

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

**Институт перспективной инженерии
Департамент цифровых, робототехнических систем и электроники**

**ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №3
дисциплины «Программирование на Python»**

Выполнила:
Федорова Дарья Юрьевна
2 курс, группа ИВТ-б-о-24-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р. А.

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2025 г.

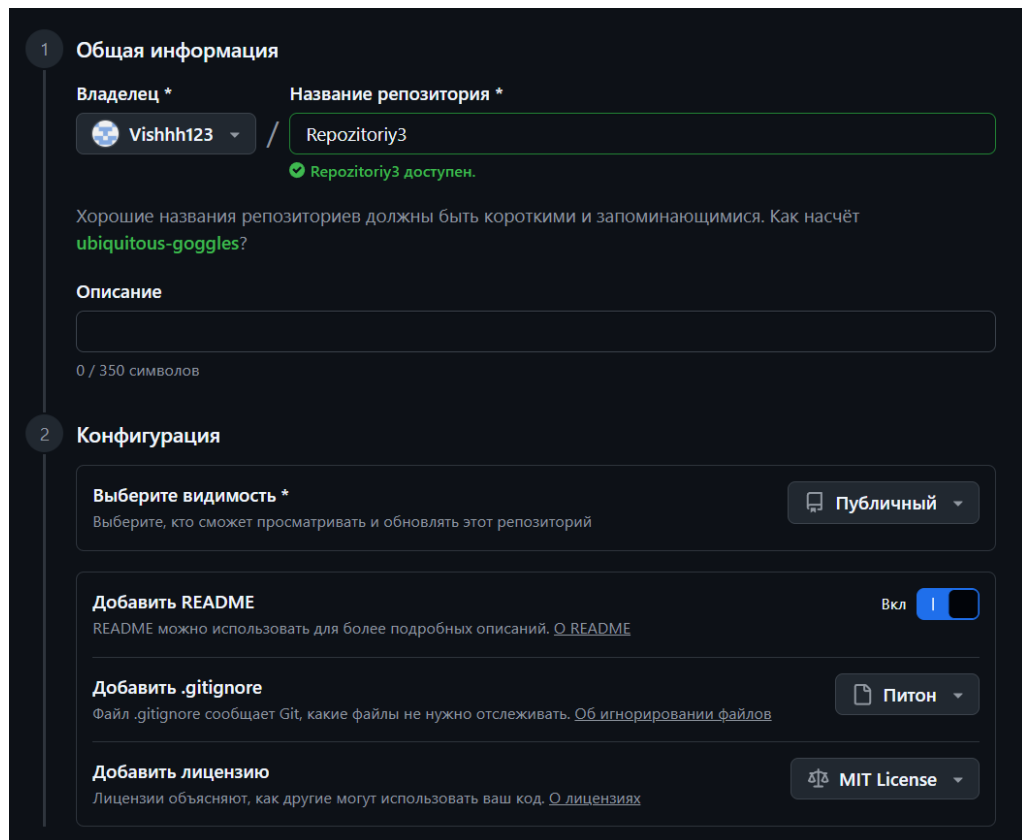
Вариант 25

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,

Ссылка на репозиторий: <https://github.com/Vishhh123/Repozitoriy3>

Ход работы:



1 Общая информация

Владелец * Vishhh123 / Название репозитория * Repozitoriy3

Repozitoriy3 доступен.

Хорошие названия репозитория должны быть короткими и запоминающимися. Как насчёт ubiquitous-goggles?

Описание

0 / 350 символов

2 Конфигурация

Выберите видимость * Публичный

Выберите, кто сможет просматривать и обновлять этот репозиторий

Добавить README Вкл

README можно использовать для более подробных описаний. [О README](#)

Добавить .gitignore Питон

Файл .gitignore сообщает Git, какие файлы не нужно отслеживать. [Об игнорировании файлов](#)

Добавить лицензию MIT License

Лицензии объясняют, как другие могут использовать ваш код. [О лицензиях](#)

Рис 1 - создание общедоступного репозитория с лицензией MIT, языком программирования Python.

```
C:\Users\user>git clone https://github.com/Vishhh123/Repozitoriy3.git
Cloning into 'Repozitoriy3'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (5/5), done.
```

Рис. 2 - клонирование репозитория на устройство.

```
C:\Users\user>cd Repozitoriy3
C:\Users\user\Repozitoriy3>git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Рис. 3 - переход к репозиторию, его проверка.

```

C:\Users\user\Repozitoriy3>git flow init -d
Using default branch names.

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:/Users/user/Repozitoriy3/.git/hooks]

```

Рис. 4 – создание ветвления git-flow.

