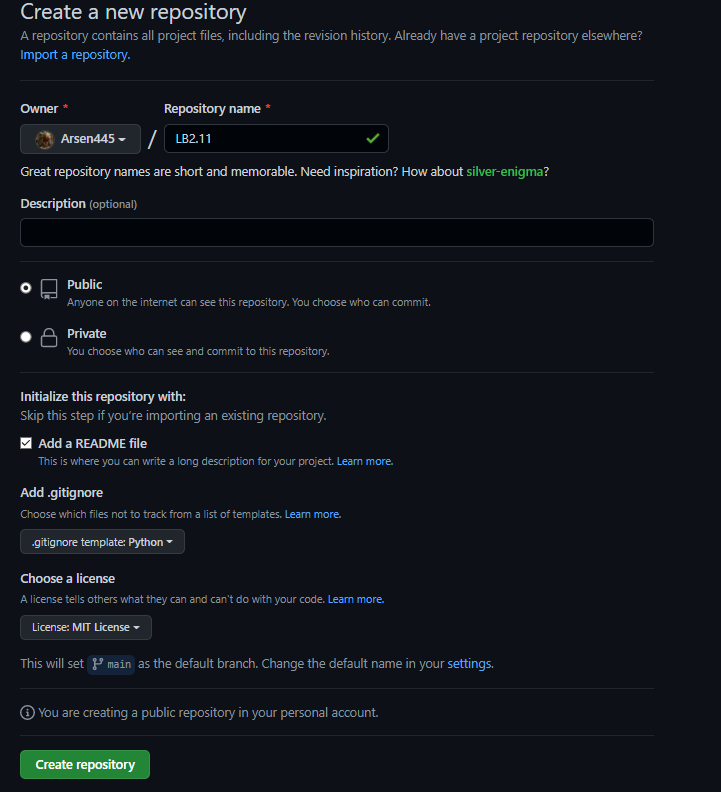
Лабораторная работа №2

Выполнил Эсеналиев Арсен

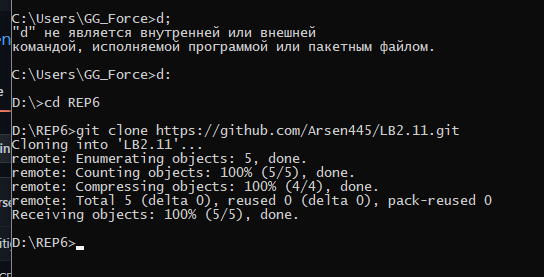
ИВТ-б-о-21-1

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

1. Создал общедоступный репозиторий на GitHub c MIT

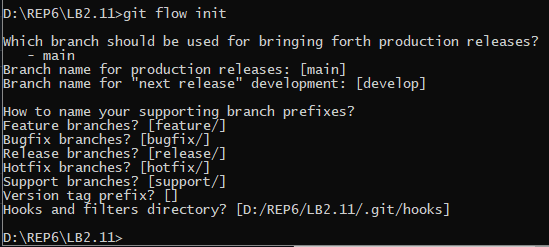


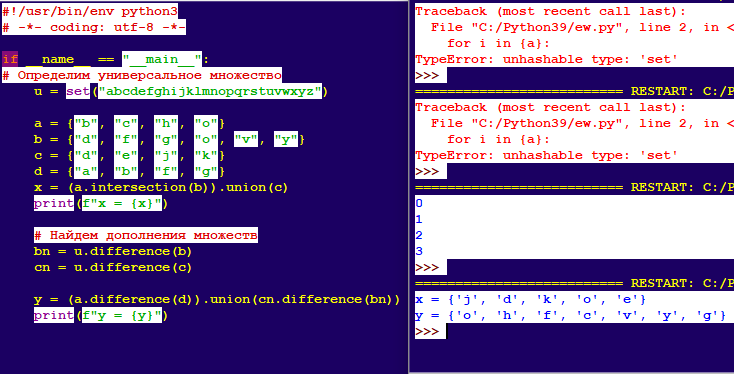
1. Выполнил клонирование созданного репозитория.



