

**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

Хашиев Адам Мухарбекович

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 соответствие с моделью ветвления git-flow.

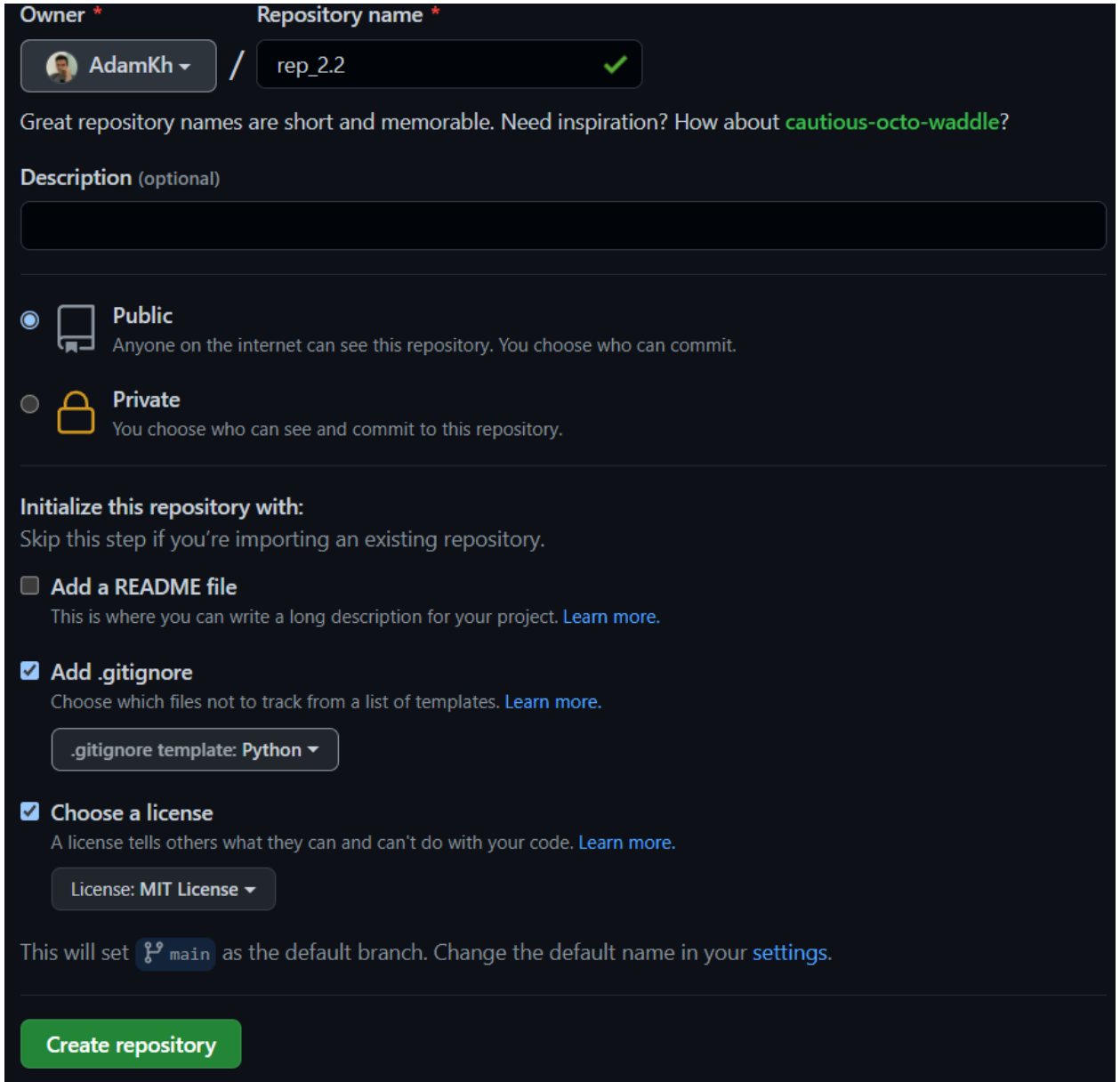
The image shows the GitHub 'Create repository' form. At the top, the 'Owner' is set to 'AdamKh' and the 'Repository name' is 'rep\_2.2', which is marked as valid with a green checkmark. Below this, there is a suggestion for repository names: 'Great repository names are short and memorable. Need inspiration? How about cautious-octo-waddle?'. The 'Description' field is empty. Under the 'Visibility' section, 'Public' is selected, with the note 'Anyone on the internet can see this repository. You choose who can commit.' The 'Private' option is also visible. In the 'Initialize this repository with:' section, three options are listed: 'Add a README file' (unchecked), 'Add .gitignore' (checked), and 'Choose a license' (checked). The '.gitignore' dropdown is set to 'Python' and the 'License' dropdown is set to 'MIT License'. At the bottom, it states 'This will set main as the default branch. Change the default name in your settings.' and a large green 'Create repository' button is at the very bottom.

