

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 №5
дисциплины «Основы программной инженерии»

Выполнил:
Яблоновский Дмитрий Николаевич
2 курс, группа ПИЖ-б-о-22-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка и
сопровождение программного
обеспечения», очная форма обучения

(подпись)

Руководитель практики:
Богданов С.С., ассистент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: Условные операторы и циклы в языке Python.

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.

Порядок выполнения работы

1. Создал репозиторий GitHub.

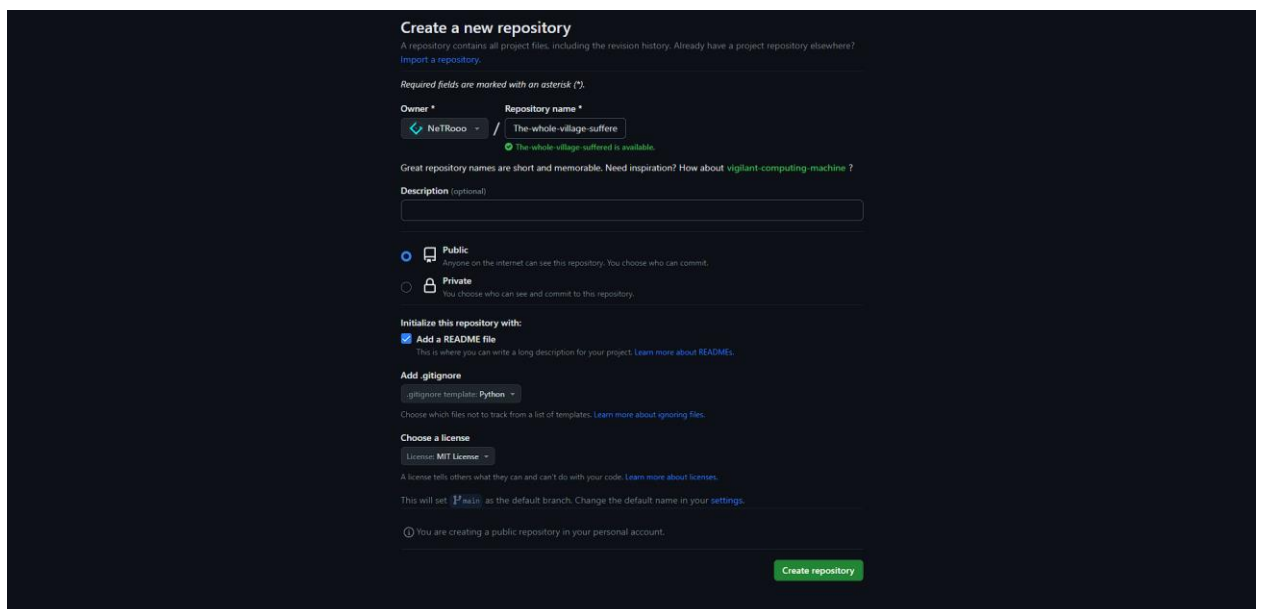


Рисунок 1- Создание репозитория

2. Проработал примеры из лабораторной работы.

