

**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 №2.1
Дисциплина: «Основы программной инженерии»
Тема: «Основы ветвления GIT»

Выполнила:
студентка 2 курса
группы Пиж-б-о-21-1
Джолдошова Мээрим
Бекболотовна

Ставрополь 2022

Цель работы: исследование процесса установки и базовых возможностей языка Python версии 3.x.

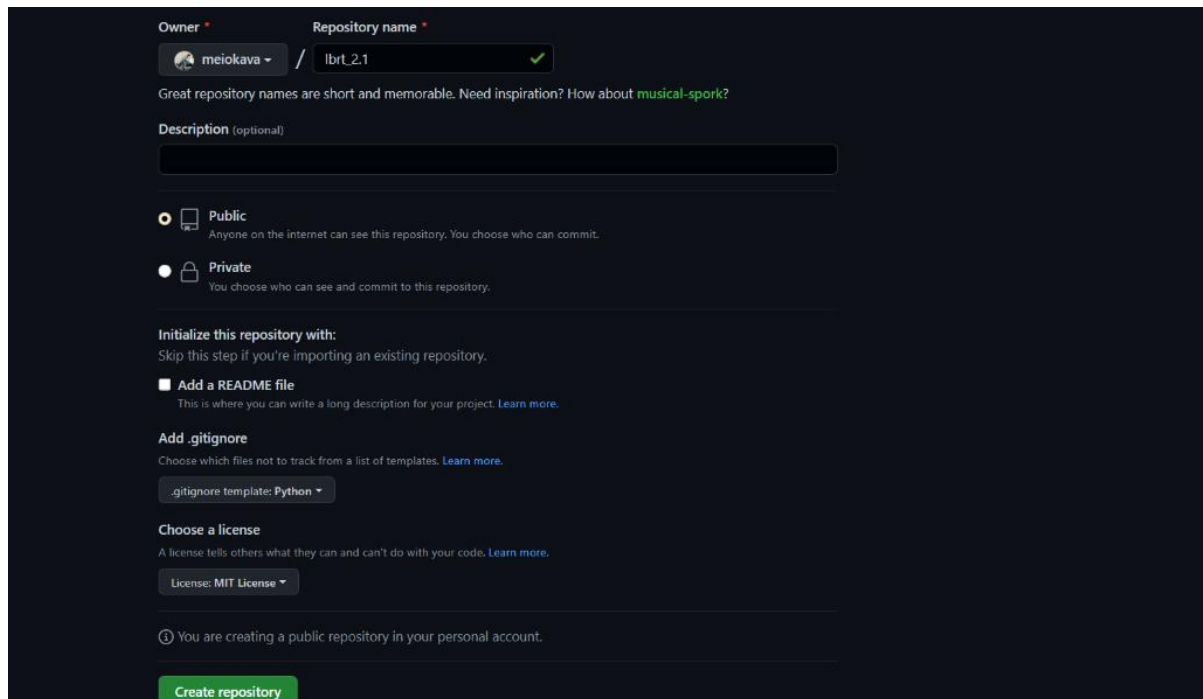


Рисунок 1 – был создан общедоступный репозиторий на Github с лицензией MIT и языком программирования Python