Рисунок 1.1 Создание репозитория

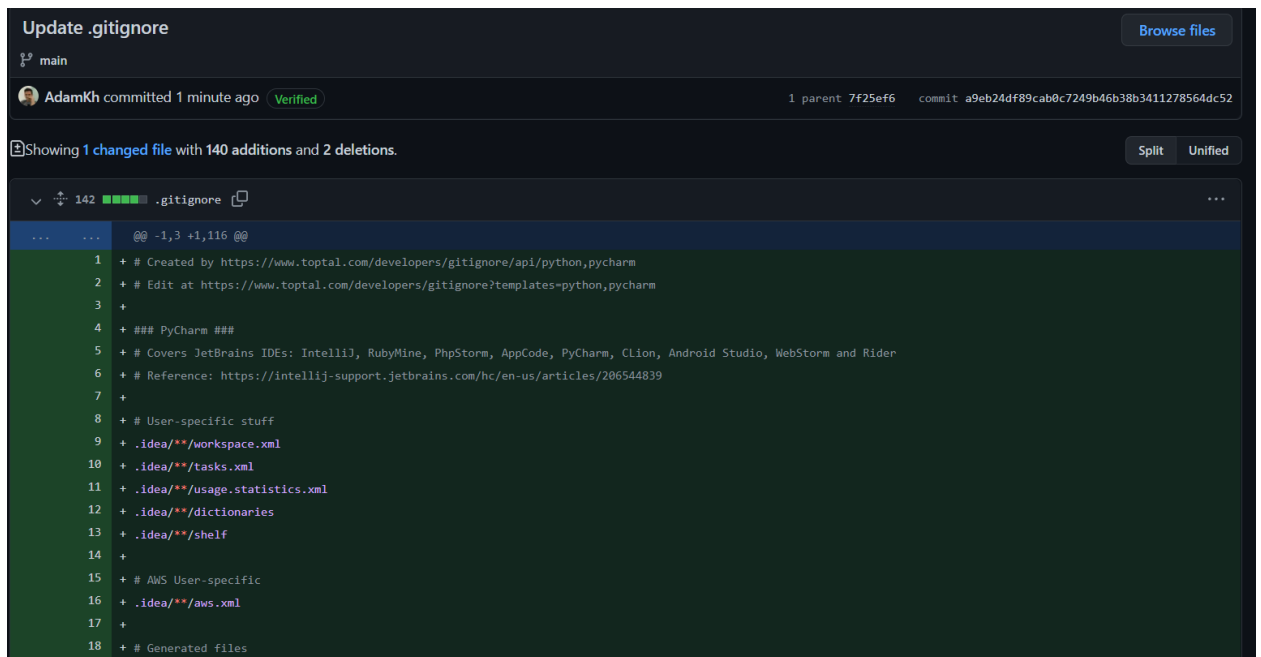


Рисунок 1.2 Добавление правил в .gitignore

```
C:\Users\adamk\OneDrive\Рабочий стол\ЛР 2.2>git clone https://github.com/AdamKh/rep_2.2.git
Cloning into 'rep_2.2'...
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), 4.25 KiB | 1.06 MiB/s, done.
Resolving deltas: 100% (1/1), done.

C:\Users\adamk\OneDrive\Рабочий стол\ЛР 2.2>git flow init
Initialized empty Git repository in C:/Users/adamk/OneDrive/Рабочий стол/ЛР 2.2/.git/
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:/Users/adamk/OneDrive/Рабочий стол/ЛР 2.2/.git/hooks]
```

Рисунок 1.3 Клонирование и организация репозитория согласно модели ветвления git-flow

2. Создал проект PyCharm в папке репозитория, проработал примеры ЛР.