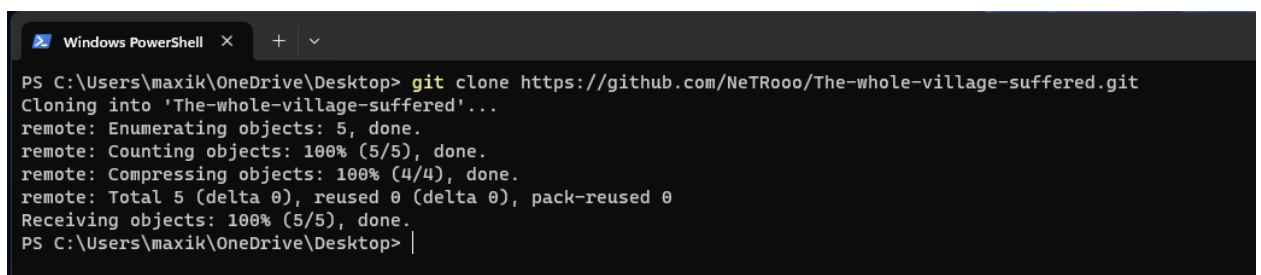


Рисунок 2 – Выполнил клонирование созданного репозитория

```
test1.py U X
test1.py > ...
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 import math
4 if __name__ == '__main__':
5     x = float(input("Value of x? "))
6     if x <= 0:
7         y = 2 * x * x + math.cos(x)
8     elif x < 5:
9         y = x + 1
10    else:
11        y = math.sin(x) - x * x
12    print(f"y = {y}")

ПРОБЛЕМЫ  ВЫХОДНЫЕ ДАННЫЕ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ  ПОРТЫ
PS C:\Users\maxik\OneDrive\Desktop\The-whole-village-suffered> & C:/Users/maxik/AppData/Local/Programs/Python/Python39/python.exe c:/Users/maxik/OneDrive/Desktop/The-whole-village-suffered/test1.py
Value of x? 25
y = -625.1323517500978
PS C:\Users\maxik\OneDrive\Desktop\The-whole-village-suffered> █
```

Рисунок 3 – Нахождение значения функции и вывод программы
(задание №1)

```
test1.py U  test2.py U X
test2.py > ...
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 import sys
4 if __name__ == '__main__':
5     n = int(input("Введите номер месяца: "))
6     if n == 1 or n == 2 or n == 12:
7         print("Зима")
8     elif n == 3 or n == 4 or n == 5:
9         print("Весна")
10    elif n == 6 or n == 7 or n == 8:
11        print("Лето")
12    elif n == 9 or n == 10 or n == 11:
13        print("Осень")
14    else:
15        print("Ошибка!", file=sys.stderr)
16    exit(1)

ПРОБЛЕМЫ  ВЫХОДНЫЕ ДАННЫЕ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ  ПОРТЫ
PS C:\Users\maxik\OneDrive\Desktop\The-whole-village-suffered> & C:/Users/maxik/AppData/Local/Programs/Python/Python39/python.exe c:/Users/maxik/OneDrive/Desktop/The-whole-village-suffered/test2.py
Введите номер месяца: 6
Лето
PS C:\Users\maxik\OneDrive\Desktop\The-whole-village-suffered> & C:/Users/maxik/AppData/Local/Programs/Python/Python39/python.exe c:/Users/maxik/OneDrive/Desktop/The-whole-village-suffered/test2.py
Введите номер месяца: 124
Ошибка!
PS C:\Users\maxik\OneDrive\Desktop\The-whole-village-suffered> █
```

Рисунок 4 – Вывод времени года по номеру месяца (задание №2)

```
test1.py U  test2.py U  test3.py U X
test3.py > ...
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 import math
4 if __name__ == '__main__':
5     n = int(input("Value of n? "))
6     x = float(input("Value of x? "))
7     S = 0
8     for k in range(1, n + 1):
9         a = math.log(k * x) / (k * k)
10        S += a
11    print(f"S = {S}")

ПРОБЛЕМЫ  ВЫХОДНЫЕ ДАННЫЕ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ  ПОРТЫ
PS C:\Users\maxik\OneDrive\Desktop\The-whole-village-suffered> & C:/Users/maxik/AppData/Local/Programs/Python/Python39/python.exe c:/Users/maxik/OneDrive/Desktop/The-whole-village-suffered/test3.py
Value of n? 10
Value of x? 25
S = 5.6070125280518965
PS C:\Users\maxik\OneDrive\Desktop\The-whole-village-suffered> █
```

Рисунок 5 – Вычисление конечной суммы и вывод работы программы
(задание №3)

```
test4.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  import math
4  import sys
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     x, eps = 1, 1e-10
11     while True:
12         xp = x
13         x = (x + a / x) / 2
14         if math.fabs(x - xp) < eps:
15             break
16     print(f"x = {x}\nx = {math.sqrt(a)}")
```

ПРОБЛЕМЫ Выходные данные КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

PS C:\Users\maxik\OneDrive\Desktop\The-whole-village-suffered> & C:/Users/maxik/AppData/Local/Programs/Python/Python39/python.exe c:/Users/maxik/OneDrive/Desktop/The-whole-village-suffered/test4.py

Value of a? 10
x = 3.162277660168379
x = 3.1622776601683795
PS C:\Users\maxik\OneDrive\Desktop\The-whole-village-suffered>

Рисунок 6 – Нахождение квадратного корня числа и его сравнение с результатом метода sqrt() стандартной библиотеки Python (задание №4)

```
test5.py > [E] EULER
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  import math
4  import sys
5  # Константа Эйлера.
6  EULER = 0.5772156649015329
7  # Точность вычислений.
8  EPS = 1e-10
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     a = x
15     S, k = a, 1
16     # Найти сумму членов ряда.
17     while math.fabs(a) > EPS:
18         a *= x * k / (k + 1) ** 2
19         S += a
20         k += 1
21     # Вывести значение функции.
22     print(f"Ei({x}) = {EULER + math.log(math.fabs(x)) + S}")
```