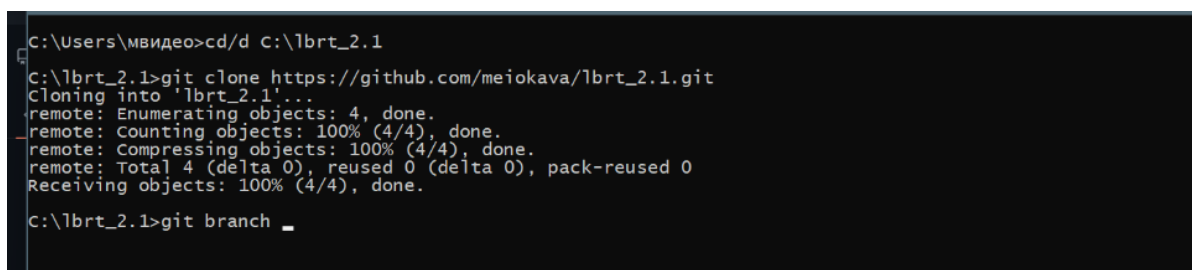


Рисунок 2 – Клонирование репозитория

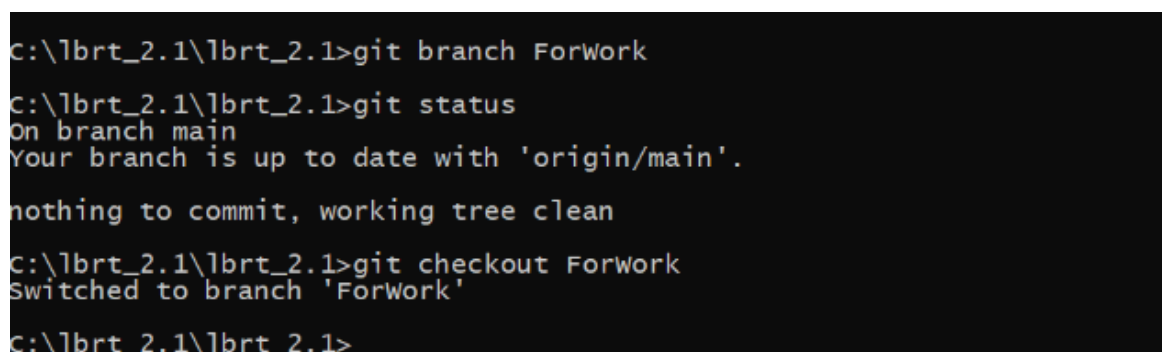


Рисунок 3 – Организация репозитория в соответствии с моделью ветвления git-flow

1. Напишите программу (файл user.py), которая запрашивала бы у пользователя:

его имя (например, "What is your name?")

возраст ("How old are you?")

место жительства ("Where are you live?")

После этого выводила бы три строки:

"This is `имя`"

"It is `возраст`"

"(S)he lives in `место_жительства`"

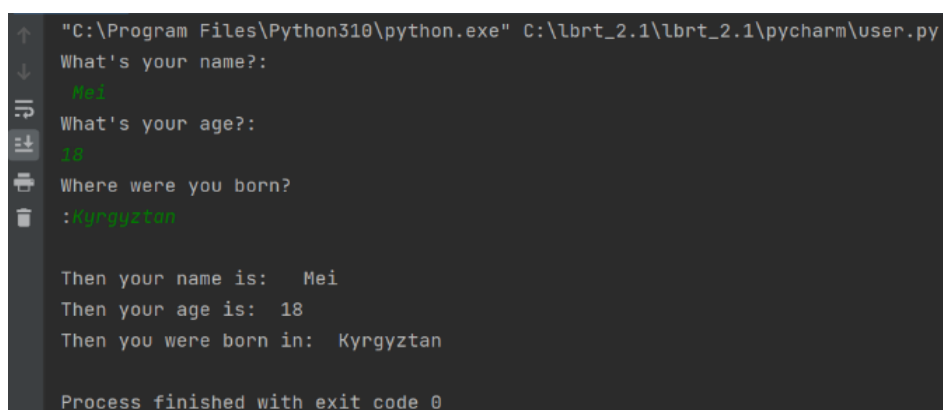
Вместо имя, возраст, местожительства должны быть данные, введенные пользователем.

Примечание: можно писать фразы на русском языке, но, если вы планируете стать профессиональным программистом, привыкайте к английскому.

Код программы

```
name = input ("What's your name?:\n")
age = input ("What's your age?:\n")
locBirth = input ("Where were you born?\n:")

print ("\nThen your name is: ", name)
print ("Then your age is: ", age)
print ("Then you were born in: ", locBirth)
```



```
"C:\Program Files\Python310\python.exe" C:\lbrt_2.1\lbrt_2.1\pycharm\user.py
What's your name?:
Mei
What's your age?:
18
Where were you born?
:Kyrgyztan

Then your name is: Mei
Then your age is: 18
Then you were born in: Kyrgyztan

Process finished with exit code 0
```