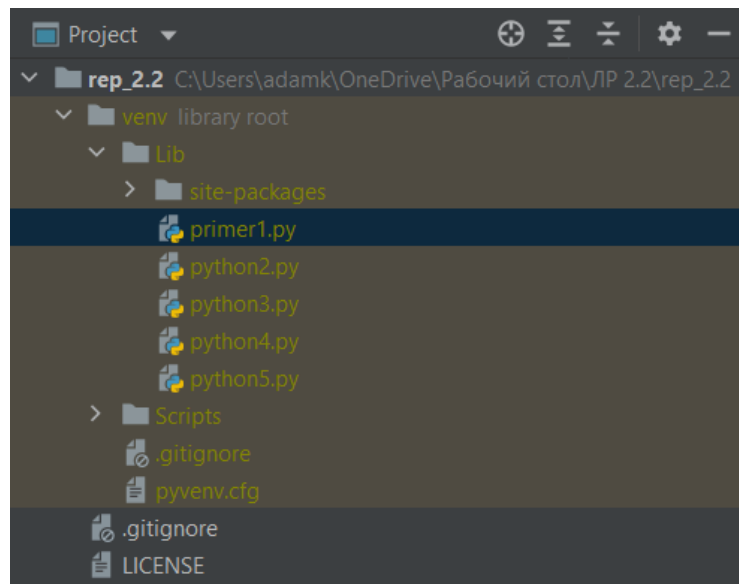


Рисунок 2.1 Примеры в проекте

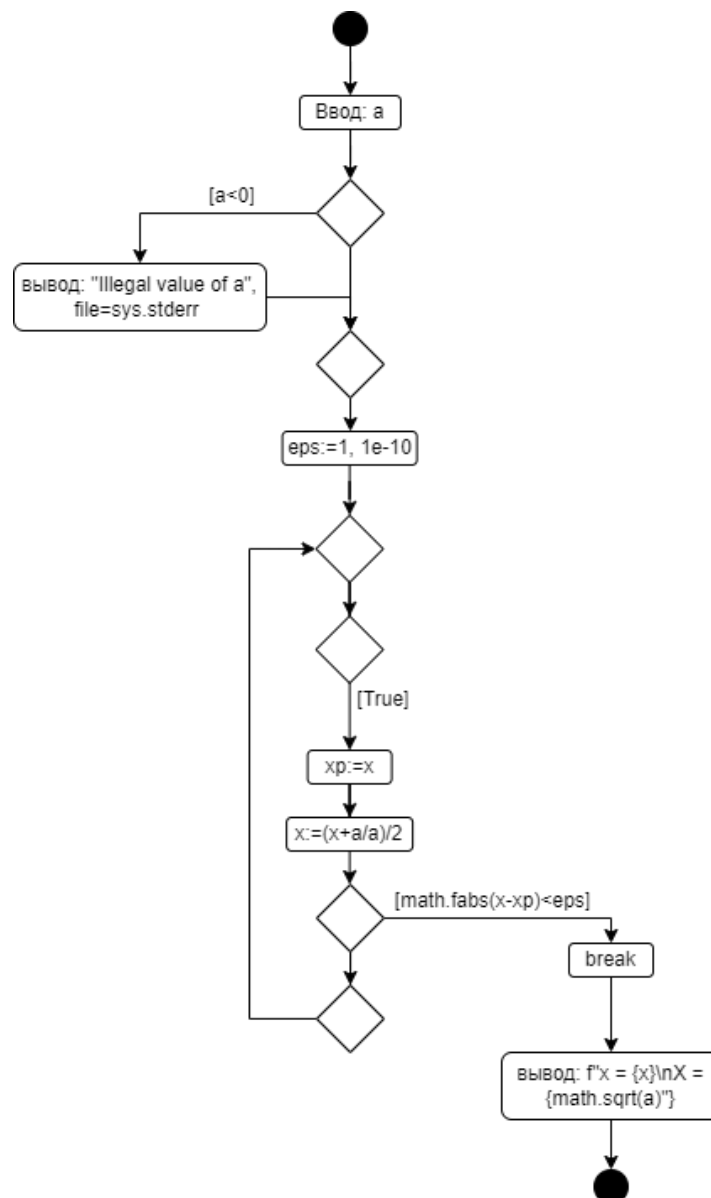


Рисунок 2.2 UML-диаграмма программы 4 примера

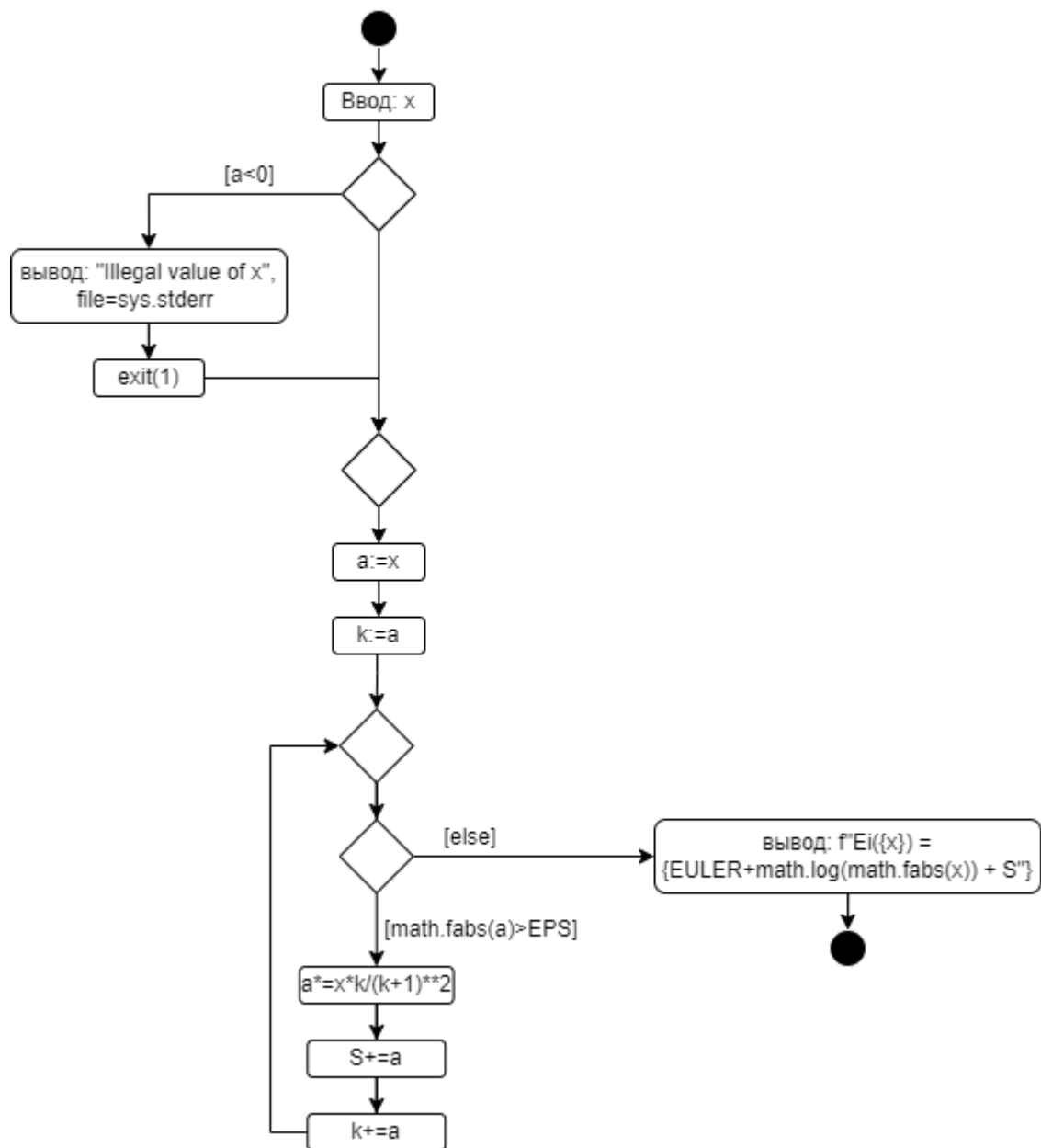


Рисунок 2.3 UML-диаграмма программы 5 примера

3. Выполнил индивидуальные задания и задание повышенной сложности согласно своему варианту. Построил UML диаграммы программ.

```

"""
Вводится число N<=20 экзаменов . Напечатать фразу "Мы успешно сдали N экзаменов",
согласовав слово "экзамен" с числом .
"""
N = int(input("Введите число сданных экзаменов: "))
print("Мы успешно сдали ", N, " экзаменов")

```

3) ×

Введите число сданных экзаменов: >? 12

Мы успешно сдали 12 экзаменов

Рисунок 3.1 Программа к инд. заданию №1



Рисунок 3.2 UML – диаграмма к программе инд. задания 1

```

"""
Треугольник задан координатами своих вершин. Определить принадлежит ли данная точка
треугольнику. Координаты вершин треугольника и координаты точки задать самостоятельно.
"""
ax = int(input('Введите координаты первой вершины:\n'))
ay = int(input())
bx = int(input('Введите координаты второй вершины:\n'))
by = int(input())
cx = int(input('Введите координаты третьей вершины:\n'))
cy = int(input())
xx = int(input('Введите координаты точки:\n'))
xy = int(input())

"""
Если точка лежит внутри треугольника, то сумма площадей образованных ей треугольников
равна площади данного треугольника
"""
if(abs((ax - xx)*(by - xy) - (bx - xx)*(ay - xy)) +
    abs((ax - cx)*(xy - cy) - (xx - cx)*(ay - cy)) +
    abs((xx - cx)*(by - cy) - (bx - cx)*(xy - cy)) ==
    abs((ax - cx)*(by - cy) - (bx - cx)*(ay - cy))):
    print('Точка ВХОДИТ в треугольник')
else:
    print('Точка НЕ ВХОДИТ в треугольник')

```

#### Python Console

```

Введите координаты первой вершины:
>? 1
>? 1
Введите координаты второй вершины:
>? 3
>? 5
Введите координаты третьей вершины:
>? 6
>? 0
Введите координаты точки:
>? 2
>? 3
Точка ВХОДИТ в треугольник