ПРОБЛЕМЫ Выходные данные КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

PS C:\Users\maxik\OneDrive\Desktop\The-whole-village-suffered> & C:/Users/maxik/AppData/Local/Programs/Python/Python39/python.exe c:/Users/maxik/OneDrive/Desktop/The-whole-village-suffered/test5.py

Value of x? 20
Ei(20.0) = 25615652.86405659
PS C:\Users\maxik\OneDrive\Desktop\The-whole-village-suffered>

Рисунок 7 – Вычисление значения специальной (интегральной показательной) функции (задание №5)

```

individual1.py U X
individual1.py > ...
1 def calculate_earnings(kg_collected):
2     if kg_collected <= 50:
3         earnings = kg_collected * 0.30 # 30 копеек за 1 кг
4     elif 50 < kg_collected <= 75:
5         earnings = kg_collected * 0.50 # 50 копеек за 1 кг
6     elif 75 < kg_collected <= 90:
7         earnings = kg_collected * 0.65 # 65 копеек за 1 кг
8     else:
9         earnings = kg_collected * 0.70 + 20 # 70 копеек за 1 кг + 20 рублей премии
10
11     return earnings
12
13 if __name__ == "__main__":
14     kg_collected = float(input("Введите количество килограммов помидоров, собранных студентом: "))
15     earnings = calculate_earnings(kg_collected)
16     print(f"Заработок студента составил {earnings:.2f} рублей.")
17
ПРОБЛЕМЫ Выходные данные КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ
+ Python
PS C:\Users\maxik\OneDrive\Desktop\The-whole-village-suffered> & C:/Users/maxik/AppData/Local/Programs/Python/Python39/python.exe c:/Users/maxik/OneDrive/Desktop/The-whole-village-suffered/individual1.py
Введите количество килограммов помидоров, собранных студентом: 50
Заработок студента составил 15.00 рублей.
PS C:\Users\maxik\OneDrive\Desktop\The-whole-village-suffered> & C:/Users/maxik/AppData/Local/Programs/Python/Python39/python.exe c:/Users/maxik/OneDrive/Desktop/The-whole-village-suffered/individual1.py
Введите количество килограммов помидоров, собранных студентом: 578
Заработок студента составил 424.60 рублей.
PS C:\Users\maxik\OneDrive\Desktop\The-whole-village-suffered> & C:/Users/maxik/AppData/Local/Programs/Python/Python39/python.exe c:/Users/maxik/OneDrive/Desktop/The-whole-village-suffered/individual1.py
Введите количество килограммов помидоров, собранных студентом: 12
Заработок студента составил 3.60 рублей.
PS C:\Users\maxik\OneDrive\Desktop\The-whole-village-suffered>