```

C:\Users\user\Repozitoriy3>git branch -a
* develop
main
remotes/origin/HEAD -> origin/main
remotes/origin/main

```

Рис. 5 – проверка активных веток git-flow (main - основная стабильная ветка, develop - ветка для разработки, origin/main - копия main на GitHub, origin/HEAD - техническая ветка, указывает какая ветка основная)

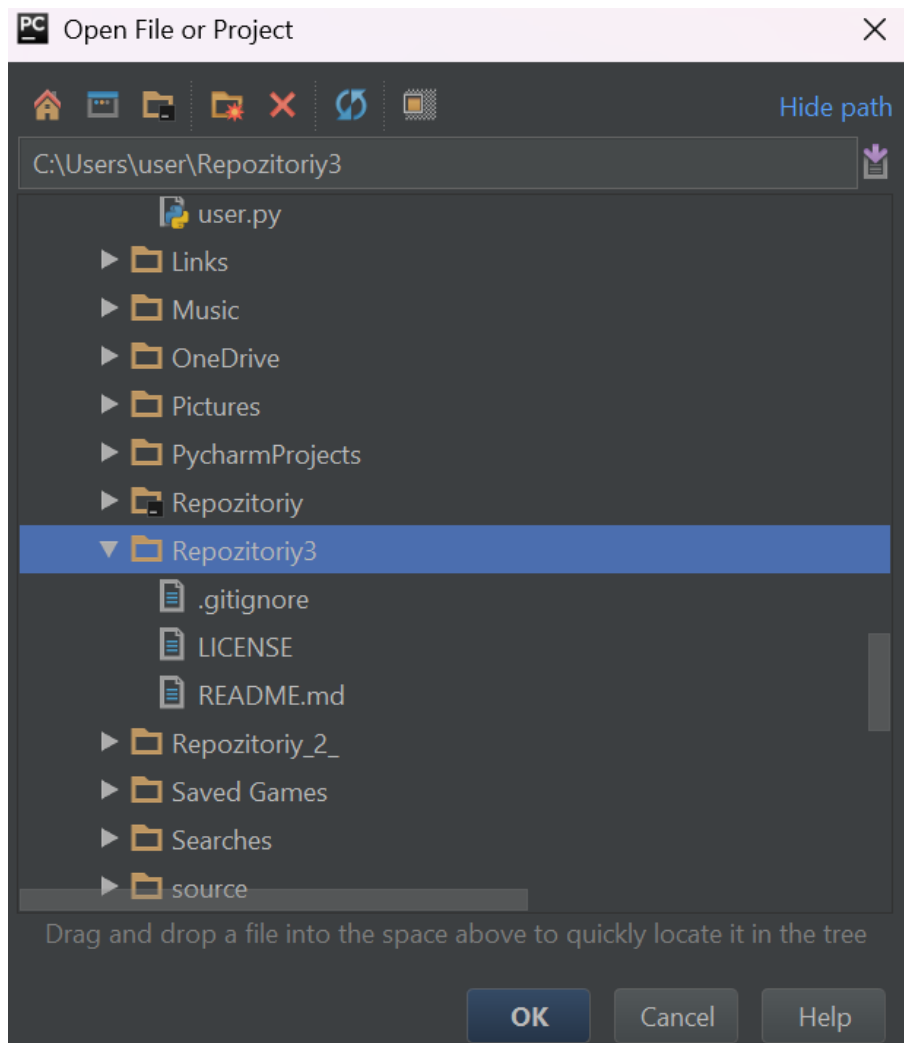


Рис. 6 - создание в PyCharm проекта из существующих файлов (file -> open -> путь к файлу)

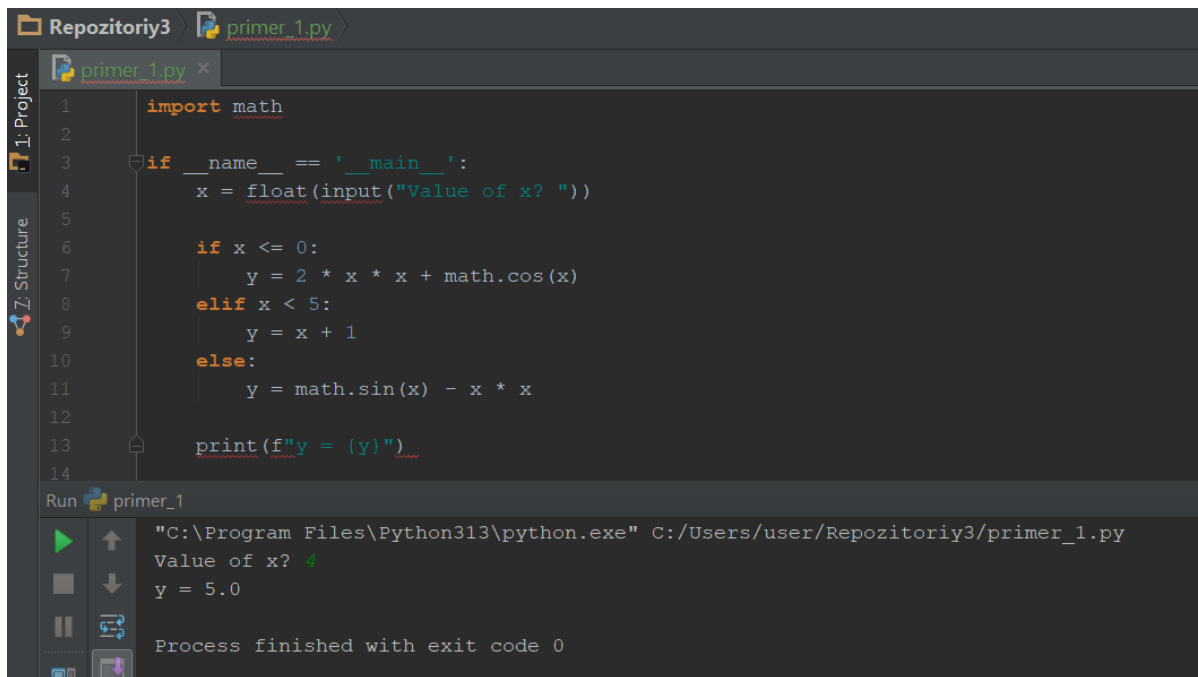


Рис. 7 – код к примеру 1.

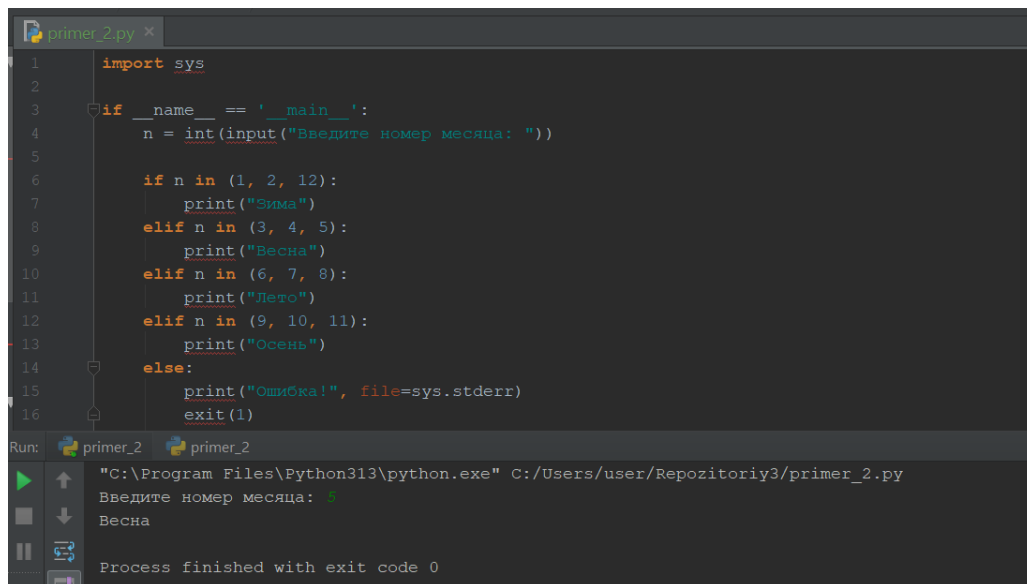
```
C:\Users\user\Repository3>git add .
warning: in the working copy of '.idea/Repository3.iml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '.idea/misc.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '.idea/modules.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '.idea/workspace.xml', LF will be replaced by CRLF the next time Git touches it

C:\Users\user\Repository3>git commit -m"Код для функции к заданию 1"
[develop a66086e] Код для функции к заданию 1
6 files changed, 92 insertions(+), 1 deletion(-)
create mode 100644 .idea/Repository3.iml
create mode 100644 .idea/misc.xml
create mode 100644 .idea/modules.xml
create mode 100644 .idea/workspace.xml
create mode 100644 primer_1.py
```

Рис. 8 – сохранение, добавление коммита.

```
C:\Users\user\Repository3> git push --set-upstream origin develop
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 24 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 2.03 KiB | 1.01 MiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/Vishhh123/Repository3/pull/new/develop
remote:
To https://github.com/Vishhh123/Repository3.git
 * [new branch]      develop -> develop
branch 'develop' set up to track 'origin/develop'.
```

Рис. 9 - отправка на Github.



```
1 import sys
2
3 if __name__ == '__main__':
4     n = int(input("Введите номер месяца: "))
5
6     if n in (1, 2, 12):
7         print("Зима")
8     elif n in (3, 4, 5):
9         print("Весна")
10    elif n in (6, 7, 8):
11        print("Лето")
12    elif n in (9, 10, 11):
13        print("Осень")
14    else:
15        print("Ошибка!", file=sys.stderr)
16        exit(1)
```

Run: primer_2 primer_2

"C:\Program Files\Python313\python.exe" C:/Users/user/Repozitoriy3/primer_2.py

Введите номер месяца: 3

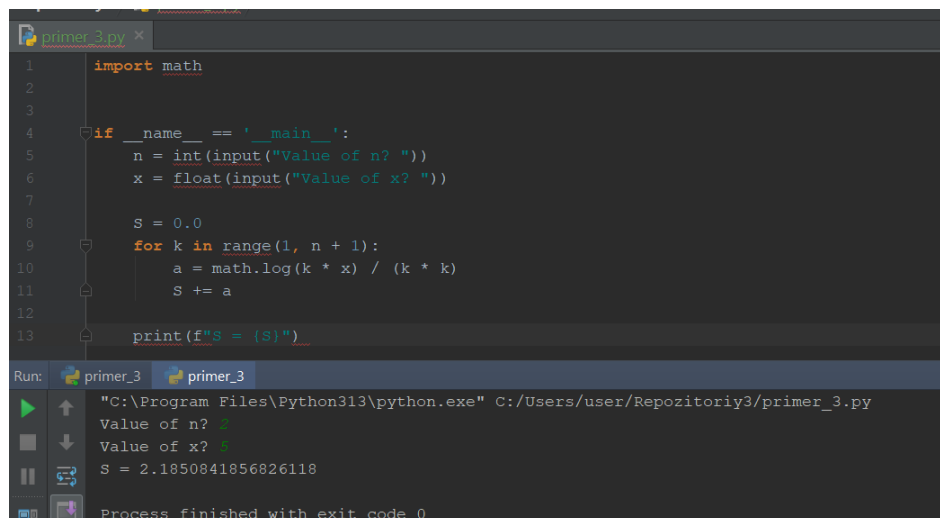
Весна

Process finished with exit code 0

Рис. 10 – код к примеру 2.

```
C:\Users\user\Repozitoriy3>git add .
C:\Users\user\Repozitoriy3>git commit -m"Код к примеру 2"
[develop baddb3b] Код к примеру 2
1 file changed, 18 insertions(+)
create mode 100644 primer_2.py
C:\Users\user\Repozitoriy3> git push --set-upstream origin develop
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 24 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 536 bytes | 536.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Vishhh123/Repozitoriy3.git
 a66086e..baddb3b develop -> develop
branch 'develop' set up to track 'origin/develop'.
```

Рис. 11 – сохранение, добавление коммита, отправка на Github.



