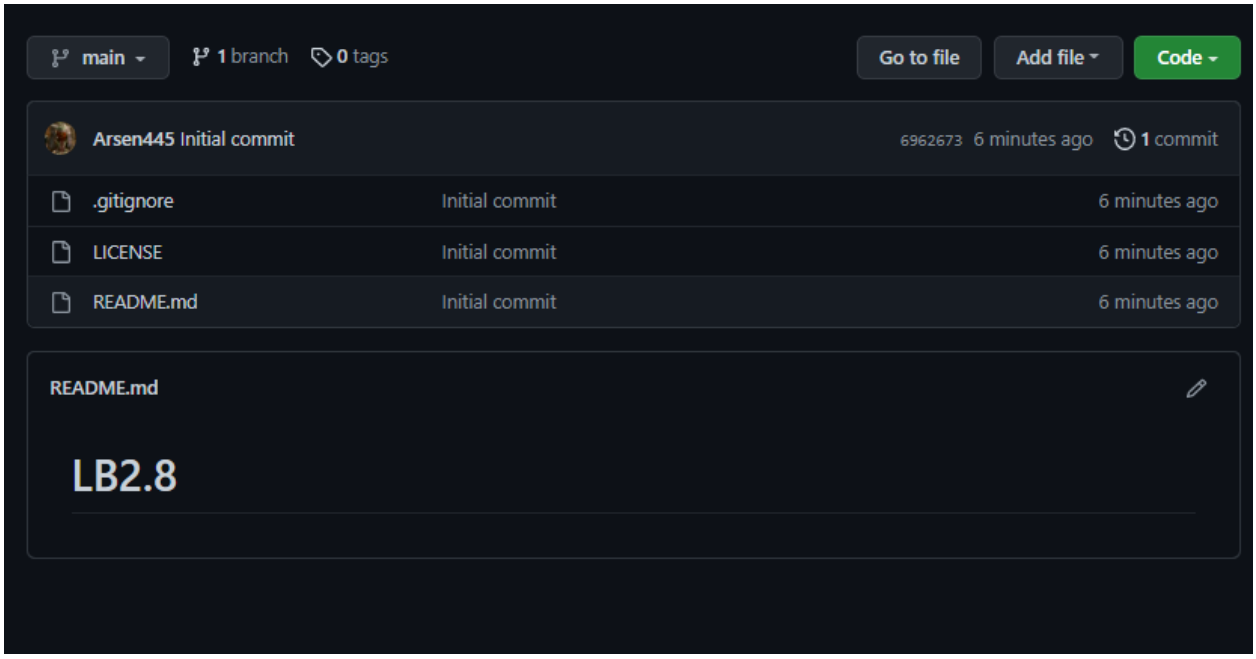
Лабораторная работа №2

Выполнил Эсеналиев Арсен

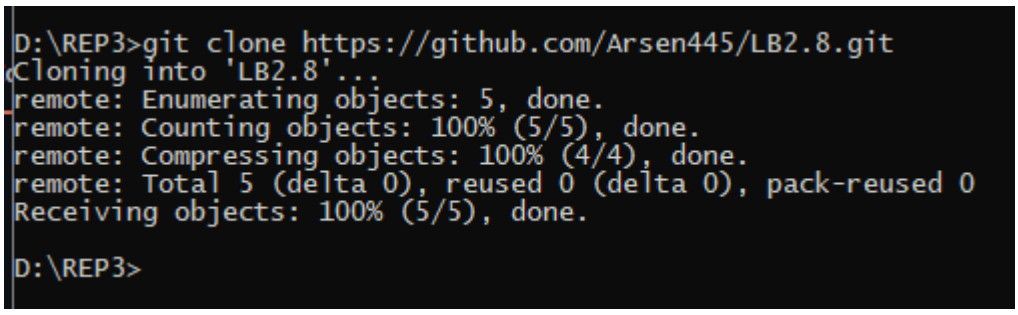
ИВТ-б-о-21-1

Цель: приобретение навыков по работе с функциями при написании программ с помощью языка программирования Python версии 3.x.

1. Создал общедоступный репозиторий на GitHub с MIT



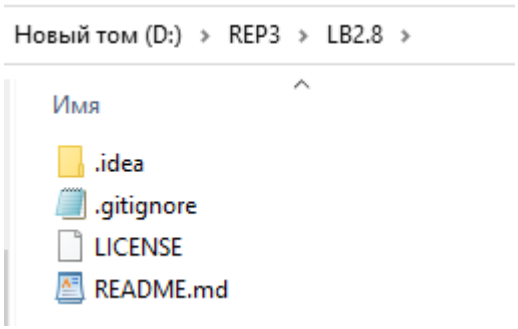
2. Выполнил клонирование созданного репозитория.