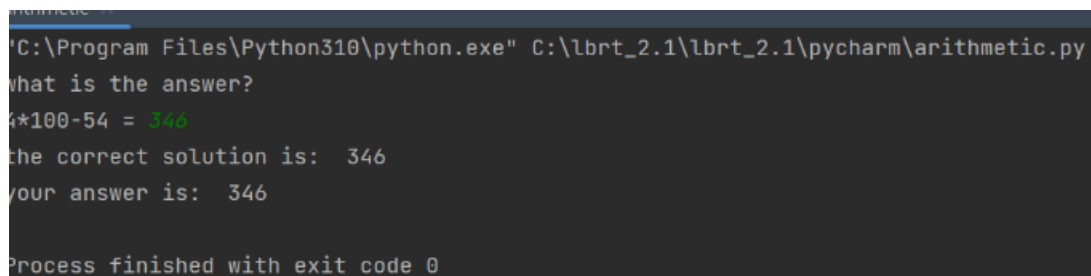
Рисунок 4 – Результат работы программы

2. Напишите программу (файл arithmetic.py), которая предлагала бы

пользователю решить пример $4 * 100 - 54$. Потом выводила бы на экран правильный ответ и ответ пользователя. Подумайте, нужно ли здесь преобразовывать строку в число.

Код программы:

```
sol = input("what is the answer?\n4*100-54 = ")  
  
corsol = 346  
print ("the correct solution is: ",corsol)  
print ("your answer is: ",sol)
```



```
C:\Program Files\Python310\python.exe "C:\lbrt_2.1\lbrt_2.1\pycharm\arithmetic.py"  
what is the answer?  
4*100-54 = 346  
the correct solution is: 346  
your answer is: 346  
  
Process finished with exit code 0
```

Рисунок 5 – Результат работы программы

3. Запросите у пользователя четыре числа (файл numbers.py). Отдельно сложите первые два и отдельно вторые два. Разделите первую сумму на вторую. Выведите результат на экран так, чтобы ответ содержал две цифры после запятой.

Код программы:

```
print("enter four digits: ")  
dig1 = int(input())  
dig2 = int(input())  
dig3 = int(input())  
dig4 = int(input())  
  
sum1 = dig1 + dig2  
sum2 = dig3 + dig4  
div = sum1 / sum2  
  
print("we have made some calculation\nthe result is: ",format(div,".3"))
```

```
"C:\Program Files\Python310\python.exe" C:\lbrt_2.1\lbrt_2.1\pycharm\numbers.py
enter four digits:
4
5
6
7
we have made some calculation
the result is: 0.692

Process finished with exit code 0
```

Рисунок 6 – Результат работы программы

4. Напишите программу (файл individual.py) для решения индивидуального задания. Вариант индивидуального задания уточните у преподавателя.

Даны длины сторон прямоугольного параллелепипеда. Найти его объем и площадь боковой поверхности.

Код программы:

```
print("enter sides of the figure: ")
a = int(input())
b = int(input())
c = int(input())

v = a * b * c
s = 2 * (a*b + b*c + a*c)
print("the volume is: ",v,"\nthe square is: ",s)
```

```
"C:\Program Files\Python310\python.exe" C:\lbrt_2.1\lbrt_2.1\pycharm\individual.py
enter sides of the figure:
5
7
4
the volume is: 140
the square is: 166

Process finished with exit code 0
|
```

Рисунок 7 – Результат выполнения программы

```

c:\lbrt_2.1\lbrt_2.1>git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

c:\lbrt_2.1\lbrt_2.1>git checkout ForWork
Switched to branch 'ForWork'

c:\lbrt_2.1\lbrt_2.1>git add .

c:\lbrt_2.1\lbrt_2.1>git commit -m "additions of folders"
[ForWork 615d0c6] additions of folders
10 files changed, 82 insertions(+)
create mode 100644 difficultT/dif1.py
create mode 100644 difficultT/dif2.py
create mode 100644 difficultT/dif3.py
create mode 100644 difficultT/dif5.py
create mode 100644 difficultT/dif6.py
create mode 100644 difficultT/dif7.py
create mode 100644 pycharm/arithmetic.py
create mode 100644 pycharm/individual.py
create mode 100644 pycharm/numbers.py
create mode 100644 pycharm/user.py

c:\lbrt_2.1\lbrt_2.1>git push