```
1 import math
2
3
4 if __name__ == '__main__':
5     n = int(input("Value of n? "))
6     x = float(input("Value of x? "))
7
8     S = 0.0
9     for k in range(1, n + 1):
10        a = math.log(k * x) / (k * k)
11        S += a
12
13    print(f"S = {S}")
```

Run: primer_3 primer_3

"C:\Program Files\Python313\python.exe" C:/Users/user/Repozitoriy3/primer_3.py

Value of n? 3

Value of x? 3

S = 2.1850841856826118

Process finished with exit code 0

Рис. 12 – код к примеру 3.

```

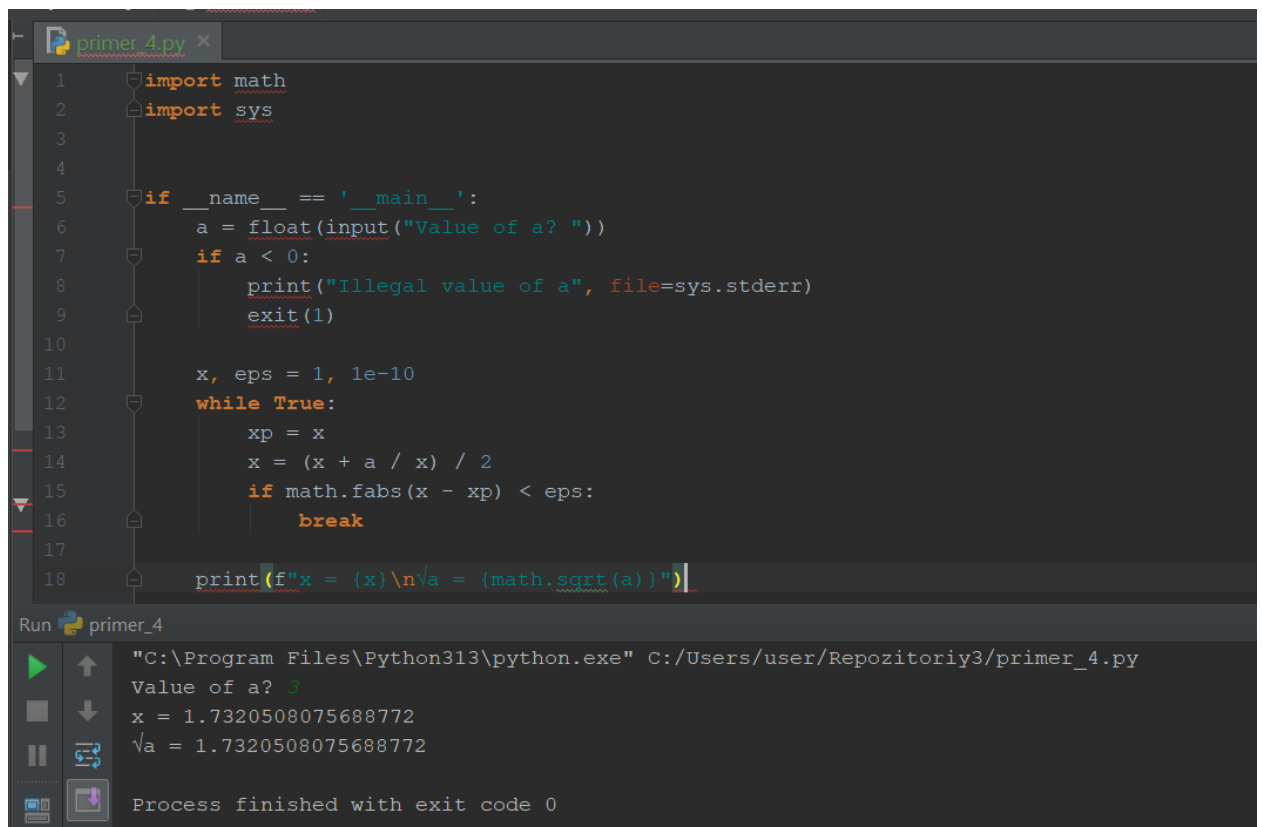
C:\Users\user\Repozitoriy3>git add .

C:\Users\user\Repozitoriy3>git commit -m"Код к примеру 3"
[develop 451c056] Код к примеру 3
2 files changed, 14 insertions(+), 1 deletion(-)
create mode 100644 primer_3.py

C:\Users\user\Repozitoriy3> git push --set-upstream origin develop
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 24 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 527 bytes | 527.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/Vishhh123/Repozitoriy3.git
   baddb3b..451c056  develop -> develop
branch 'develop' set up to track 'origin/develop'.

```

Рис. 13 – сохранение, добавление коммита, отправка на Github.



```

1  import math
2  import sys
3
4
5  if __name__ == '__main__':
6      a = float(input("Value of a? "))
7      if a < 0:
8          print("Illegal value of a", file=sys.stderr)
9          exit(1)
10
11      x, eps = 1, 1e-10
12      while True:
13          xp = x
14          x = (x + a / x) / 2
15          if math.fabs(x - xp) < eps:
16              break
17
18      print(f"x = {x}\n√a = {math.sqrt(a)}")

```

Run primer_4

```

C:\Program Files\Python313\python.exe C:/Users/user/Repozitoriy3/primer_4.py
Value of a? 2
x = 1.7320508075688772
√a = 1.7320508075688772
Process finished with exit code 0

```

Рис. 14 – код к примеру 4.

```

C:\Users\user\Repozitoriy3>git add .

C:\Users\user\Repozitoriy3>git commit -m"Код к заданию 4"
[develop 4659889] Код к заданию 4
1 file changed, 18 insertions(+)
create mode 100644 primer_4.py

C:\Users\user\Repozitoriy3> git push --set-upstream origin develop
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 24 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 532 bytes | 532.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Vishhh123/Repozitoriy3.git
   451c056..4659889  develop -> develop
branch 'develop' set up to track 'origin/develop'.

```

Рис. 15 - сохранение, добавление коммита, отправка на Github.

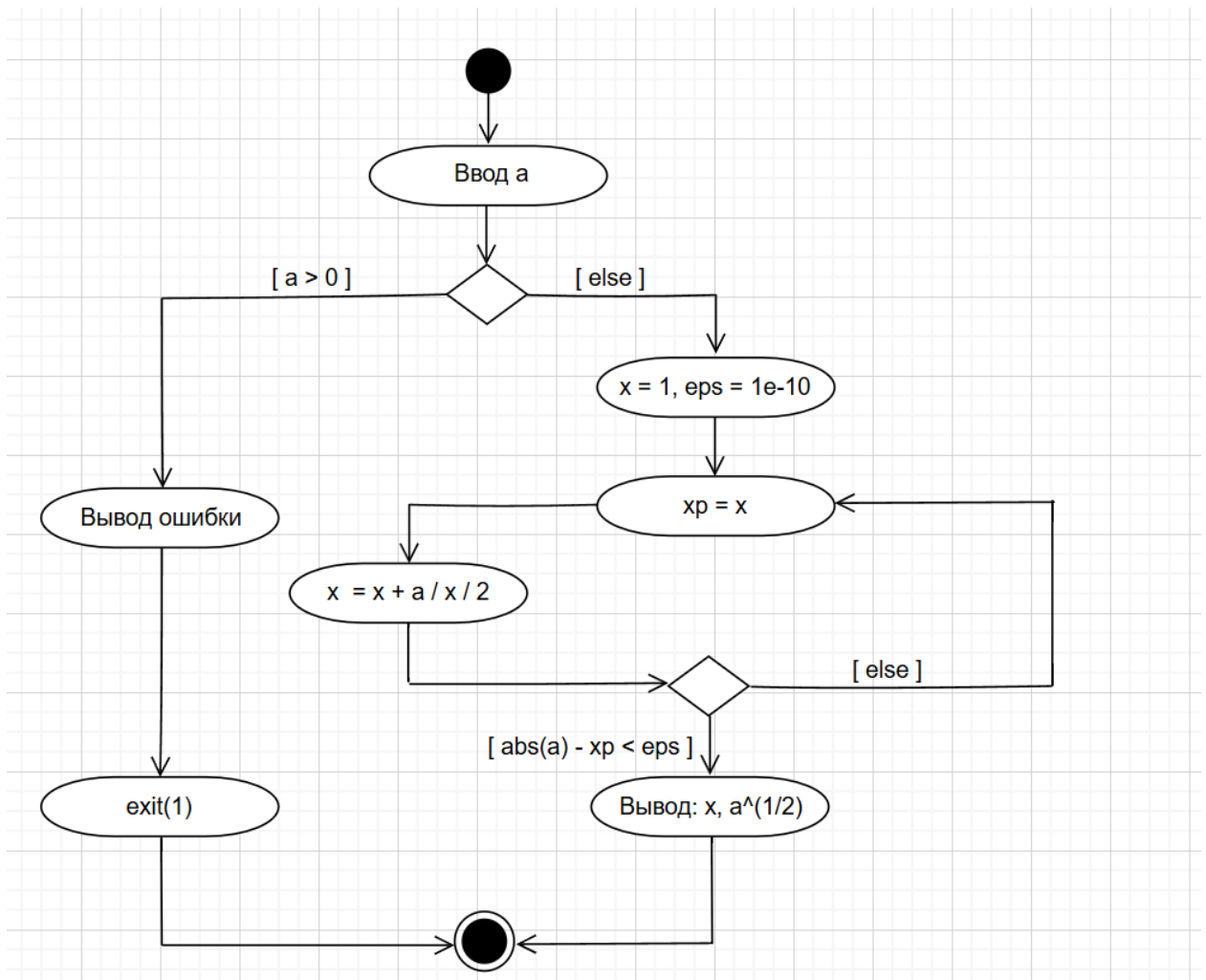


Рис. 16 - UML-диаграмма к примеру 4.