3. Организовал свой репозиторий в соответствие с моделью ветвления git-flow. (Перешел с главной main на develop)

```
D:\REP3\LB2.8> 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? [D:/REP3/LB2.8/.git/hooks]
```

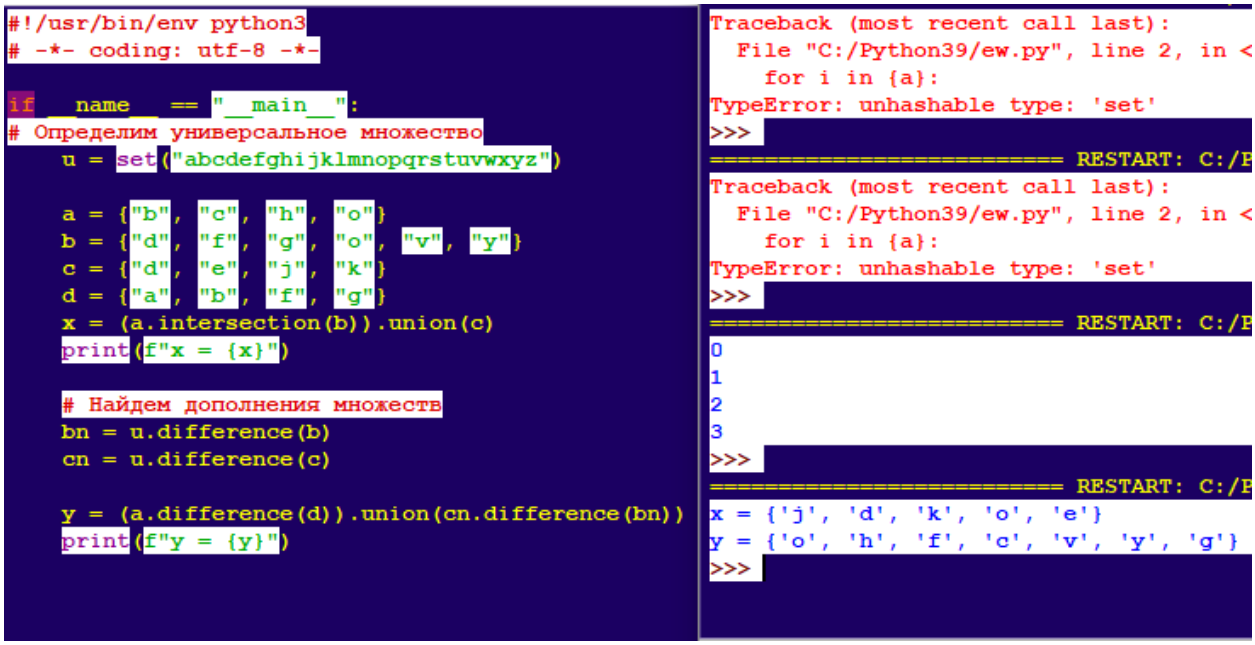
4. Создал проект PyCharm в папке репозитория.



5. Дополнил файл .gitignore необходимыми правилами для работы с IDE PyCharm.

```
# Created by https://www.toptal.com/developers/gitignore/api/python,pycharm
# Edit at https://www.toptal.com/developers/gitignore?templates=python,pycharm

### PyCharm ###
# Covers JetBrains IDEs: IntelliJ, RubyMine, PhpStorm, AppCode, PyCharm, CLion, Android Studio
# Reference: https://intellij-support.jetbrains.com/hc/en-us/articles/206544839

# User-specific stuff
.idea/**/workspace.xml
.idea/**/tasks.xml
.idea/**/usage.statistics.xml
.idea/**/dictionaries
.idea/**/shelf

# AWS User-specific
.idea/**/aws.xml

# Generated files
.idea/**/contentModel.xml

# Sensitive or high-churn files
.idea/**/dataSources/
```