```

Рисунок 3.3 Программа к инд. заданию №1

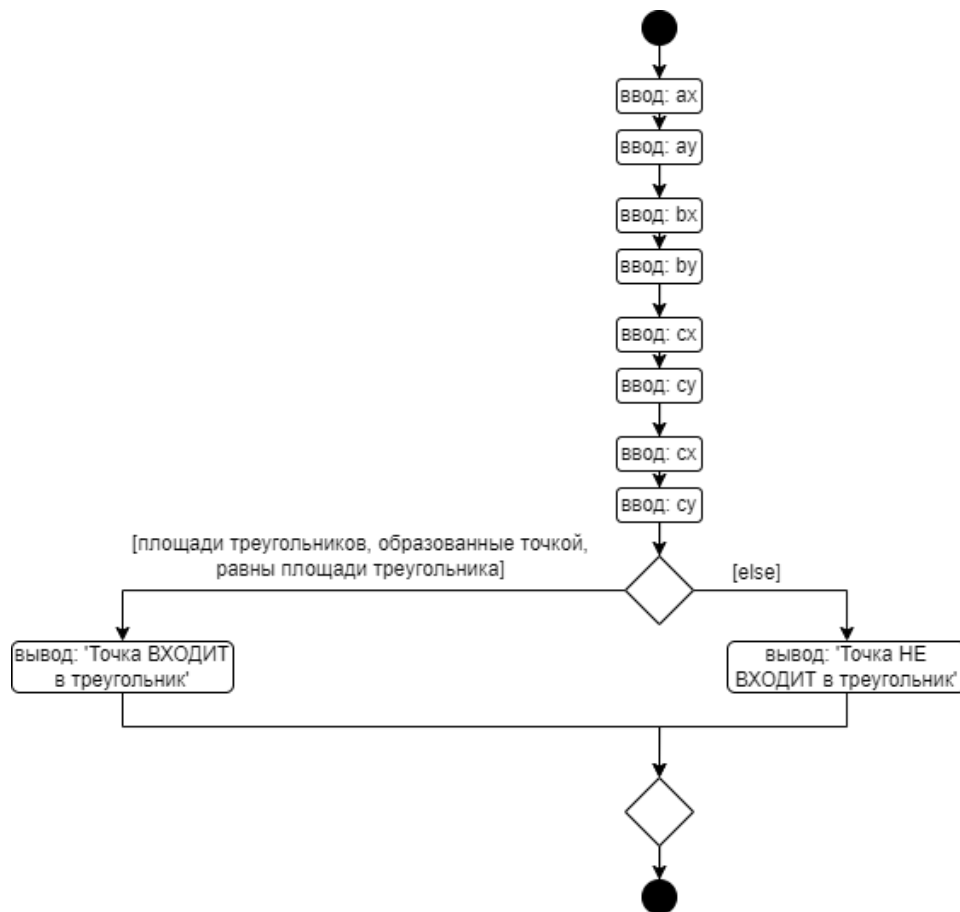


Рисунок 3.4 UML – диаграмма к программе инд. задания 2

```

NumDay = int(input('add number day'))
m = 0
for mounth in range(1, 13):
    if mounth == 2:
        a = 28
    elif (mounth == 1 or mounth == 3 or mounth == 5 or mounth == 7 or mounth == 8 or mounth == 10 or mounth == 12):
        a = 31
    else:
        a = 30
    m += a
    if (NumDay - m <= 31):
        day = NumDay - m
        if (NumDay > 31):
            mounth += 1
            break
        else:
            break

if (mounth < 10):
    print(NumDay, f'-й день года - ', day, '.0', mounth, sep='')
else:
    print(NumDay, '-й день года - ', day, '.', mounth, sep='')