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

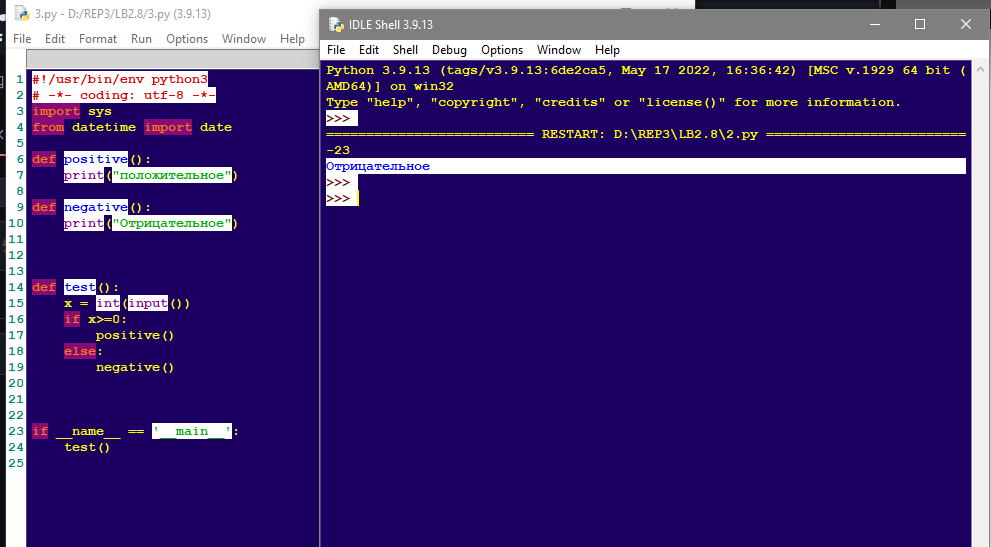


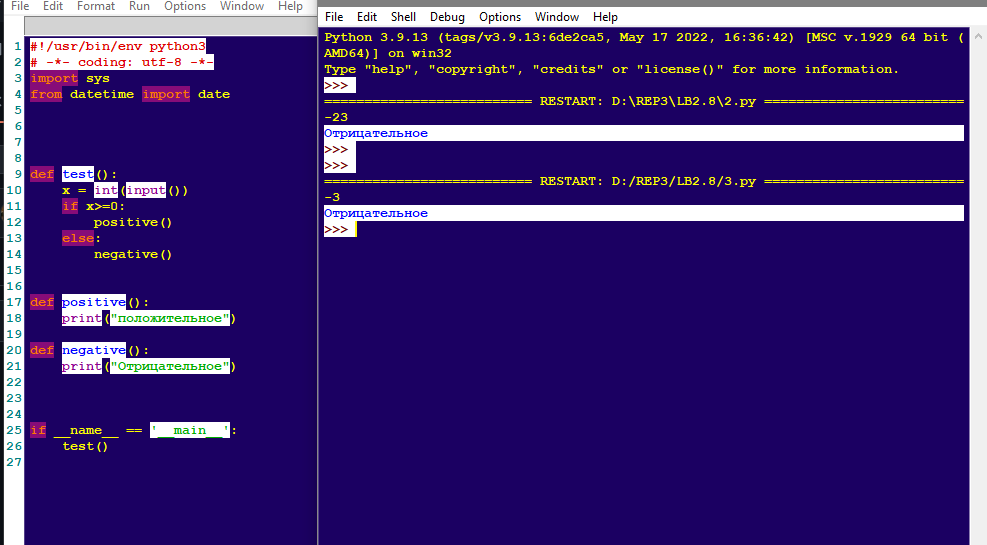
1. Организовал репазиторий в соответствии с моделью ветвления git-flow. 
2. Проработал пример лабораторной работы.



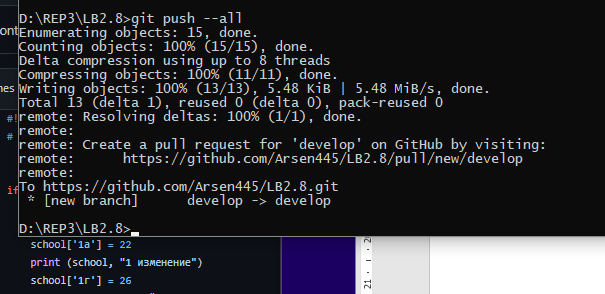
1. Решить следующую задачу: основная ветка программы, не считая заголовков функций, состоит из двух строки кода. Это вызов функции test() и инструкции if \_\_name\_\_ == '\_\_main\_\_' . В ней запрашивается на ввод целое число. Если оно положительное, то вызывается функция positive(), тело которой содержит 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.