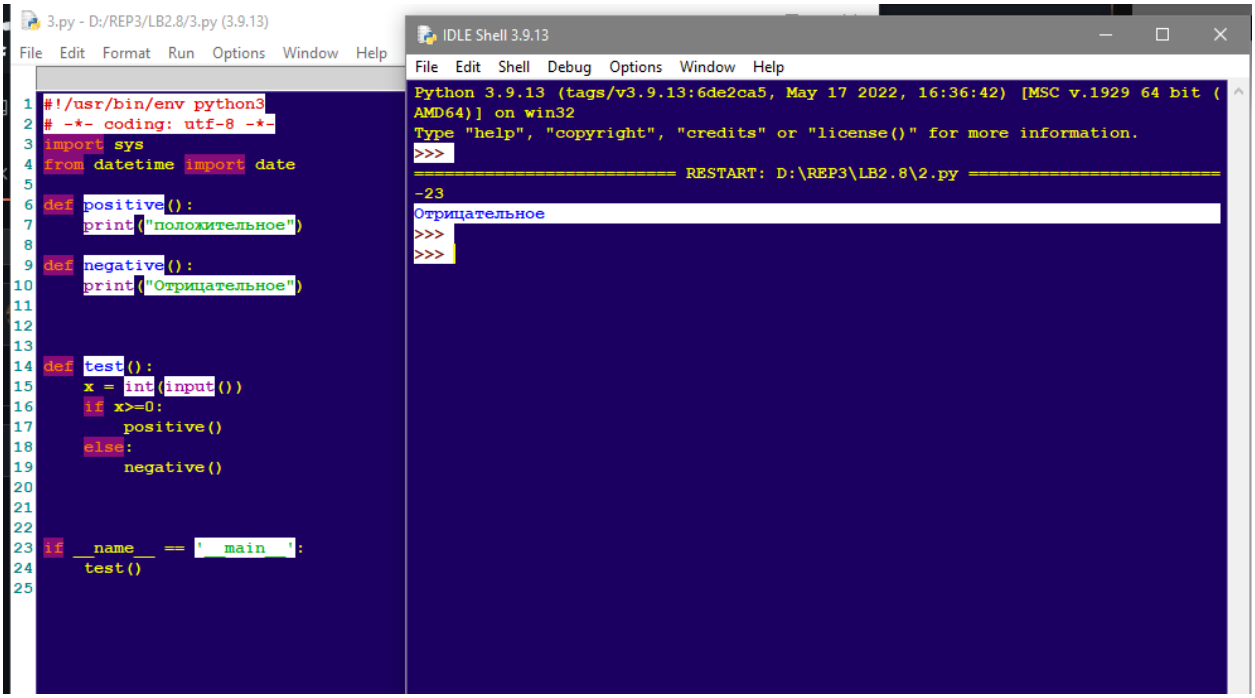
6. Проработал пример лабораторной работы.

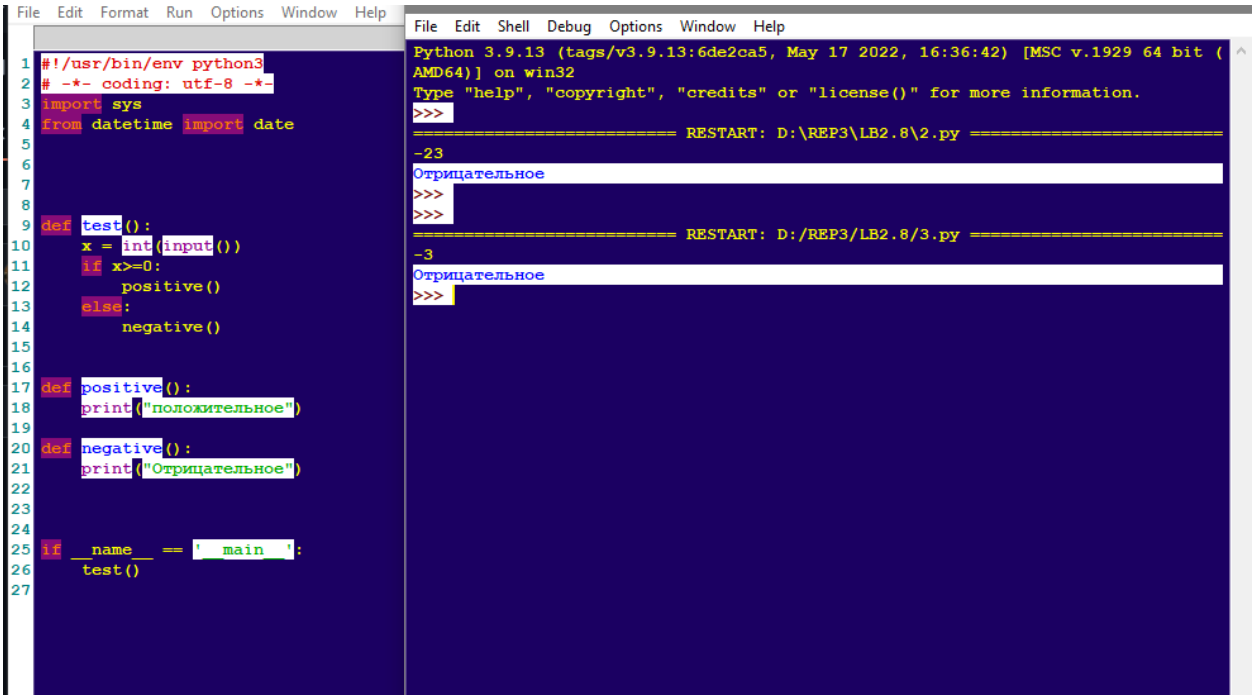


```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == "__main__":
# Определим универсальное множество
    u = set("abcdefghijklmnopqrstuvwxyz")

    a = {"b", "c", "h", "o"}
    b = {"d", "f", "g", "o", "v", "y"}
    c = {"d", "e", "j", "k"}
    d = {"a", "b", "f", "g"}
    x = (a.intersection(b)).union(c)
    print(f"x = {x}")

    # Найдем дополнения множеств
    bn = u.difference(b)
    cn = u.difference(c)

    y = (a.difference(d)).union(cn.difference(bn))
    print(f"y = {y}")
```