```
primer_5.py x
1 import math
2 import sys
3
4 # Постоянная Эйлера.
5 EULER = 0.5772156649015328606
6 # Точность вычислений.
7 EPS = 1e-10
8
9 if __name__ == '__main__':
10     x = float(input("Value of x? "))
11     if x == 0:
12         print("Illegal value of x", file=sys.stderr)
13         exit(1)
14
15     a = x
16     S, k = a, 1
17
18     # Найти сумму членов ряда.
19     while math.fabs(a) > EPS:
20         a *= x * k / (k + 1) ** 2
21         S += a
22         k += 1
23
24     # Вывести значение функции.
25     print(f"Ei(x) = {EULER + math.log(math.fabs(x)) + S}")
26
Run primer_5
C:\Program Files\Python313\python.exe C:/Users/user/Repozitoriy3/primer_5.py
Value of x? 5
Ei(x) = 40.18527535579794
Process finished with exit code 0
```

Рис. 17 – код к примеру 5.

```
C:\Users\user\Repozitoriy3>git add .
C:\Users\user\Repozitoriy3>git commit -m"Код к примеру 5"
[develop 56a0361] Код к примеру 5
1 file changed, 29 insertions(+)
create mode 100644 primer_5.py
C:\Users\user\Repozitoriy3> git push --set-upstream origin develop
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 24 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 705 bytes | 705.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Vishhh123/Repozitoriy3.git
4659889..56a0361 develop -> develop
branch 'develop' set up to track 'origin/develop'.
```

Рис. 18 - сохранение, добавление коммита, отправка на Github.

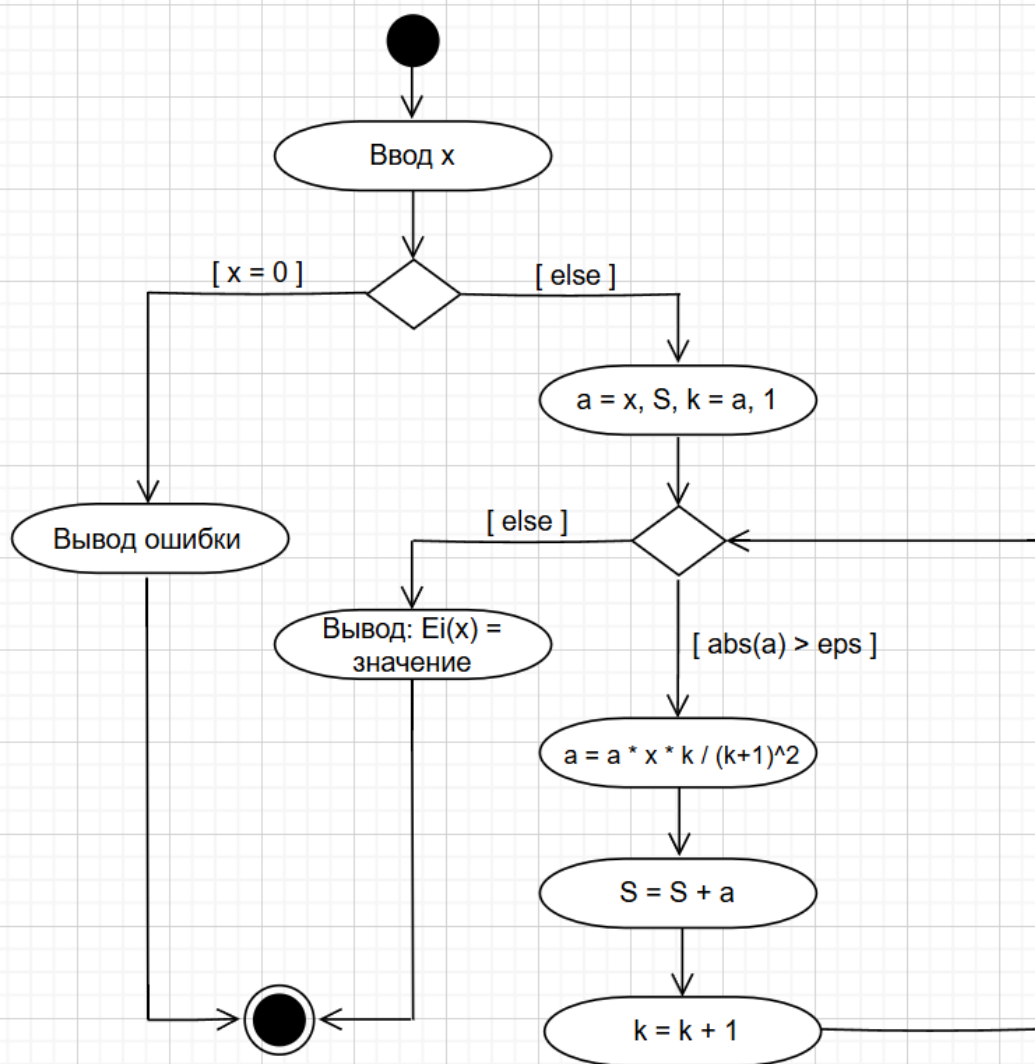


Рис. 19 - UML-диаграмма к примеру 5.

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

Рис. 20 – индивидуальное задание 1.

```

1  import sys
2
3
4  if __name__ == '__main__':
5      m = int(input('введите день недели: '))
6
7      if m == 1:
8          print('Понедельник')
9      elif m == 2:
10         print('Вторник')
11     elif m == 3:
12         print('Среда')
13     elif m == 4:
14         print('Четверг')
15     elif m == 5:
16         print('Пятница')
17     elif m == 6:
18         print('Суббота')
19     elif m == 7:
20         print('Воскресенье')
21     else:
22         print('Ошибка!', file=sys.stderr)
23         exit(1)
  
```

Run individualnoe_1

"C:\Program Files\Python313\python.exe" C:/Users/user/Repozitoriy3/individualnoe_1.py
 введите день недели:
 Ошибка!
 Process finished with exit code 1

Рис. 21 – код к индивидуальному заданию 1.