```

Рисунок 8 – Коммит изменений ветки ForWork

```

c:\lbrt_2.1\lbrt_2.1>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

c:\lbrt_2.1\lbrt_2.1>git merge ForWork
Updating 5c22421..615d0c6
Fast-forward
 difficultT/dif1.py      | 7 ++++++
 difficultT/dif2.py      | 8 ++++++
 difficultT/dif3.py      | 11 ++++++++
 difficultT/dif5.py      | 7 ++++++
 difficultT/dif6.py      | 7 ++++++
 difficultT/dif7.py      | 9 ++++++++
 pycharm/arithmetic.py   | 6 ++++++
 pycharm/individual.py   | 9 ++++++++
 pycharm/numbers.py      | 11 ++++++++
 pycharm/user.py         | 7 ++++++
10 files changed, 82 insertions(+)
create mode 100644 difficultT/dif1.py
create mode 100644 difficultT/dif2.py
create mode 100644 difficultT/dif3.py
create mode 100644 difficultT/dif5.py
create mode 100644 difficultT/dif6.py
create mode 100644 difficultT/dif7.py
create mode 100644 pycharm/arithmetic.py
create mode 100644 pycharm/individual.py
create mode 100644 pycharm/numbers.py
create mode 100644 pycharm/user.py

c:\lbrt_2.1\lbrt_2.1>_

```

Рисунок 8 – Слияние ветки ForWork с веткой main

```

c:\lbrt_2.1\lbrt_2.1>git push
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 8 threads
Compressing objects: 100% (14/14), done.
Writing objects: 100% (14/14), 1.98 KiB | 677.00 KiB/s, done.
Total 14 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/meiokava/lbrt_2.1.git
 5c22421..615d0c6 main -> main

c:\lbrt_2.1\lbrt_2.1>_

```

Рисунок 9 – Push на удаленный сервер

Вывод: в результате лабораторной работы исследования процесса установки и базовых возможностей языка Python, был получен опыт установки таких программ как Anaconda, PyCharm, Python, также были получены знания работы с PyCharm

Контрольные вопросы

1. Опишите основные этапы установки Python в Windows и Linux.

Для установки интерпретатора Python первое, что нужно сделать – это скачать дистрибутив. Загрузить его можно с официального сайта, перейдя по ссылке <https://www.python.org/downloads/>.

2. Запустить скачанный установочный файл.
3. Выбрать способ установки.
4. Отметить необходимые опции установки
5. Выбирать место установки

При установке для Linux, в случае ошибки необходимо либо собрать Python из исходников, либо взять из репозитория. Для установки из репозитория в Ubuntu воспользуйтесь командой «`sudo apt-get install python3`»

2. В чем отличие пакета Anaconda от пакета Python, скачиваемого с официального сайта?

Пакет Anaconda содержит версии языка Python 2 и 3, набор наиболее часто используемых библиотек и удобную среду разработки

3. Как осуществить проверку работоспособности пакета Anaconda?

В Windows это можно сделать, выбрав следующий пункт главного меню системы Пуск Anaconda3 (64-bit) Anaconda Prompt. В появившейся командной строке необходимо ввести «`jupyter notebook`», в результате чего отобразиться процесс загрузки веб-среды Jupyter Notebook, после чего запустится веб-сервер и среда разработки в браузере.

Создайте ноутбук для разработки, для этого нажмите на кнопку New (в правом углу окна) и в появившемся списке выберете Python. В результате

будет создана новая страница в браузере с ноутбуком. Введите в первой ячейке команду «`print("Hello, World!")`» и нажмите Alt+Enter на клавиатуре. Ниже ячейки должна появиться соответствующая надпись.

4. Как задать используемый интерпретатор языка Python в IDE PyCharm?

Указать путь до интерпретатора в настройках IDE, для этого:

- 1) Нажмите на шестеренку в верхнем правом углу, выберите "Add..".
- 2) Далее выберите "System Interpreter";
- 3) Нажмите на 3 точки "..." справа от поля в выбор интерпретатора;
- 4) Укажите путь до интерпретатора

5. Как осуществить запуск программы с помощью IDE PyCharm?
Shift+F10

6. В чем суть интерактивного и пакетного режимов работы Python?
В интерактивном.

Python можно использовать как калькулятор для различных вычислений, а если дополнительно подключить необходимые математические библиотеки, то по своим возможностям он становится практически равным таким пакетам как Matlab, Octave и т.п.