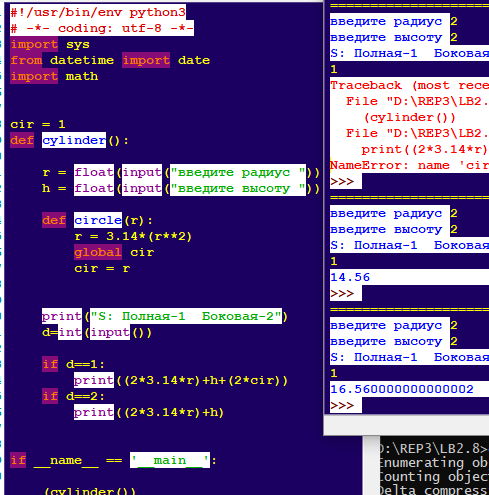




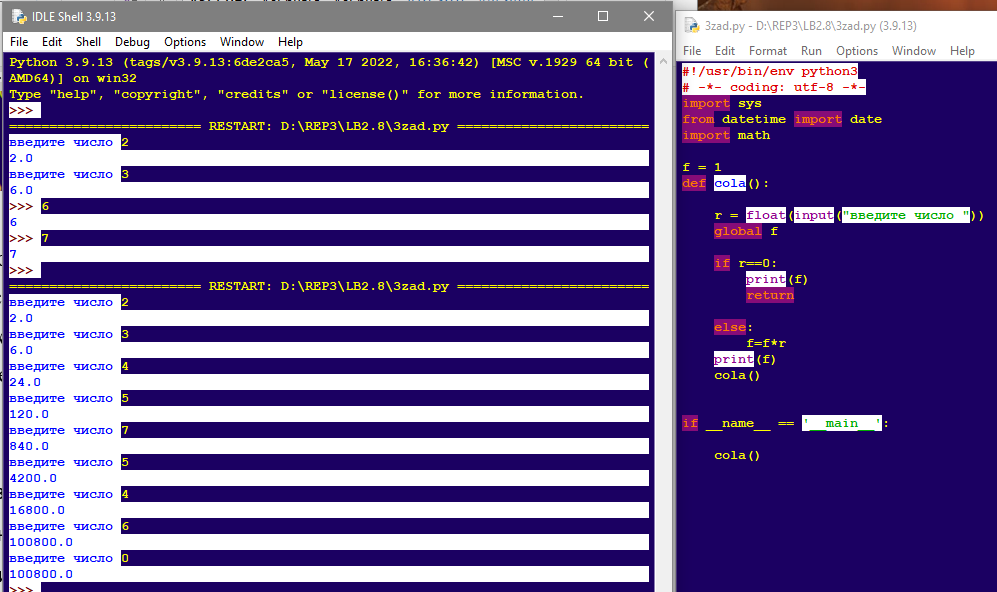
1. Зафиксируйте сделанные изменения в репозитории.(после создания веток не запушил, поэтому не работало)



1. 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 последнеm случае к площади боковой поверхности цилиндра должен добавляться удвоенный результат вычислений функции circle().



1. 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.



1. 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.