```

C:\Users\user\Repozitoriy3>git add .
C:\Users\user\Repozitoriy3>>git commit -m"Индивидуальное задание 1"
"commit" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.
C:\Users\user\Repozitoriy3> git push --set-upstream origin develop
branch 'develop' set up to track 'origin/develop'.
Everything up-to-date
  
```

Рис. 22 - сохранение, добавление коммита, отправка на Github.

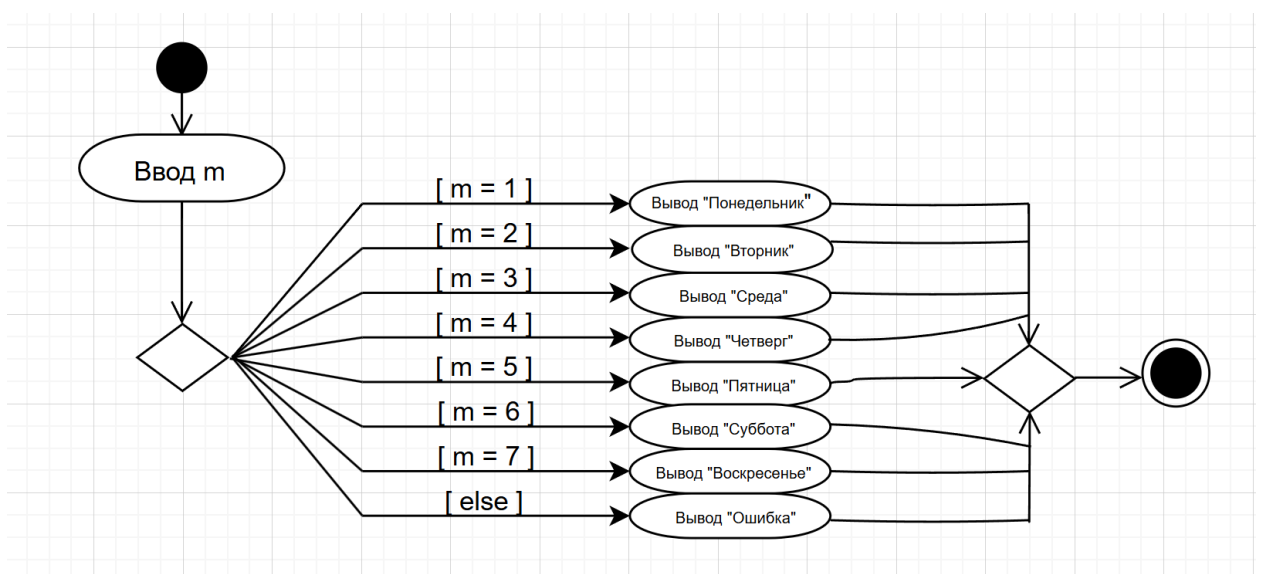
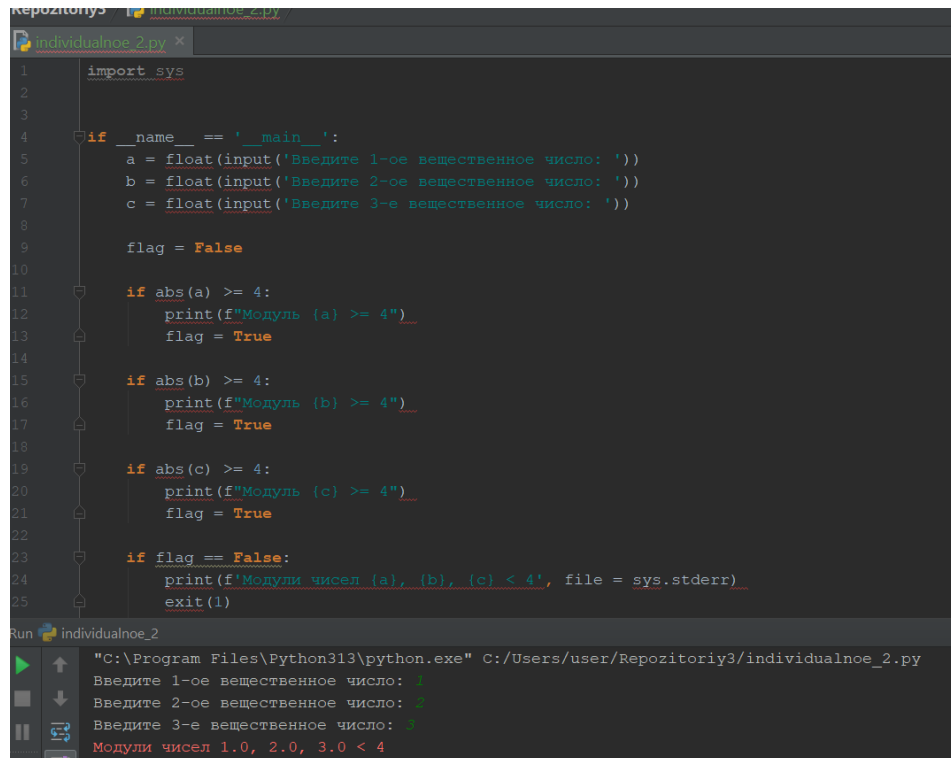


Рис. 23 - UML-диаграмма к инд. заданию 1.

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

Рис. 24 - индивидуальное задание 2.



```
1 import sys
2
3
4 if __name__ == '__main__':
5     a = float(input('Введите 1-ое вещественное число: '))
6     b = float(input('Введите 2-ое вещественное число: '))
7     c = float(input('Введите 3-е вещественное число: '))
8
9     flag = False
10
11     if abs(a) >= 4:
12         print(f'Модуль {a} >= 4')
13         flag = True
14
15     if abs(b) >= 4:
16         print(f'Модуль {b} >= 4')
17         flag = True
18
19     if abs(c) >= 4:
20         print(f'Модуль {c} >= 4')
21         flag = True
22
23     if flag == False:
24         print(f'Модули чисел {a}, {b}, {c} < 4', file = sys.stderr)
25         exit(1)
```

Run individualnoe_2

```
"C:\Program Files\Python313\python.exe" C:/Users/user/Repozitoriy3/individualnoe_2.py
Введите 1-ое вещественное число: 1
Введите 2-ое вещественное число: 2
Введите 3-е вещественное число: 3
Модули чисел 1.0, 2.0, 3.0 < 4
```

Рис. 25 - код к индивидуальному заданию 2.