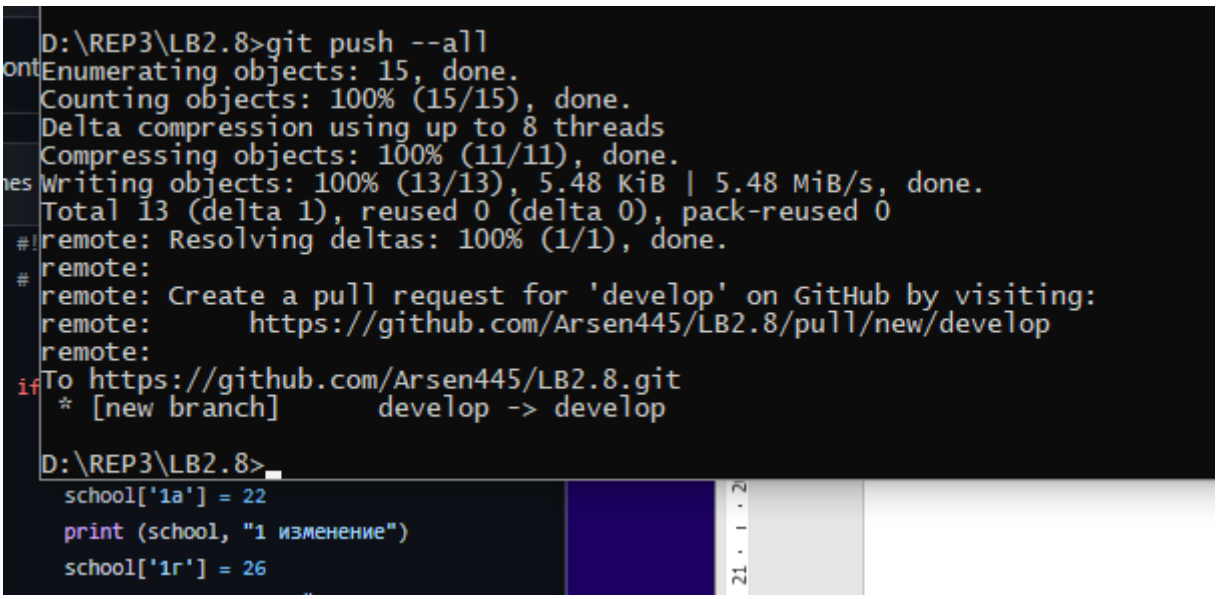
```
Traceback (most recent call last):
  File "C:/Python39/ew.py", line 2, in <
    for i in {a}:
TypeError: unhashable type: 'set'
>>>
========== RESTART: C:/P
Traceback (most recent call last):
  File "C:/Python39/ew.py", line 2, in <
    for i in {a}:
TypeError: unhashable type: 'set'
>>>
========== RESTART: C:/P
0
1
2
3
>>>
========== RESTART: C:/P
x = {'j', 'd', 'k', 'o', 'e'}
y = {'o', 'h', 'f', 'c', 'v', 'y', 'g'}
>>>
```

7. Решить следующую задачу: основная ветка программы, не считая заголовков функций, состоит из двух строки кода. Это вызов функции test() и инструкции if __name__ == '__main__' . В ней запрашивается на ввод целое число. Если оно положительное, то вызывается функция positive(), тело которой содержит команду вывода на экран слова "Положительное". Если число отрицательное, то вызывается функция negative(), ее тело содержит выражение вывода на экран слова "Отрицательное". Понятно, что вызов test() должен следовать после определения функций. Однако имеет ли значение порядок определения самих функций? То есть должны ли определения positive() и negative() предшествовать test() или могут следовать после него? Проверьте вашу гипотезу, поменяв объявления функций местами. Попробуйте объяснить результат.