```

ind1 (1) ×

Python Console

add number day-> 68

68-й день года - 9.03

Рисунок 3.5 Программа к инд. заданию №1



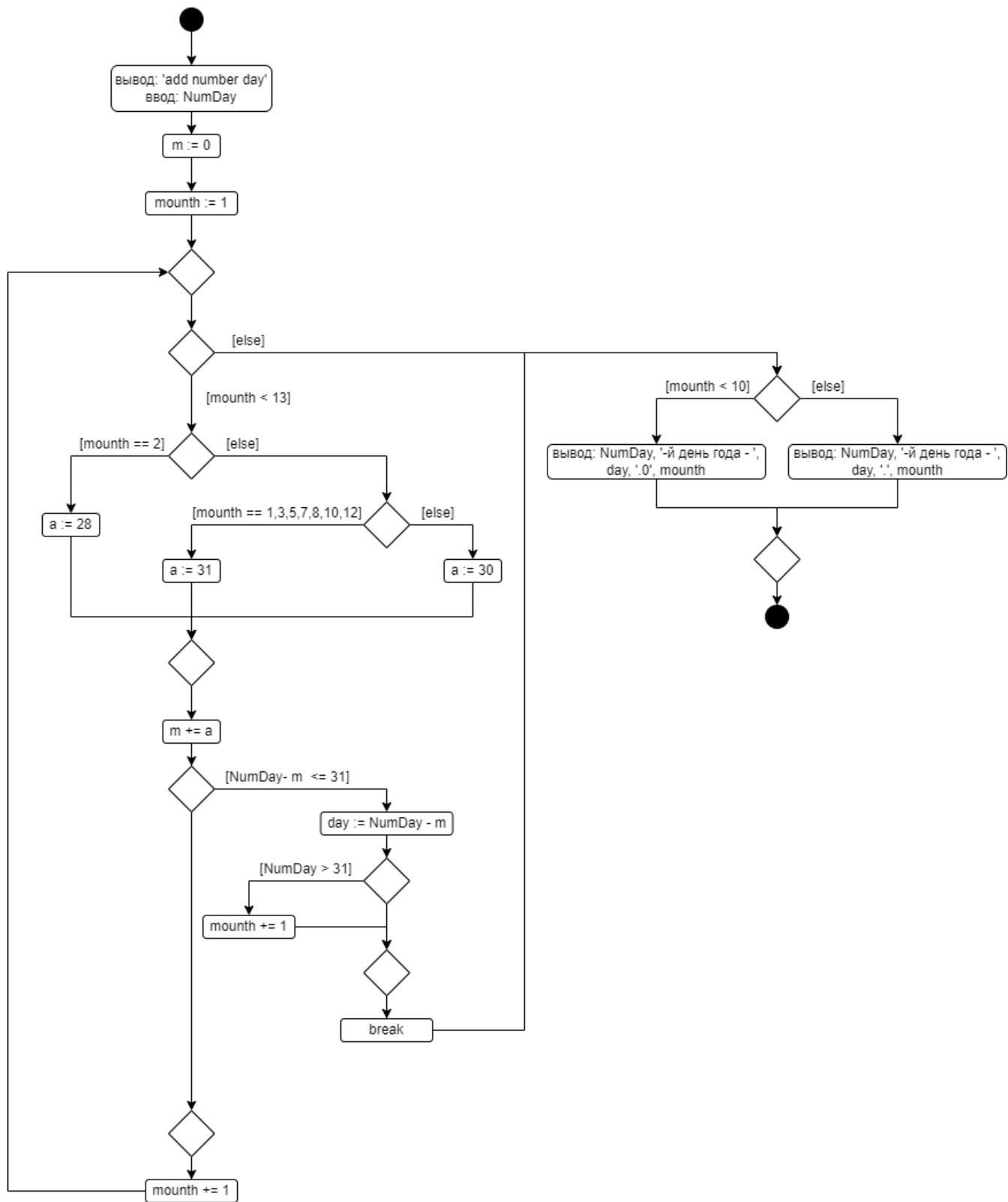


Рисунок 3.6 UML – диаграмма к программе инд. задания 3

```
import math
import sys

EULER = 0.5772156649015328606
EPS = 10 ** -10

x = float(input("add value for x: "))
if x == 0:
    print("Illegal value of x", file=sys.stderr)
    exit(1)
a = -x ** 2 / 4
S, n = a, 1
while math.fabs(a) > EPS:
    a *= (-1 * x ** 2 * 2 * 2 * n) / (2 * (n + 1)) ** 2
    S += a
    n += 1
print(f"Ci({x}) = {EULER + math.log(math.fabs(x)) + S}")
```

Console × ind1 ×

Python Console

add value for x: > 32

Ci(32.0) = 1.070004389939887e+202

Рисунок 3.7 Программа для задачи повышенной сложности.