```
C:\Users\user\Repozitoriy3>git add .
C:\Users\user\Repozitoriy3>git commit -m"Индивидуальное задание 2"
[develop 5bcb3f0] Индивидуальное задание 2
3 files changed, 54 insertions(+)
create mode 100644 git
create mode 100644 individualnoe_1.py
create mode 100644 individualnoe_2.py
C:\Users\user\Repozitoriy3> git push --set-upstream origin develop
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 24 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 923 bytes | 461.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Vishhh123/Repozitoriy3.git
56a0361..5bcb3f0 develop -> develop
branch 'develop' set up to track 'origin/develop'.
```

Рис. 26 - сохранение, добавление коммита, отправка на Github.

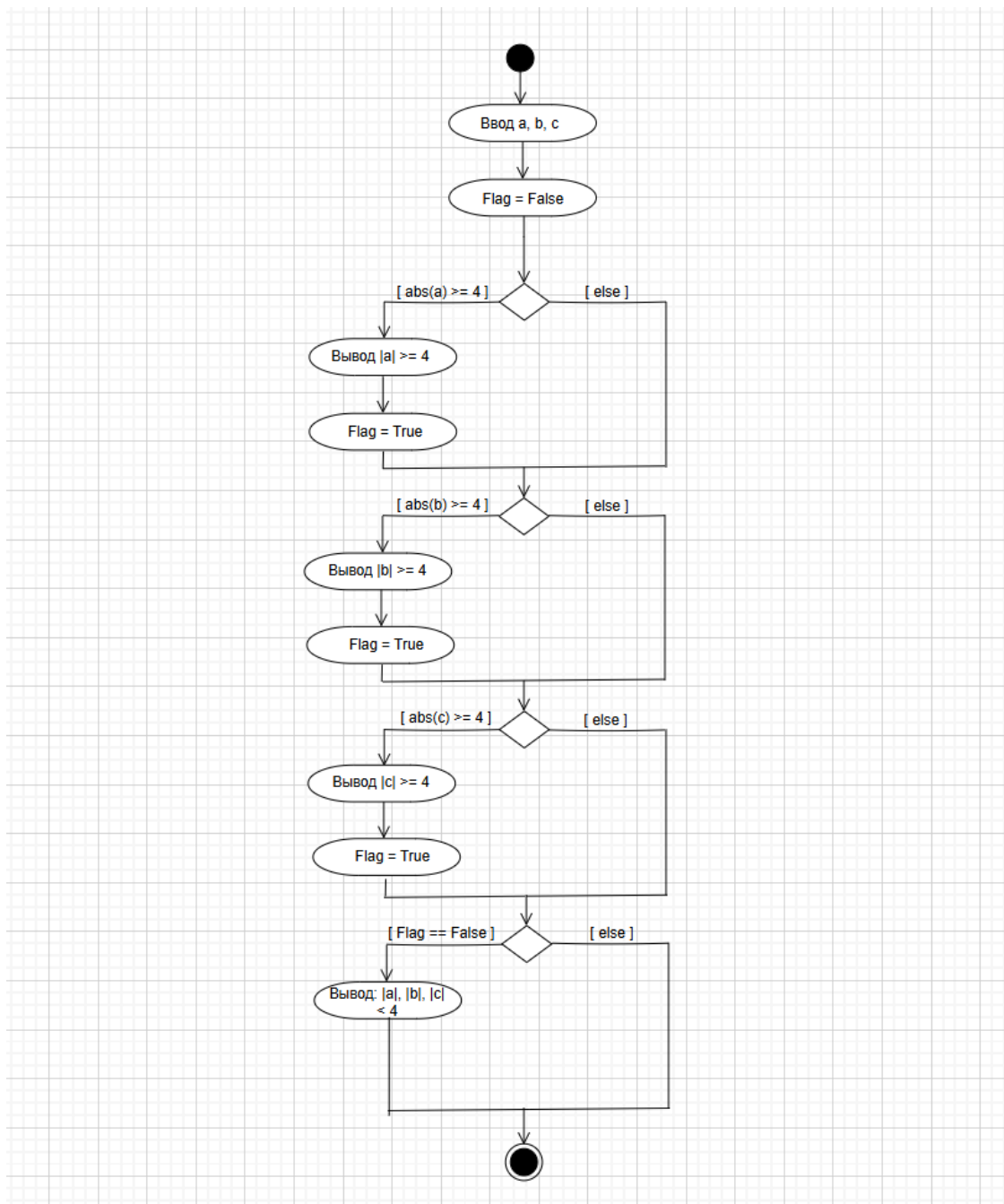
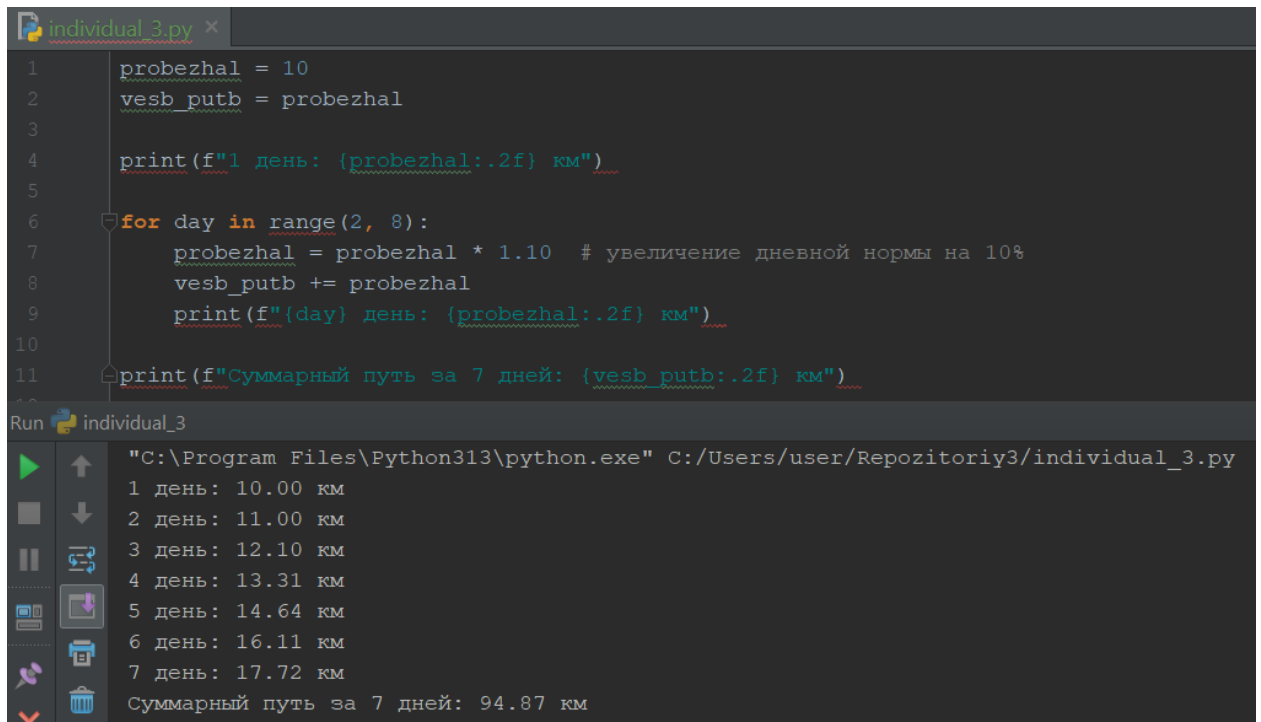


Рис. 27 - UML-диаграмма к инд. заданию 2.

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?

Рис. 28 - индивидуальное задание 3.



```
individual_3.py
1  probezhal = 10
2  vesb_putb = probezhal
3
4  print(f"1 день: {probezhal:.2f} км")
5
6  for day in range(2, 8):
7      probezhal = probezhal * 1.10 # увеличение дневной нормы на 10%
8      vesb_putb += probezhal
9      print(f"{day} день: {probezhal:.2f} км")
10
11 print(f"Суммарный путь за 7 дней: {vesb_putb:.2f} км")
```

Run individual_3

```
"C:\Program Files\Python313\python.exe" C:/Users/user/Repozitoriy3/individual_3.py
1 день: 10.00 км
2 день: 11.00 км
3 день: 12.10 км
4 день: 13.31 км
5 день: 14.64 км
6 день: 16.11 км
7 день: 17.72 км
Суммарный путь за 7 дней: 94.87 км
```

Рис. 29 - код к индивидуальному заданию 3.

```
C:\Users\user\Repozitoriy3>git add .
"git" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

C:\Users\user\Repozitoriy3>git commit -m"Индивидуальное задание 3"
[develop e519f69] Индивидуальное задание 3
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 individual_3.py

