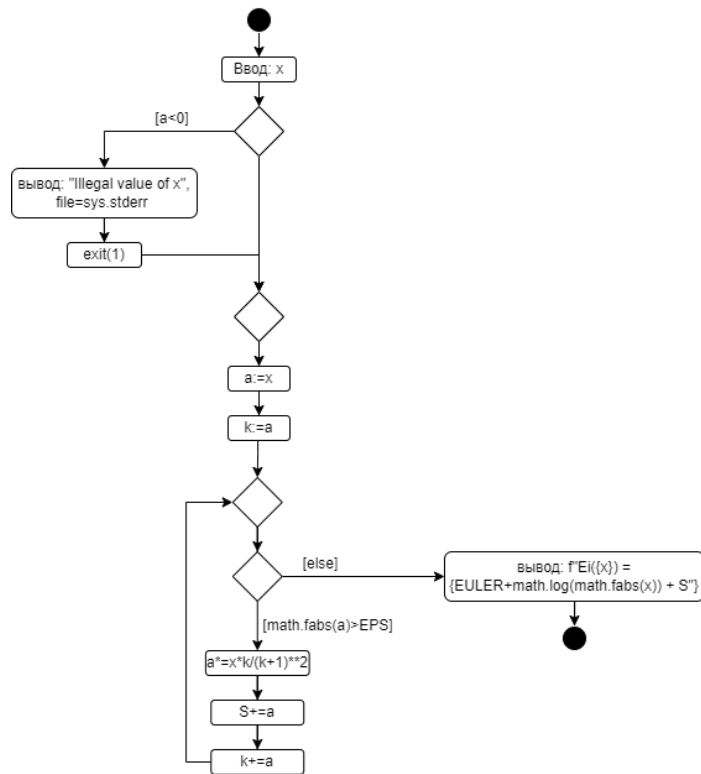


Рисунок 2.6 UML-диаграмма программы 5 примера

Индивидуальное задание

Вариант 3

Задание 1. Дано число m ($1 \leq m \leq 7$). Вывести на экран название дня недели, который соответствует этому номеру.

```
= RESTART: C:\Games\Программы\СУ\1.2КС\Программирование\Коды\2.2\Индивидуальное  
задание\1.py  
Введите число:1  
Понедельник
```

Рисунок 3.1 Результат выполнения программы.

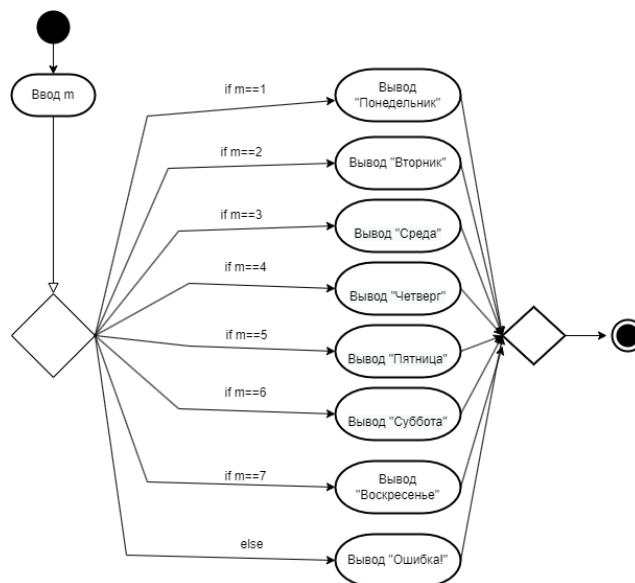


Рисунок 3.2 UML – диаграмма к программе инд. задания 1

Задание 2. Из трёх действительных чисел a, b и c выбрать те, модули которых не меньше 4.

```
>>> = RESTART: C:/Games/Программы/СУ/1.2КС/Программирование/Коды/2.2/Индивидуальное  
задание/2.py  
Введите действительное число a:6  
Введите действительное число b:5  
Введите действительное число c:2  
6.0  
5.0
```

Рисунок 3.3 Результат выполнения программы

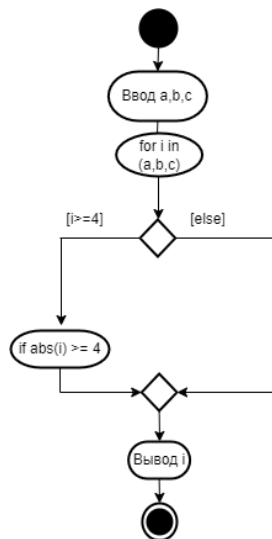


Рисунок 3.4 UML – диаграмма к программе инд. задания 2

Задание 3. Начав тренировки, спортсмен пробежал 10 км. Каждый следующий день он увеличивал дневную норму на 10% от нормы предыдущего дня. Какой суммарный путь пробежит спортсмен за 7 дней?

```

>>> = RESTART: C:/Games/Программы/CV/1.2KC/Программирование/Коды/2.2/Индивидуальное
      задание/3.py
      После увеличения нормы на 10 % суммарно за 7 дней пробежит:
      94.87171
  