8. Зафиксируйте сделанные изменения в репозитории.(после создания веток не запушил, поэтому не работало)



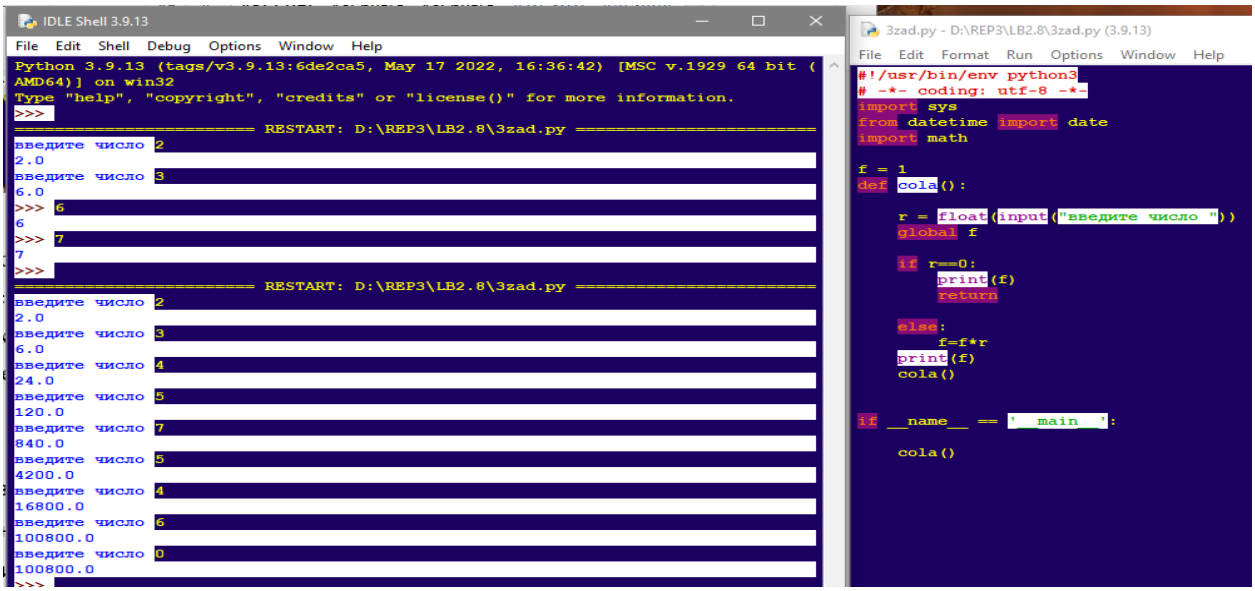
9. Решите следующую задачу: в основной ветке программы вызывается функция cylinder(), которая вычисляет площадь цилиндра. В теле cylinder() определена функция circle(), вычисляющая площадь круга по формуле . В теле cylinder() у пользователя спрашивается, хочет ли он получить только площадь боковой поверхности цилиндра, которая вычисляется по формуле , или полную площадь цилиндра. В последнем случае к площади боковой поверхности цилиндра должен добавляться удвоенный результат вычислений функции circle().

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


cir = 1
def cylinder():

    r = float(input("введите радиус "))
    h = float(input("введите высоту "))

    def circle(r):
        r = 3.14*(r**2)
        global cir
        cir = r


    print("S: Полная-1  Боковая-2")
    d=int(input())

    if d==1:
        print((2*3.14*r)+h+(2*cir))
    if d==2:
        print((2*3.14*r)+h)


if __name__ == '__main__':

    (cylinder())
```

```
==========
введите радиус 2
введите высоту 2
S: Полная-1  Боковая
1
Traceback (most rece
  File "D:\REP3\LB2.
    (cylinder())
  File "D:\REP3\LB2.
    print((2*3.14*r)
NameError: name 'cir
>>>
==========
введите радиус 2
введите высоту 2
S: Полная-1  Боковая
1
14.56
>>>
==========
введите радиус 2
введите высоту 2
S: Полная-1  Боковая
1
16.560000000000002
>>>
```

```
D:\REP3\LB2.8>
Enumerating ob
Counting objec
Delta compress
```

10. Решите следующую задачу: напишите функцию, которая считывает с клавиатуры числа и перемножает их до тех пор, пока не будет введен 0. Функция должна возвращать полученное произведение. Вызовите функцию и выведите на экран результат ее работы.



```
IDLE Shell 3.9.13
File  Edit  Shell  Debug  Options  Window  Help
Python 3.9.13 (tags/v3.9.13:6de2ca5, May 17 2022, 16:36:42) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:\REP3\LB2.8\3zad.py ==========
введите число 2
2.0
введите число 3
6.0
>>> 6
6
>>> 7
7
>>>
========== RESTART: D:\REP3\LB2.8\3zad.py ==========
введите число 2
2.0
введите число 3
6.0
введите число 4
24.0
введите число 5
120.0
введите число 7
840.0
введите число 5
4200.0
введите число 4
16800.0
введите число 6
100800.0
введите число 0
100800.0
>>>
```