C:\Users\user\Repozitoriy3> git push --set-upstream origin develop
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 24 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 276 bytes | 276.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Vishhh123/Repozitoriy3.git
 5bcb3f0..e519f69 develop -> develop
branch 'develop' set up to track 'origin/develop'.
```

Рис. 30 - сохранение, добавление коммита, отправка на Github.

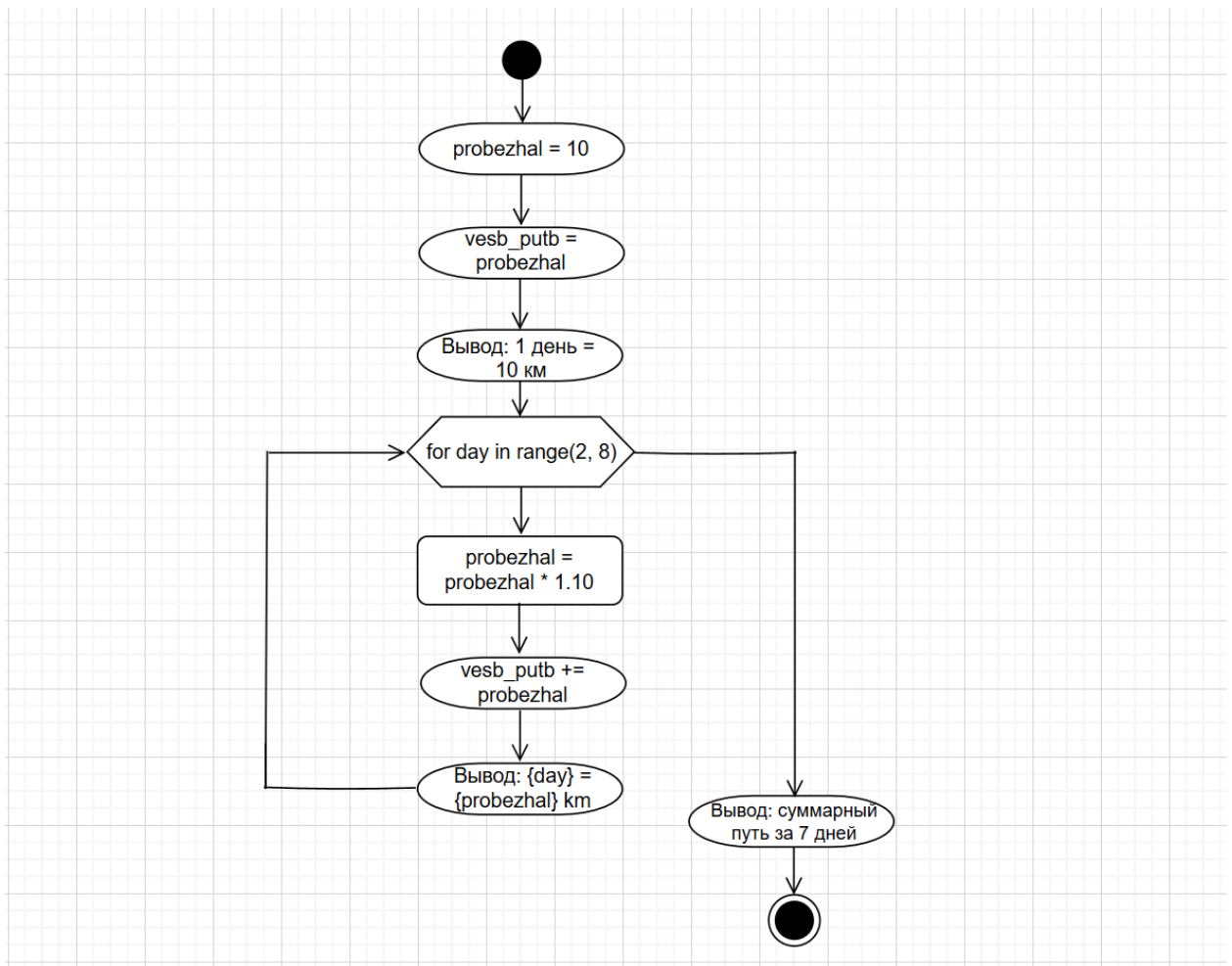


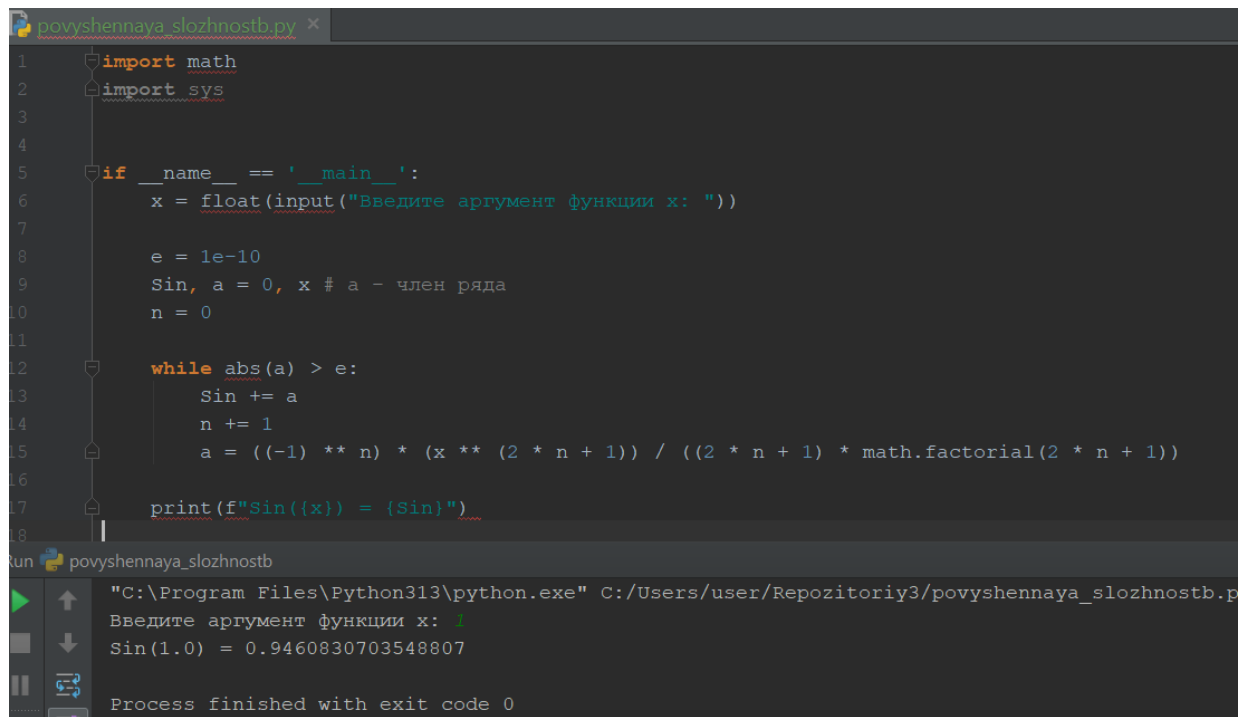
Рис. 31 - UML-диаграмма к инд. заданию 3.

Составить UML-диаграмму деятельности, программу и произвести вычисления вычисление значения специальной функции по ее разложению в ряд с точностью $\epsilon = 10^{-10}$, аргумент функции x вводится с клавиатуры. Номер варианта необходимо получить у преподавателя.

1. Интегральный синус:

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

Рис. 32 - задание повышенной сложности.



```
1 import math
2 import sys
3
4
5 if __name__ == '__main__':
6     x = float(input("Введите аргумент функции x: "))
7
8     e = 1e-10
9     Sin, a = 0, x # a - член ряда
10    n = 0
11
12    while abs(a) > e:
13        Sin += a
14        n += 1
15        a = ((-1) ** n) * (x ** (2 * n + 1)) / ((2 * n + 1) * math.factorial(2 * n + 1))
16
17    print(f"Sin({x}) = {Sin}")
18
```

Run povyshennaya_slozhnostb

"C:\Program Files\Python313\python.exe" C:/Users/user/Repozitoriy3/povyshennaya_slozhnostb.py

Введите аргумент функции x: 1

Sin(1.0) = 0.9460830703548807

Process finished with exit code 0

Рис. 33 - код вычисления интегрального синуса.

```
C:\Users\user\Repozitoriy3>git add .

C:\Users\user\Repozitoriy3>git commit -m"Задание повышенной сложности"
[develop 0c9e5ce] Задание повышенной сложности
 2 files changed, 32 insertions(+)
 create mode 100644 povyshennaya_slozhnostb.py