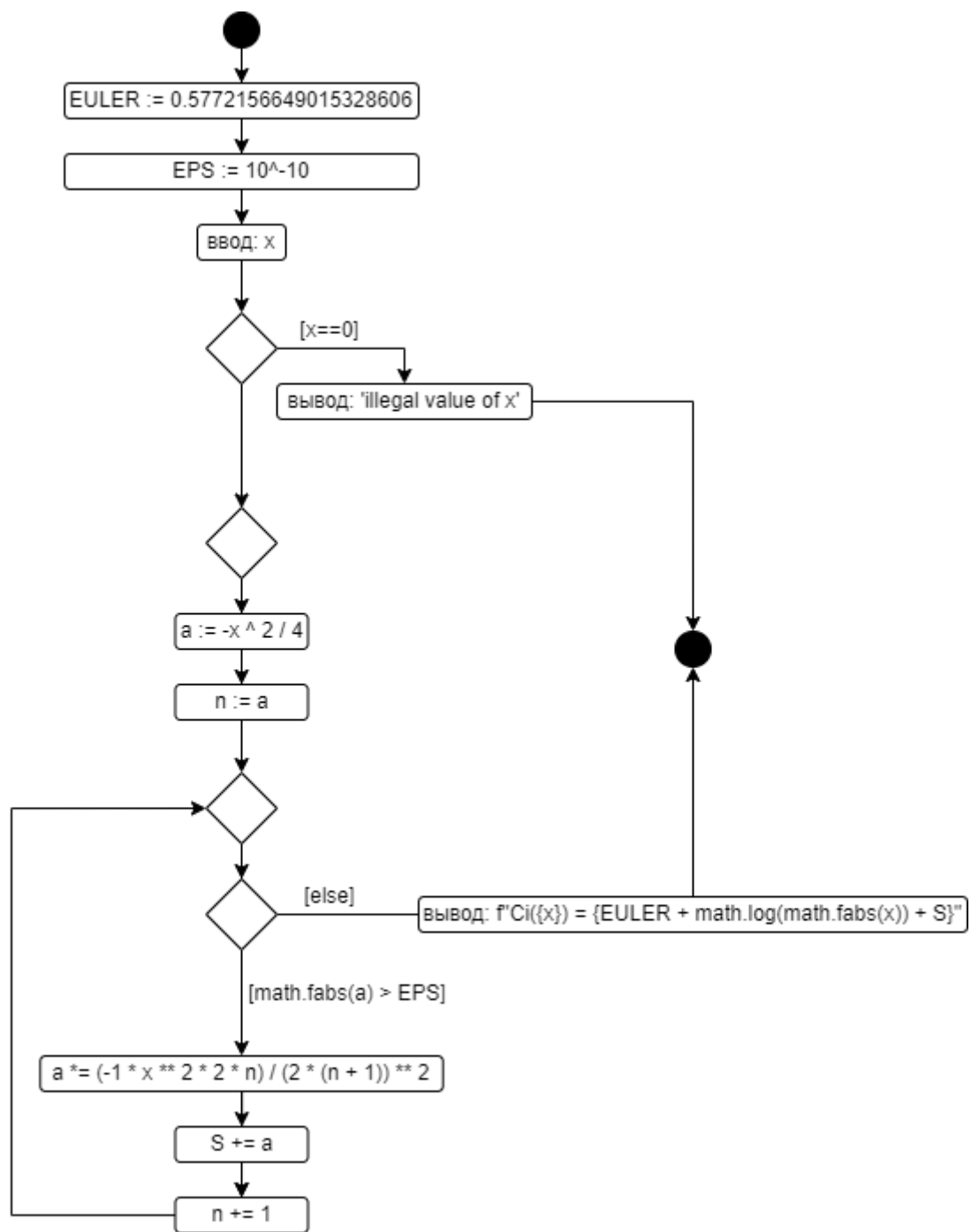


Рисунок 3.8 UML – диаграмма деятельности программы для усложненного задания

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

```

C:\Users\adamk\OneDrive\Рабочий стол\ЛР 2.2\rep_2.2>git commit -m "first com"
[develop 2ba1f0c] first com
18 files changed, 43 insertions(+)
create mode 100644 .idea/.gitignore
create mode 100644 .idea/inspectionProfiles/profiles_settings.xml
create mode 100644 .idea/misc.xml
create mode 100644 .idea/modules.xml
create mode 100644 .idea/rep_2.2.iml
create mode 100644 .idea/vcs.xml
create mode 100644 "\320\241\321\205\320\265\320\274\321\213\ind1.drawio"
create mode 100644 "\320\241\321\205\320\265\320\274\321\213\ind1.png"
create mode 100644 "\320\241\321\205\320\265\320\274\321\213\ind2.drawio"
create mode 100644 "\320\241\321\205\320\265\320\274\321\213\ind2.png"
create mode 100644 "\320\241\321\205\320\265\320\274\321\213\ind3.drawio"
create mode 100644 "\320\241\321\205\320\265\320\274\321\213\ind3.png"
create mode 100644 "\320\241\321\205\320\265\320\274\321\213\primer4.drawio"
create mode 100644 "\320\241\321\205\320\265\320\274\321\213\primer4.png"
create mode 100644 "\320\241\321\205\320\265\320\274\321\213\primer5.drawio"
create mode 100644 "\320\241\321\205\320\265\320\274\321\213\primer5.png"
create mode 100644 "\320\241\321\205\320\265\320\274\321\213\usl.drawio"
create mode 100644 "\320\241\321\205\320\265\320\274\321\213\usl.png"

C:\Users\adamk\OneDrive\Рабочий стол\ЛР 2.2\rep_2.2>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

C:\Users\adamk\OneDrive\Рабочий стол\ЛР 2.2\rep_2.2>git merge develop
Updating a9eb24d..2ba1f0c
Fast-forward
 .idea/.gitignore                | 3 +++
 .idea/inspectionProfiles/profiles_settings.xml | 6 ++++++
 .idea/misc.xml                  | 4 +++++
 .idea/modules.xml               | 8 ++++++++
 .idea/rep_2.2.iml               |10 ++++++++
 .idea/vcs.xml                   | 6 ++++++
 "\320\241\321\205\320\265\320\274\321\213\ind1.drawio" | 1 +
 "\320\241\321\205\320\265\320\274\321\213\ind1.png"   | Bin 0 -> 5555 bytes
 "\320\241\321\205\320\265\320\274\321\213\ind2.drawio" | 1 +
 "\320\241\321\205\320\265\320\274\321\213\ind2.png"   | Bin 0 -> 26360 bytes
 "\320\241\321\205\320\265\320\274\321\213\ind3.drawio" | 1 +
 "\320\241\321\205\320\265\320\274\321\213\ind3.png"   | Bin 0 -> 56179 bytes
 .../primer4.drawio              | 1 +
 "\320\241\321\205\320\265\320\274\321\213\primer4.png" | Bin 0 -> 26231 bytes
 .../primer5.drawio              | 1 +
 "\320\241\321\205\320\265\320\274\321\213\primer5.png" | Bin 0 -> 29002 bytes
 "\320\241\321\205\320\265\320\274\321\213\usl.drawio" | 1 +
 "\320\241\321\205\320\265\320\274\321\213\usl.png"   | Bin 0 -> 30789 bytes
18 files changed, 43 insertions(+)
create mode 100644 .idea/.gitignore
create mode 100644 .idea/inspectionProfiles/profiles_settings.xml
create mode 100644 .idea/misc.xml
create mode 100644 .idea/modules.xml
create mode 100644 .idea/rep_2.2.iml
create mode 100644 .idea/vcs.xml
create mode 100644 "\320\241\321\205\320\265\320\274\321\213\ind1.drawio"
create mode 100644 "\320\241\321\205\320\265\320\274\321\213\ind1.png"
create mode 100644 "\320\241\321\205\320\265\320\274\321\213\ind2.drawio"
create mode 100644 "\320\241\321\205\320\265\320\274\321\213\ind2.png"
create mode 100644 "\320\241\321\205\320\265\320\274\321\213\ind3.drawio"
create mode 100644 "\320\241\321\205\320\265\320\274\321\213\ind3.png"
create mode 100644 "\320\241\321\205\320\265\320\274\321\213\primer4.drawio"
create mode 100644 "\320\241\321\205\320\265\320\274\321\213\primer4.png"
create mode 100644 "\320\241\321\205\320\265\320\274\321\213\primer5.drawio"
create mode 100644 "\320\241\321\205\320\265\320\274\321\213\primer5.png"
create mode 100644 "\320\241\321\205\320\265\320\274\321\213\usl.drawio"
create mode 100644 "\320\241\321\205\320\265\320\274\321\213\usl.png"

C:\Users\adamk\OneDrive\Рабочий стол\ЛР 2.2\rep_2.2>git push
Enumerating objects: 24, done.
Counting objects: 100% (24/24), done.
Delta compression using up to 8 threads
Compressing objects: 100% (21/21), done.
Writing objects: 100% (23/23), 157.67 KiB | 12.13 MiB/s, done.
Total 23 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/AdamKh/rep_2.2.git
 a9eb24d..2ba1f0c  main -> main

C:\Users\adamk\OneDrive\Рабочий стол\ЛР 2.2\rep_2.2>