```

Рисунок 3.5 Результат выполнения программы

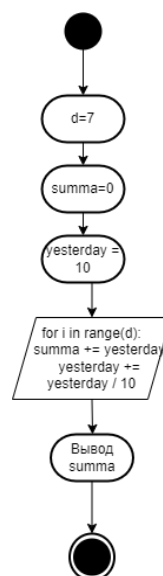


Рисунок 3.6 UML – диаграмма к программе инд. задания 3

Задание повышенной сложности

3. Интегральный гиперболический синус:

$$\text{Shi}(x) = \int_0^x \frac{\text{sh } t}{t} dt = \sum_{n=0}^{\infty} \frac{x^{2n+1}}{(2n+1)(2n+1)!}.$$

```
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Games\Программы\СУ\1.2К\Программирование\Коды\2.2\Усл задание\usl
.py
(1.710808329990548e+308, 981)
>>>
```

Рисунок 3.7 Результат выполнения программы

4. Сделал коммит, выполнил слияние с веткой main, и запустил изменения в уд. репозиторий.

```
C:\Games\Программы\СУ\1.2К\Программирование\2.2\lab2_2>git add .
C:\Games\Программы\СУ\1.2К\Программирование\2.2\lab2_2>git commit -m "save"
[develop b22ba76] save
18 files changed, 147 insertions(+), 21 deletions(-)
rewrite LICENSE (100%)
create mode 100644 "\320\221\320\273\320\276\320\272 \321\201\321\205\320\265\320\274\321\213\ind1.drawio"
create mode 100644 "\320\221\320\273\320\276\320\272 \321\201\321\205\320\265\320\274\321\213\ind1.png"
create mode 100644 "\320\221\320\273\320\276\320\272 \321\201\321\205\320\265\320\274\321\213\ind2.drawio"
create mode 100644 "\320\221\320\273\320\276\320\272 \321\201\321\205\320\265\320\274\321\213\ind2.drawio.png"
create mode 100644 "\320\221\320\273\320\276\320\272 \321\201\321\205\320\265\320\274\321\213\ind3.drawio"
create mode 100644 "\320\221\320\273\320\276\320\272 \321\201\321\205\320\265\320\274\321\213\ind3.png"
create mode 100644 "\320\221\320\273\320\276\320\272 \321\201\321\205\320\265\320\274\321\213\pr4.png"
create mode 100644 "\320\221\320\273\320\276\320\272 \321\201\321\205\320\265\320\274\321\213\pr5.png"
create mode 100644 "\320\230\320\275\320\264\320\270\320\262\320\270\320\264\321\203\320\260\320\273\321\214\320\275\32
0\276\320\265 \320\267\320\260\320\264\320\260\320\275\320\270\320\265\1.py"
create mode 100644 "\320\230\320\275\320\264\320\270\320\262\320\270\320\264\321\203\320\260\320\273\321\214\320\275\32
0\276\320\265 \320\267\320\260\320\264\320\260\320\275\320\270\320\265\2.py"
create mode 100644 "\320\230\320\275\320\264\320\270\320\262\320\270\320\264\321\203\320\260\320\273\321\214\320\275\32
0\276\320\265 \320\267\320\260\320\264\320\260\320\275\320\270\320\265\3.py"
create mode 100644 "\320\237\321\200\320\270\320\274\320\265\321\200\321\213\primer1.py"
create mode 100644 "\320\237\321\200\320\270\320\274\320\265\321\200\321\213\primer2.py"
create mode 100644 "\320\237\321\200\320\270\320\274\320\265\321\200\321\213\primer3.py"
create mode 100644 "\320\237\321\200\320\270\320\274\320\265\321\200\321\213\primer4.py"
create mode 100644 "\320\237\321\200\320\270\320\274\320\265\321\200\321\213\primer5.py"
create mode 100644 "\320\243\321\201\320\273 \320\267\320\260\320\264\320\260\320\275\320\270\320\265\usl.py"
```

Рисунок 4.1 коммит изменений и переход на ветку main

[illegible]

Рисунок 4.2 Слияние ветки main с develop

```
C:\Games\Программы\СУ\1.2КС\Программирование\2.2\lab2_2>git push
Enumerating objects: 30, done.
Counting objects: 100% (29/29), done.
Delta compression using up to 12 threads
Compressing objects: 100% (24/24), done.
Writing objects: 100% (26/26), 139.31 KiB | 15.48 MiB/s, done.
Total 26 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Artem0622/lab2_2.git
06af76a..0ecfd10 main -> main
```

Рисунок 4.3 Пуш изменений

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.

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.