```
3zad.py - D:\REP3\LB2.8\3zad.py (3.9.13)
File  Edit  Format  Run  Options  Window  Help
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import sys
from datetime import date
import math

f = 1
def cola():

    r = float(input("введите число "))
    global f

    if r==0:
        print(f)
        return

    else:
        f=f*r
    print(f)
    cola()


if __name__ == '__main__':

    cola()
```

11.Решите следующую задачу: напишите программу, в которой определены следующие четыре функции:

1. Функция get_input() не имеет параметров, запрашивает ввод с клавиатуры и возвращает в основную программу полученную строку.

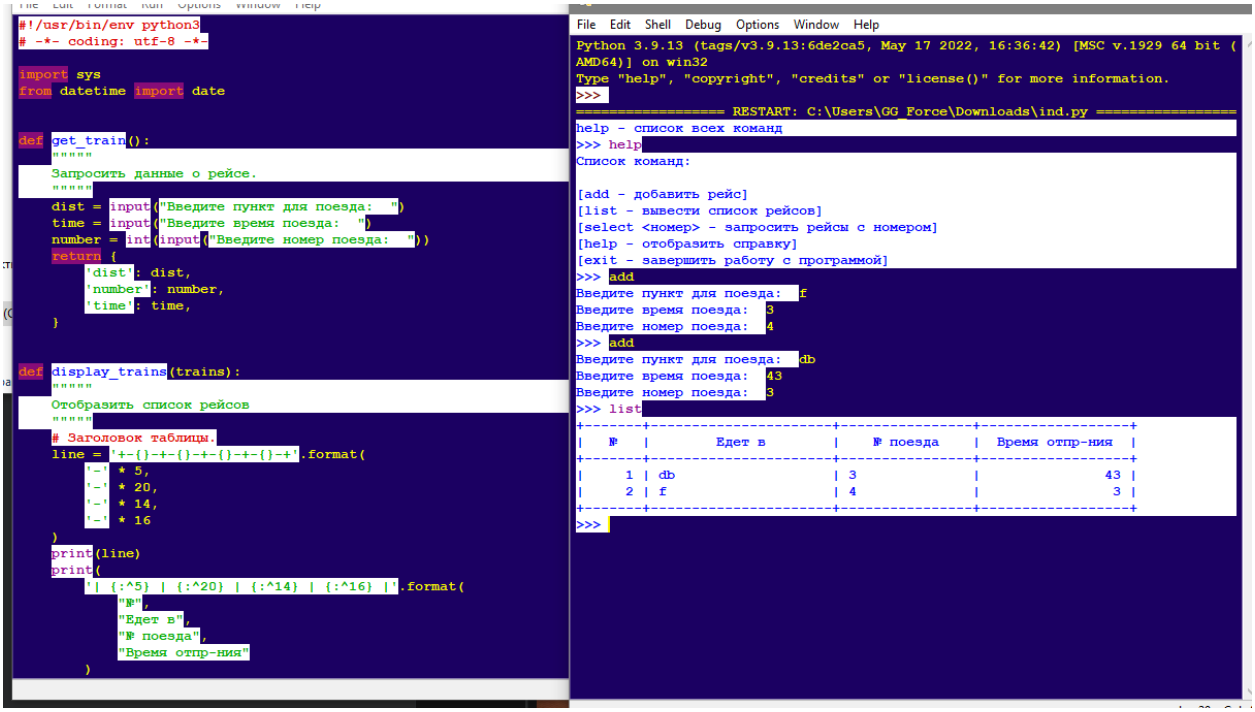
2. Функция test_input() имеет один параметр. В теле она проверяет, можно ли переданное ей значение преобразовать к целому числу. Если можно, возвращает логическое True. Если нельзя – False.

3. Функция str_to_int() имеет один параметр. В теле преобразовывает переданное значение к целочисленному типу. Возвращает полученное число.

4. Функция print_int() имеет один параметр. Она выводит переданное значение на экран и ничего не возвращает.

В основной ветке программы вызовите первую функцию. То, что она вернула, передайте во вторую функцию. Если вторая функция вернула True, то те же данные (из первой функции) передайте в третью функцию, а возвращенное третьей функцией значение – в четвертую.

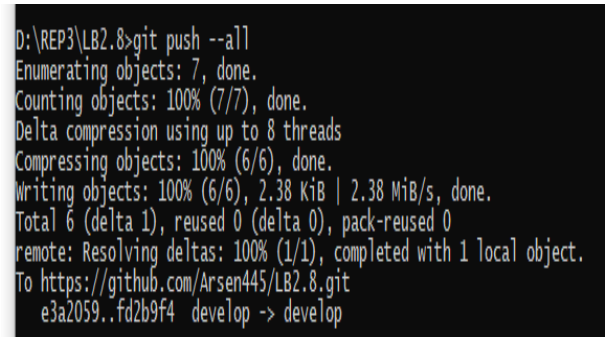
12. Решить индивидуальное задание лабораторной работы 2.6, оформив каждую команду в виде отдельной функции.



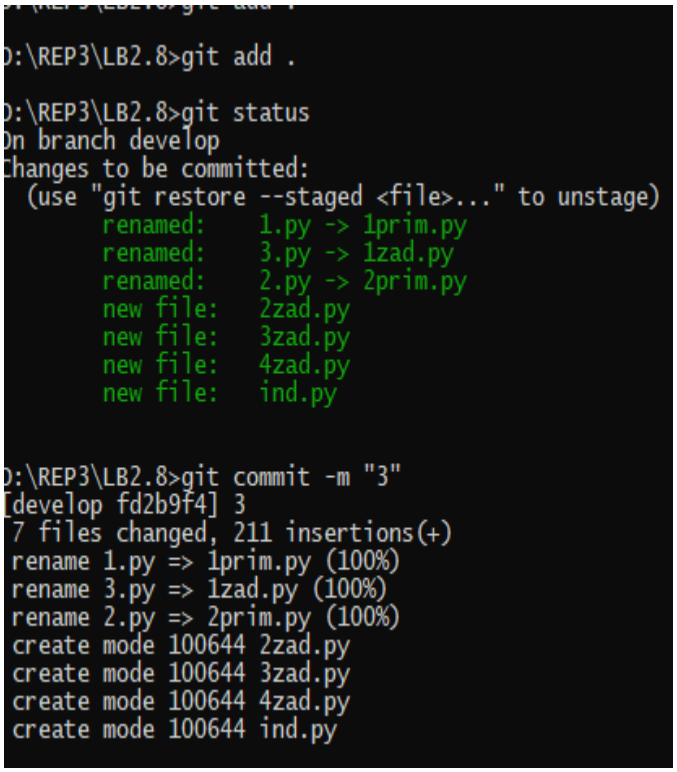
```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys
from datetime import date