```

Рисунок 4.1 Работа в GIT CMD

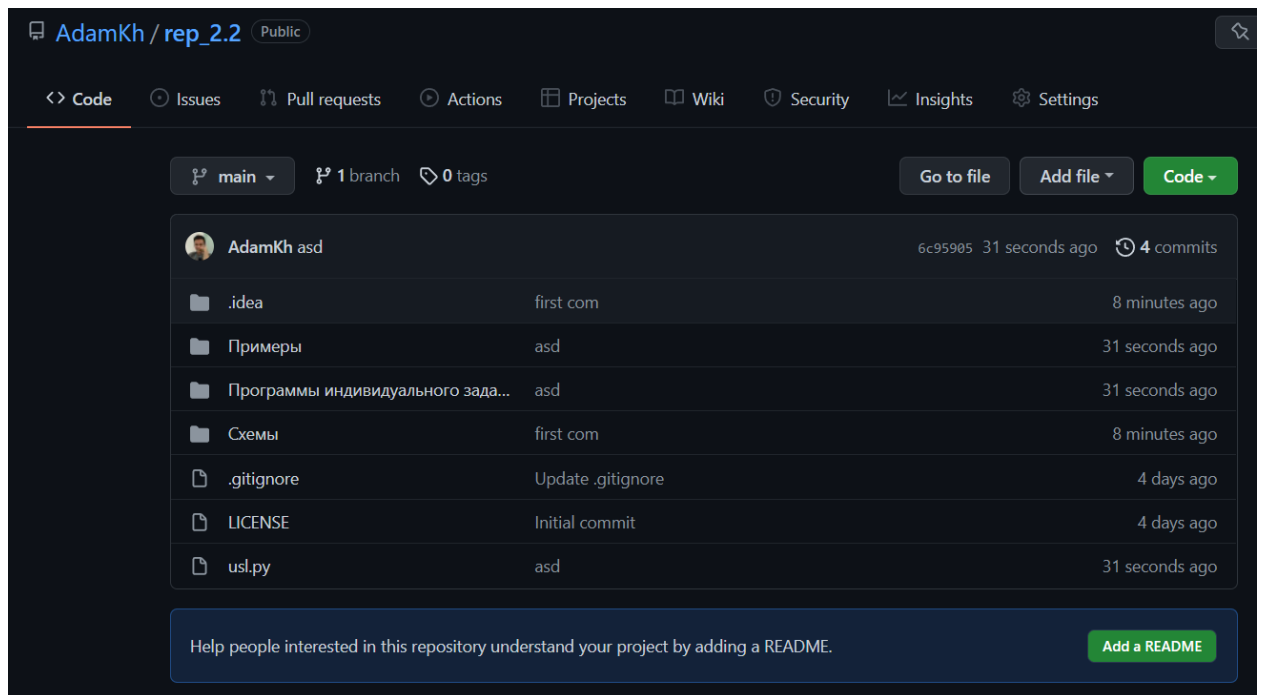


Рисунок 4.2 Изменения на уд. сервере

### 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.