```

Рисунок 8 – Результат работы задания №1

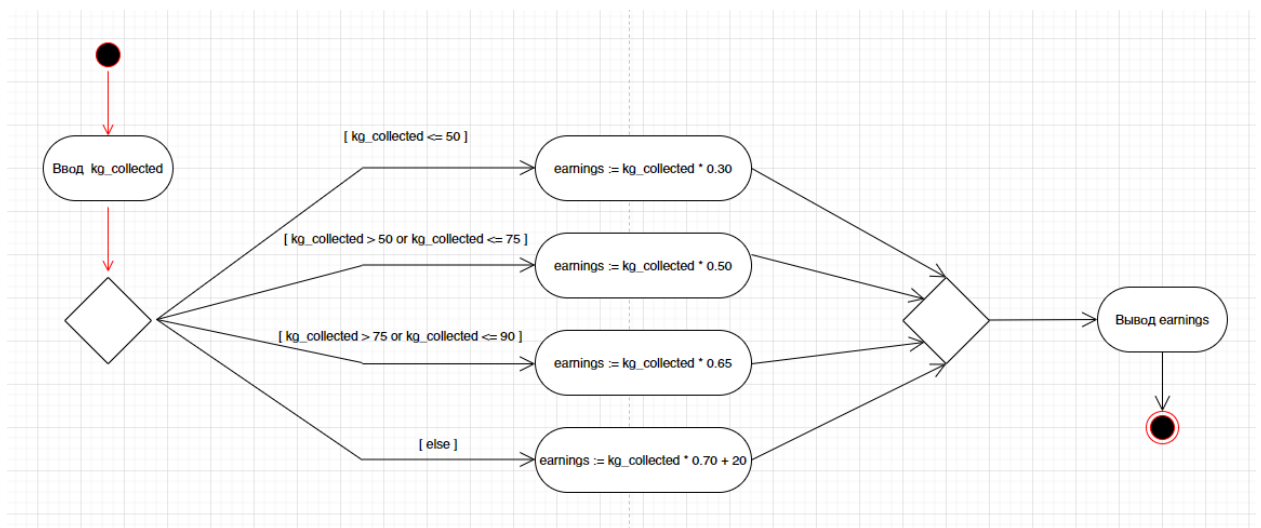


Рисунок 9 – Диаграмма программы индивидуального задания №1

```

individual1.py U individual2.py U X individual3.py U
individual2.py > ...
1 def check():
2     x = int(input("Введите натуральное число x: "))
3     y = int(input("Введите натуральное число y: "))
4
5     if x % y == 0:
6         print(f"{x} делится нацело на {y}.")
7     else:
8         print(f"{x} не делится нацело на {y}.")
9
10 if __name__ == "__main__":
11     check()

```

ПРОБЛЕМЫ Выходные данные КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ + Python

PS C:\Users\maxik\OneDrive\Desktop\The-whole-village-suffered> & C:/Users/maxik/AppData/Local/Programs/Python/Python39/python.exe c:/Users/maxik/OneDrive/Desktop/The-whole-village-suffered/individual2.py
Введите натуральное число x: 4
Введите натуральное число y: 16
4 не делится нацело на 16.

PS C:\Users\maxik\OneDrive\Desktop\The-whole-village-suffered> & C:/Users/maxik/AppData/Local/Programs/Python/Python39/python.exe c:/Users/maxik/OneDrive/Desktop/The-whole-village-suffered/individual2.py
Введите натуральное число x: 16
Введите натуральное число y: 4
16 делится нацело на 4.