def get_train():
    """
    Запросить данные о рейсе.
    """
    dist = input("Введите пункт для поезда:  ")
    time = input("Введите время поезда:  ")
    number = int(input("Введите номер поезда:  "))
    return {
        'dist': dist,
        'number': number,
        'time': time,
    }


def display_trains(trains):
    """
    Отобразить список рейсов
    """
    # Заголовок таблицы.
    line = '+-{}-+-{}-+-{}-+-{}-+'.format(
        '-' * 5,
        '-' * 20,
        '-' * 14,
        '-' * 16
    )
    print(line)
    print(
        '| {:^5} | {:^20} | {:^14} | {:^16} |'.format(
            "№",
            "Едет в",
            "№ поезда",
            "Время отпр-ния"
        )
```

```
File  Edit  Shell  Debug  Options  Window  Help
Python 3.9.13 (tags/v3.9.13:6de2ca5, May 17 2022, 16:36:42) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\GG_Force\Downloads\ind.py ==========
help - список всех команд
>>> help
Список команд:

[add - добавить рейс]
[list - вывести список рейсов]
[select <номер> - запросить рейсы с номером]
[help - отобразить справку]
[exit - завершить работу с программой]
>>> add
Введите пункт для поезда:  f
Введите время поезда:  3
Введите номер поезда:  4
>>> add
Введите пункт для поезда:  db
Введите время поезда:  43
Введите номер поезда:  3
>>> list
+-------+----------------------+---------------+------------------+
|   №   |        Едет в        |    № поезда   |  Время отпр-ния  |
+-------+----------------------+---------------+------------------+
|     1 | db                   | 3             |               43 |
|     2 | f                    | 4             |                3 |
+-------+----------------------+---------------+------------------+
>>>
```

13. Зафиксируйте сделанные изменения в репозитории.

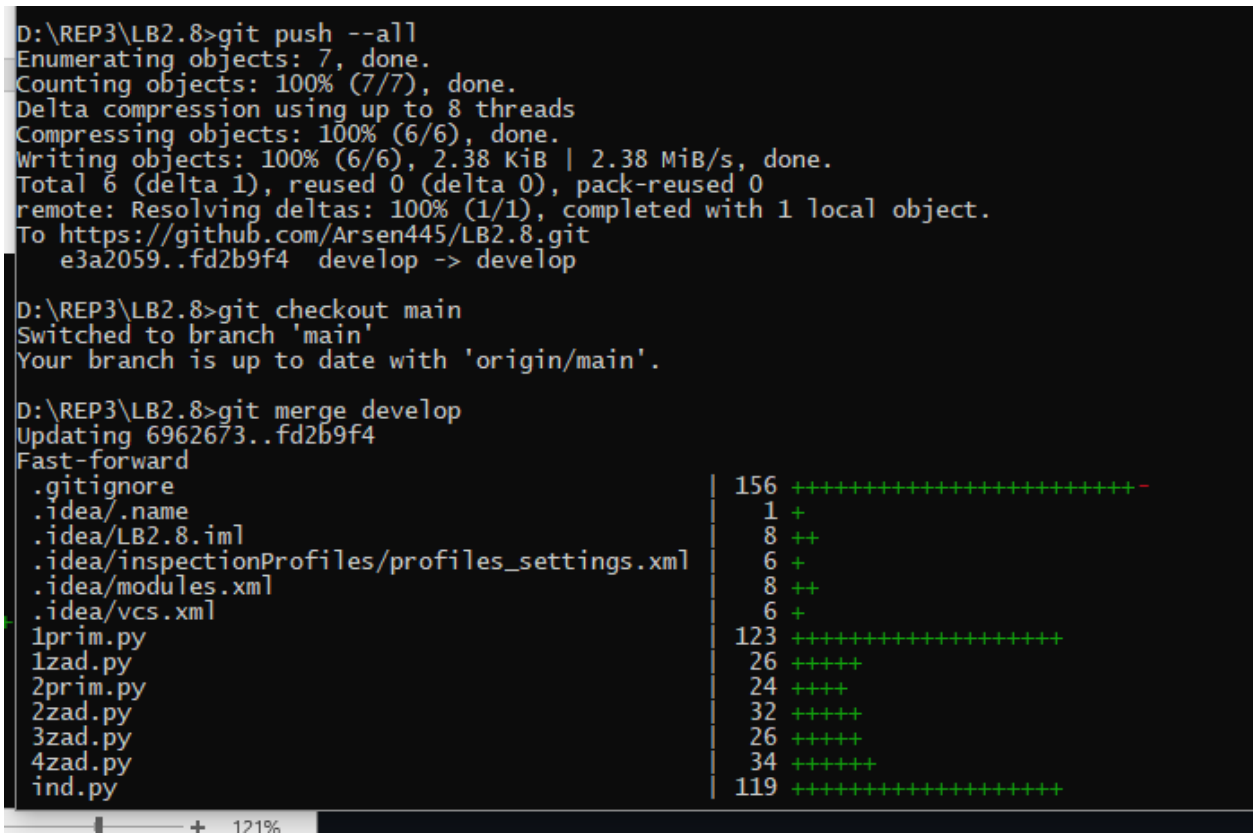


```
D:\REP3\LB2.8>git push --all
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 2.38 KiB | 2.38 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Arsen445/LB2.8.git
   e3a2059..fd2b9f4  develop -> develop
```



```
D:\REP3\LB2.8>git add .

D:\REP3\LB2.8>git status
On branch develop
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        renamed:    1.py -> 1prim.py
        renamed:    3.py -> 1zad.py
        renamed:    2.py -> 2prim.py