В проектном.

В этом режиме сначала записывается вся программа, а потом эта программа выполняется полностью.

7. Почему язык программирования Python называется языком динамической типизации?

В нем проверка типа происходит во время выполнения, а не компиляции

8. Какие существуют основные типы в языке программирования

Python?

1. None
2. Логические переменные
3. Числа
4. Списки
5. Строки
6. Бинарные списки
7. Множества
8. Словари

9. Как создаются объекты в памяти? Каково их устройство? В чем заключается процесс

Для того, чтобы объявить и сразу инициализировать переменную необходимо написать её имя, потом поставить знак равенства и значение, с которым эта переменная будет создана.

При инициализации переменной, на уровне интерпретатора, создается целочисленный объект, который имеет некоторый идентификатор, значение и тип. Посредством оператора “=” создается ссылка между переменной и объектом.

10. Как получить список ключевых слов в Python?

Нужно подключить модуль `keyword` и воспользоваться командой `keyword.kwlist`.

11. Каково назначение функций `id()` и `type()`?

Функция `id()` предназначена для получения значения идентичности объекта. С помощью функции `type()` можно получить тип конкретного объекта.

12. Что такое изменяемые и неизменяемые типы в Python.

К неизменяемым (immutable) типам относятся: целые числа (int), числа с плавающей точкой (float), комплексные числа (complex), логические переменные (bool), кортежи (tuple), строки (str) и неизменяемые множества (frozenset).

К изменяемым (mutable) типам относятся: списки (list), множества (set), словари (dict).

13. Чем отличаются операции деления и целочисленного деления?

При целочисленном делении отбрасывается дробная часть от деления чисел, при операции деления дробная часть не отбрасывается.

14. Какие имеются средства в языке Python для работы с комплексными числами?

Для создания комплексного числа можно использовать функцию `complex(a, b)`, в которую, в качестве первого аргумента, передается действительная часть, в качестве второго – мнимая. Либо записать число в виде $a + bj$. Комплексные числа можно складывать, вычитать, умножать, делить и возводить в степень. У комплексного числа можно извлечь действительную (`x.real`) и мнимую части (`x.imag`). Для получения комплексно сопряжённого числа необходимо использовать метод `conjugate()`.

15. Каково назначение и основные функции библиотеки (модуля) `math`? По аналогии с модулем `math` изучите самостоятельно назначение и основные функции модуля `cmath`.

`math.ceil(x)` - возвращает ближайшее целое число большее, чем x .

`math.fabs(x)` - возвращает абсолютное значение числа.

`math.factorial(x)` - вычисляет факториал x .

`math.floor(x)` - возвращает ближайшее целое число меньшее, чем x .

`math.exp(x)` - вычисляет $e^{**}x$.

`math.log2(x)` - логарифм по основанию 2.

`math.log10(x)` - логарифм по основанию 10.

`math.log(x[, base])` - по умолчанию вычисляет логарифм по основанию e , дополнительно можно указать основание логарифма.

`math.pow(x, y)` - вычисляет значение x в степени y .

`math.sqrt(x)` - корень квадратный от x .

`math.cos(x)` - косинус от x .

`math.sin(x)` - синус от x .

`math.tan(x)` - тангенс от x .

`math.acos(x)` - арккосинус от x .

`math.asin(x)` - арксинус от x .

`math.atan(x)` - арктангенс от x .

`math.pi` - число π .

`math.e` - число e .

16. Каково назначение именных параметров `sep` и `end` в функции `print()`?

`sep()` устанавливает отличный от пробела разделитель строк.

`End()` указывает, что делать, после вывода строки (по умолчанию стоит переход на новую строку)

17. Каково назначение метода `format()`? Какие еще существуют средства для форматирования строк в Python? Примечание: в дополнение к рассмотренным средствам изучите самостоятельно работу с f-строками в Python.

Форматирование может выполняться в так называемом старом стиле или с помощью строкового метода `format`. Символы `%s`, `%d`, `%f` подставляются значения переменных. Буквы `s`, `d`, `f` обозначают типы данных – строку, целое число, вещественное число.

18. Каким образом осуществить ввод с консоли значения целочисленной и вещественной переменной в языке Python?

Указать перед `input` тип данных: `int(input())`.