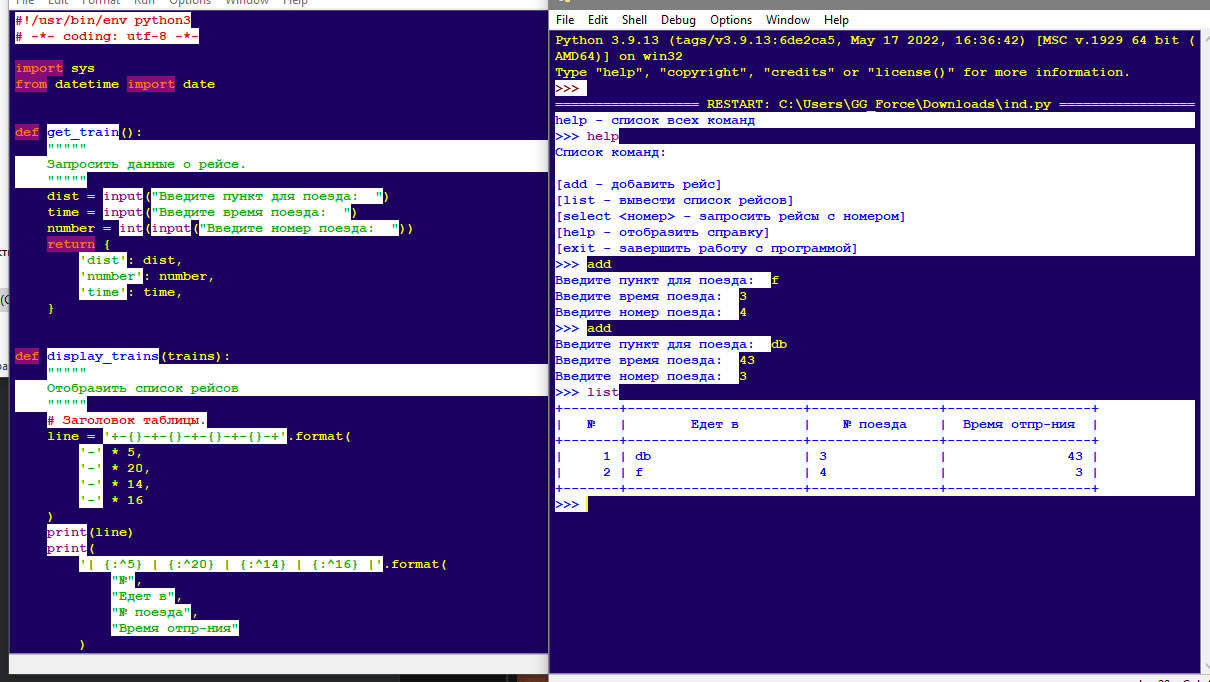
3. Функция str\_to\_int() имеет один параметр. В теле преобразовывает переданное значение к целочисленному типу. Возвращает полученное число.

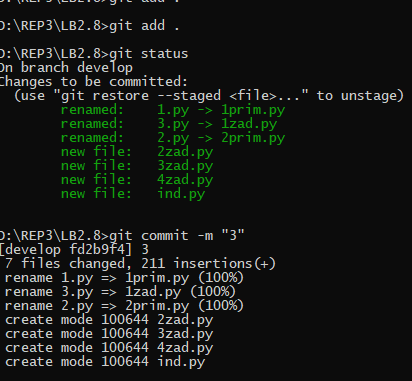
4. Функция print\_int() имеет один параметр. Она выводит переданное значение на экран и ничего не возвращает.

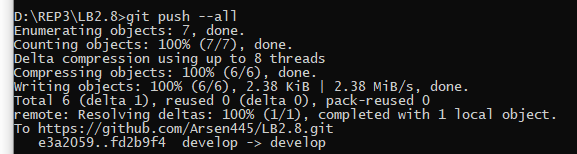
В основной ветке программы вызовите первую функцию. То, что она вернула, передайте во вторую функцию. Если вторая функция вернула True, то те же данные (из первой функции) передайте в третью функцию, а возвращенное третьей функцией значение – в четвертую.



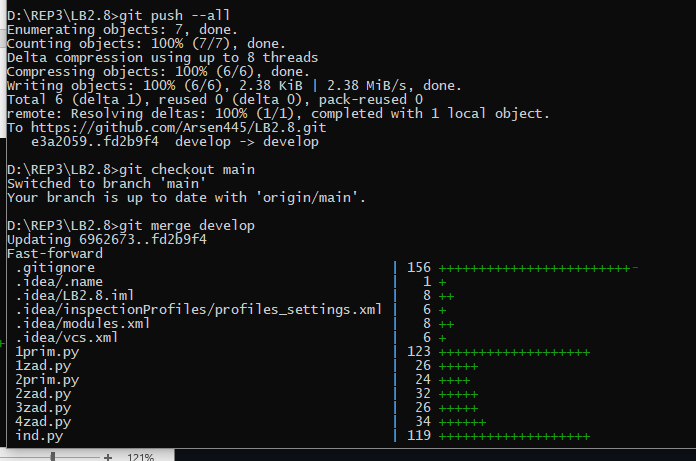
1. Решить индивидуальное задание лабораторной работы 2.6, оформив каждую команду в виде отдельной функции.



1. Зафиксируйте сделанные изменения в репозитории.



1. 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.