C:\Users\user\Repozitoriy3>git push --set-upstream origin develop
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 24 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 842 bytes | 842.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Vishhhh123/Repozitoriy3.git
 e519f69..0c9e5ce develop -> develop
branch 'develop' set up to track 'origin/develop'.
```

Рис. 34 - сохранение, добавление коммита, отправка на Github.

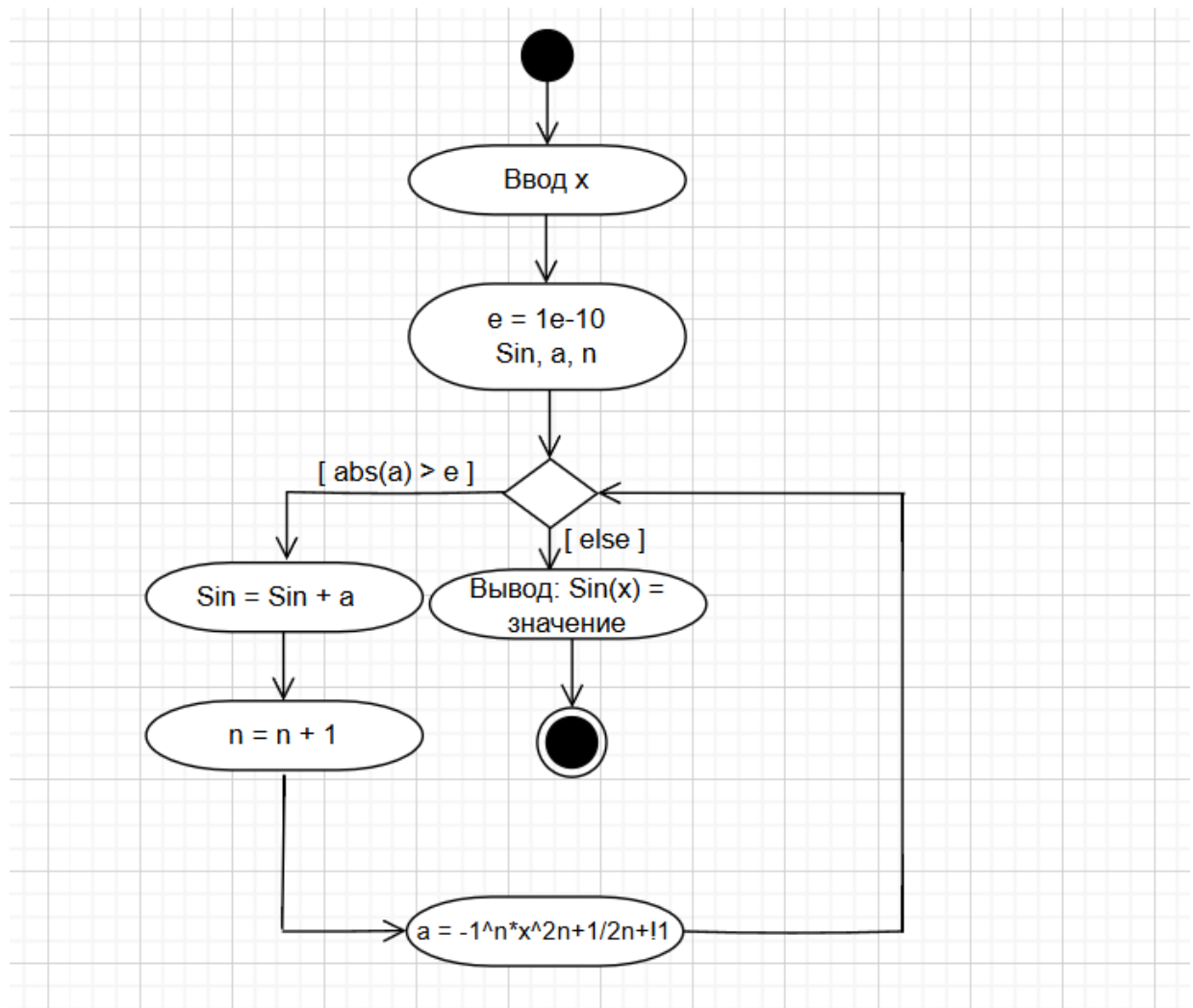


Рис. 35 - UML-диаграмма к заданию повышенной сложности.

база: основная

← сравнить: развить

✓ Можно объединить. Эти ветки можно объединить автоматически.

Разработка #1

Описание недоступно

Просмотреть запрос на вытягивание

8 коммиты

15 файлы изменены

1 автор

Опубликовано 2 октября 2025 г.

Код для функции к заданию 1	a66886e	<>
Фёдорова_Дарья прислужная 5 часов назад		
Код к примеру 2	baddb3b	<>
Фёдорова_Дарья прислужная 4 часа назад		
Код к примеру 3	451c056	<>
Фёдорова_Дарья прислужная 4 часа назад		
Код к заданию 4	4659889	<>
Фёдорова_Дарья прислужная 4 часа назад		
Код к примеру 5	56a0361	<>
Фёдорова_Дарья прислужная 4 часа назад		
Индивидуальное задание 2	5bcb3f0	<>
Фёдорова_Дарья прислужная 1 час назад		
Индивидуальное задание 3	e519f69	<>
Фёдорова_Дарья прислужная 1 час назад		
Задание повышенной сложности	0c9e5ce	<>
Фёдорова_Дарья прислужная 2 минуты назад		

Рис. 36 - слияние ветки разработки с главной.

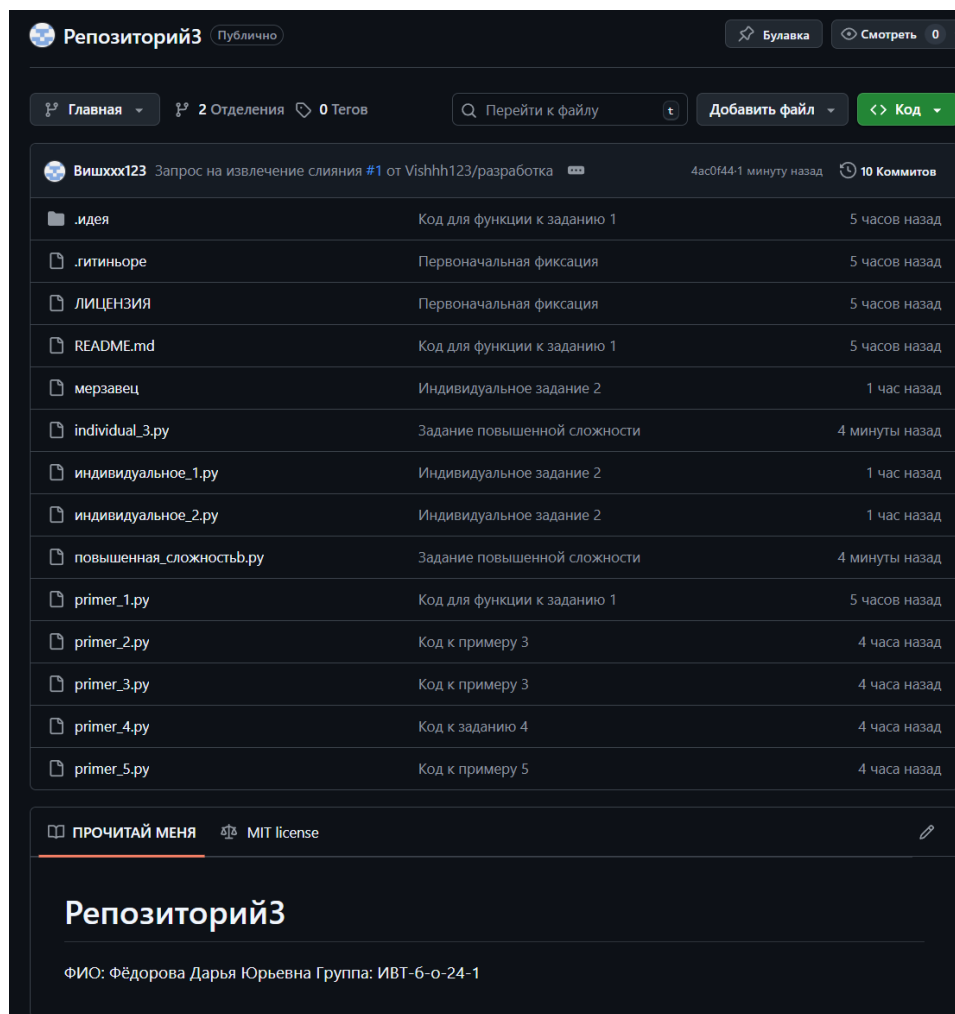


Рис. 37 - главная ветка.

Ответы на контрольные вопросы:

1. Диаграммы деятельности UML используются для моделирования бизнес-процессов, рабочих потоков и алгоритмов выполнения операций в системе.
2. Состояние действия - элементарный шаг в деятельности; состояние деятельности - группа связанных действий, объединенных в подпроцесс.
3. Для переходов используются стрелки, для ветвлений - ромбовидные узлы (decision nodes), для параллельного выполнения - разделительные полосы (fork/join nodes).
4. Алгоритм разветвляющейся структуры содержит условие, которое изменяет порядок выполнения команд в зависимости от результата проверки.
5. Линейный алгоритм выполняется строго последовательно, разветвляющийся имеет условия и разные пути выполнения.
6. Условный оператор выполняет разные блоки кода в зависимости от истинности условия. Формы в Python: if, if-else, if-elif-else.
7. Операторы сравнения в Python: ==, !=, >, <, >=, <=

8. Простое условие содержит одно логическое выражение: "age >= 18", "temperature < 0"

9. Составное условие объединяет несколько простых условий: "age >= 18 and score > 80", "x > 0 or y < 0"

10. Логические операторы: and (и), or (или), not (не)

11. Да, оператор ветвления может содержать внутри себя другие ветвления (вложенные условия).

12. Алгоритм циклической структуры содержит повторяющиеся действия, выполняемые многократно.

13. В Python: цикл for (для известного числа итераций), цикл while (пока условие истинно)

14. Функция range() генерирует последовательность чисел. Синтаксис: range(start, stop, step)

15. range(15, -1, -2)

16. Да, циклы могут быть вложенными (цикл внутри цикла).

17. Бесконечный цикл образуется при while True:. Выход: оператор break или сочетание клавиш Ctrl+C

18. Оператор break нужен для немедленного выхода из цикла.

19. Оператор continue используется в циклах для перехода к следующей итерации, пропуская оставшийся код текущей итерации.

20. stdout - стандартный вывод для обычных сообщений, stderr - вывод для сообщений об ошибках.

21. import sys; sys.stderr.write("Текст ошибки")

22. Функция exit() завершает выполнение программы с указанным кодом выхода.

Вывод: в ходе работы были приобретены навыки программирования разветвляющихся алгоритмов и алгоритмов циклической структуры. Освоены операторы языка Python версии 3.x if, while, for,