        new file:   2zad.py
        new file:   3zad.py
        new file:   4zad.py
        new file:   ind.py


D:\REP3\LB2.8>git commit -m "3"
[develop fd2b9f4] 3
 7 files changed, 211 insertions(+)
 rename 1.py => 1prim.py (100%)
 rename 3.py => 1zad.py (100%)
 rename 2.py => 2prim.py (100%)
 create mode 100644 2zad.py
 create mode 100644 3zad.py
 create mode 100644 4zad.py
 create mode 100644 ind.py
```

14. Выполните слияние ветки для разработки с веткой main/master.



Контрольные вопросы:

1. Каково назначение функций в языке программирования Python? Главной задачей функций в Python, как и в других языках программирования, является сокращение объёма кода и его структуризация. В функции, как правило, выносятся те части кода, которые выполняются в программе многократно.

2. Каково назначение операторов def и return? Оператор def необходим для определения функции. После него идёт название самой функции, передаваемые в функцию параметры и само тело функции. Оператор return служит для возвращения результата выполнения функции в основную программу, где эта функция была вызвана.

3. Каково назначение локальных и глобальных переменных при написании функций Python? Локальные переменные существуют только внутри функции. В другой части программы как-либо вызывать или изменить их невозможно. Глобальные напротив – существуют во всей программе.

4. Как вернуть несколько значений из функции Python? После оператора return необходимо записать все возвращаемые переменные через запятую, а при вызове функции нужно задать необходимое количество переменных. Куда будут возвращены параметры.

5. Какие существуют способы передачи значений в функцию? По ссылке и по значению. 6. Как задать значение аргументов функции по умолчанию? Нужно в скобках передаваемых параметров присвоить им значение.

7. Каково назначение lambda-выражений в языке Python? Lambda-выражения – это небольшие функции, которые вызываются в программе один раз. 8. Как осуществляется документирование кода согласно PEP257? Если пояснение функции содержит одну строку, то достаточно двух кавычек с каждой стороны строки. Пример: ””Пояснение””. Если это многострочное пояснение, то необходимо три кавычки с каждой стороны. Пояснение находится в теле функции, сразу после её объявления.