PS C:\Users\maxik\OneDrive\Desktop\The-whole-village-suffered>

Рисунок 10 – Результат работы задания №2

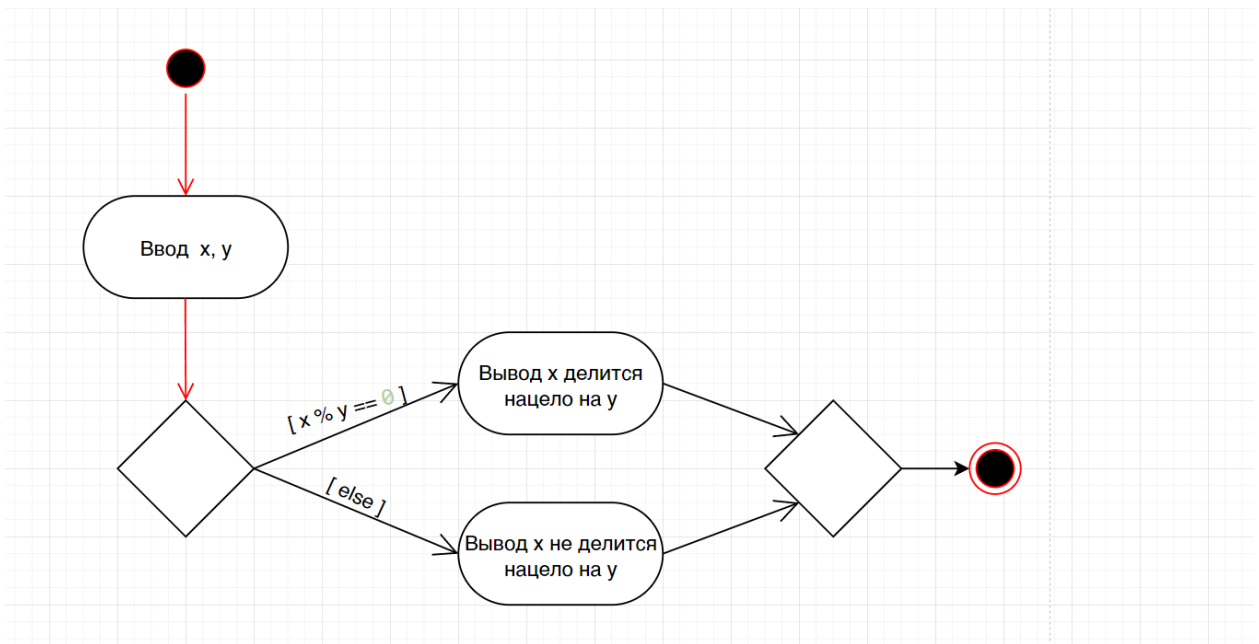


Рисунок 11 – Диаграмма программы индивидуального задания №2

```

1 def natural_divisors(n):
2     divisors = []
3     for i in range(1, n + 1):
4         if n % i == 0:
5             divisors.append(i)
6     return divisors
7
8 if __name__ == "__main__":
9     n = int(input("Enter a natural number: "))
10    divisors = natural_divisors(n)
11    print("Natural divisors of", n, "are:", divisors)
  
```

Terminal output:

```

PS C:\Users\maxik\OneDrive\Desktop\The-whole-village-suffered> & C:/Users/maxik/AppData/Local/Programs/Python/Python39/python.exe c:/Users/maxik/OneDrive/Desktop/The-whole-village-suffered/individual3.py
Enter a natural number: 15
Natural divisors of 15 are: [1, 3, 5, 15]
PS C:\Users\maxik\OneDrive\Desktop\The-whole-village-suffered> & C:/Users/maxik/AppData/Local/Programs/Python/Python39/python.exe c:/Users/maxik/OneDrive/Desktop/The-whole-village-suffered/individual3.py
Enter a natural number: 95
Natural divisors of 95 are: [1, 5, 19, 95]
PS C:\Users\maxik\OneDrive\Desktop\The-whole-village-suffered> & C:/Users/maxik/AppData/Local/Programs/Python/Python39/python.exe c:/Users/maxik/OneDrive/Desktop/The-whole-village-suffered/individual3.py
Enter a natural number: 47
Natural divisors of 47 are: [1, 47]
PS C:\Users\maxik\OneDrive\Desktop\The-whole-village-suffered>
  
```

Рисунок 12 – Результат работы задания №3

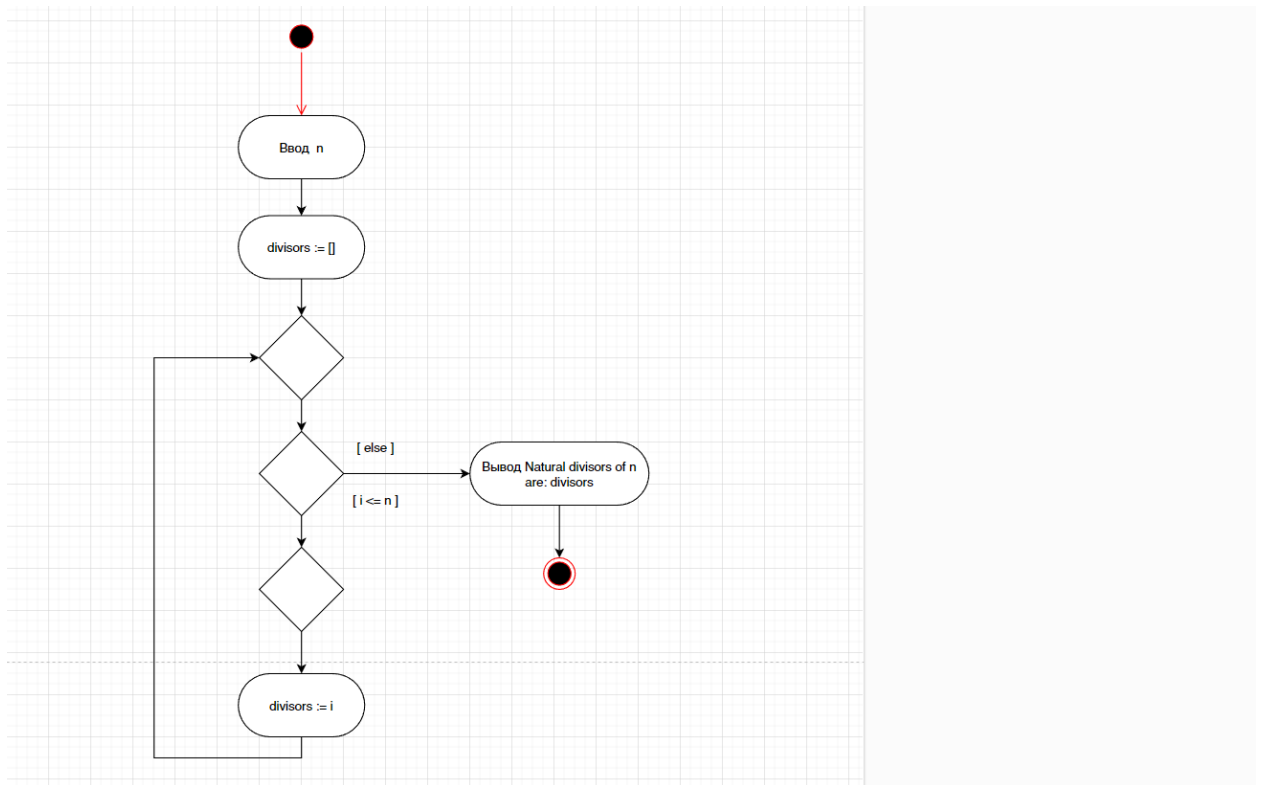


Рисунок 13 – Диаграмма программы индивидуального задания №3

```

PS C:\Users\maxik\OneDrive\Desktop\The-whole-village-suffered> git commit -m "last commit"
[dev 03bb441] last commit
9 files changed, 130 insertions(+)
create mode 100644 individual1.py
create mode 100644 individual2.py
create mode 100644 individual3.py
create mode 100644 individual4.py
create mode 100644 test1.py
create mode 100644 test2.py
create mode 100644 test3.py
create mode 100644 test4.py
create mode 100644 test5.py
PS C:\Users\maxik\OneDrive\Desktop\The-whole-village-suffered> git push
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 12 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (11/11), 2.84 KiB | 1.42 MiB/s, done.
Total 11 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/NeTRooo/The-whole-village-suffered.git
5061c3f..03bb441 dev -> dev
PS C:\Users\maxik\OneDrive\Desktop\The-whole-village-suffered> |
  
```

Рисунок 12 – Результат работы задания №3

Ответы на контрольные вопросы

1. Как выполнить историю коммитов в Git? Какие существуют дополнительные опции для просмотра истории коммитов?

С помощью команды `git log`. Существуют множество дополнительных опций вот некоторые из них:

- `-p` или `-patch` – показывает разницу, внесенную в каждый коммит;
- `--stat` – позволяет увидеть сокращенную статистику;
- `--pretty` – эта опция меняет формат вывода;
- `--since` и `--until` – опции для ограничения вывода по времени

2. Как ограничить вывод при просмотре истории коммитов?

С помощью команды `git log` и его аргументов, например `--since`.

3. Как внести изменения в уже сделанный коммит?

С помощью команды `git commit --amend`.

4. Как отменить индексацию файла в Git?

С помощью команды `git reset HEAD`.

5. Как отменить изменения в файле?

С помощью команды `git checkout -- <file>`.

6. Что такое удаленный репозиторий Git?

Удалённые репозитории представляют собой версии вашего проекта, сохранённые в интернете или ещё где-то в сети.

7. Как выполнить просмотр удаленных репозиториях данного локального репозитория?

С помощью команды `git remote -v`.

8. Как добавить удаленный репозиторий для данного локального репозитория?

С помощью команды `git remote add <shortname> <url>`.

9. Как выполнить отправку/получение изменений с удаленного репозитория?

Получение изменений – `git fetch [remote-name]`, отправка изменений `git push <remote-name> <branch-name>`.

10. Как выполнить просмотр удаленного репозитория?

С помощью команды `git remote -v`.

11. Каково назначение тэгов Git?

Тэги Git - это ссылки на определенные коммиты в истории разработки. Они используются для пометки определенных версий или моментов в вашем проекте. Тэги обычно используются для обозначения релизов или важных этапов в разработке.

12. Как осуществляется работа с тэгами Git?

Для просмотра тэгов – `git tag`, для создания аннотированного тэга – `git tag -a v1.4 -m "сообщение"`, для отправки тэга на удаленный сервер – `git push origin <tagname>`, для удаления тэгов – `git tag -d <tagname>`.

13. . Самостоятельно изучите назначение флага `--prune` в командах `git fetch` и `git push` . Каково назначение этого флага?

Флаг `--prune` в командах `git fetch` и `git push` используется для удаления удаленных веток или тэгов, которые больше не существуют на удаленном репозитории. Если выполнить команду `git fetch --prune`, Git удалит локальные ссылки на удаленные ветки и тэги, которые были удалены на удаленном репозитории. Если выполнить команду `git push --prune`, Git удалит удаленные ветки и тэги на удаленном репозитории, которых больше нет в вашем локальном репозитории.