

**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

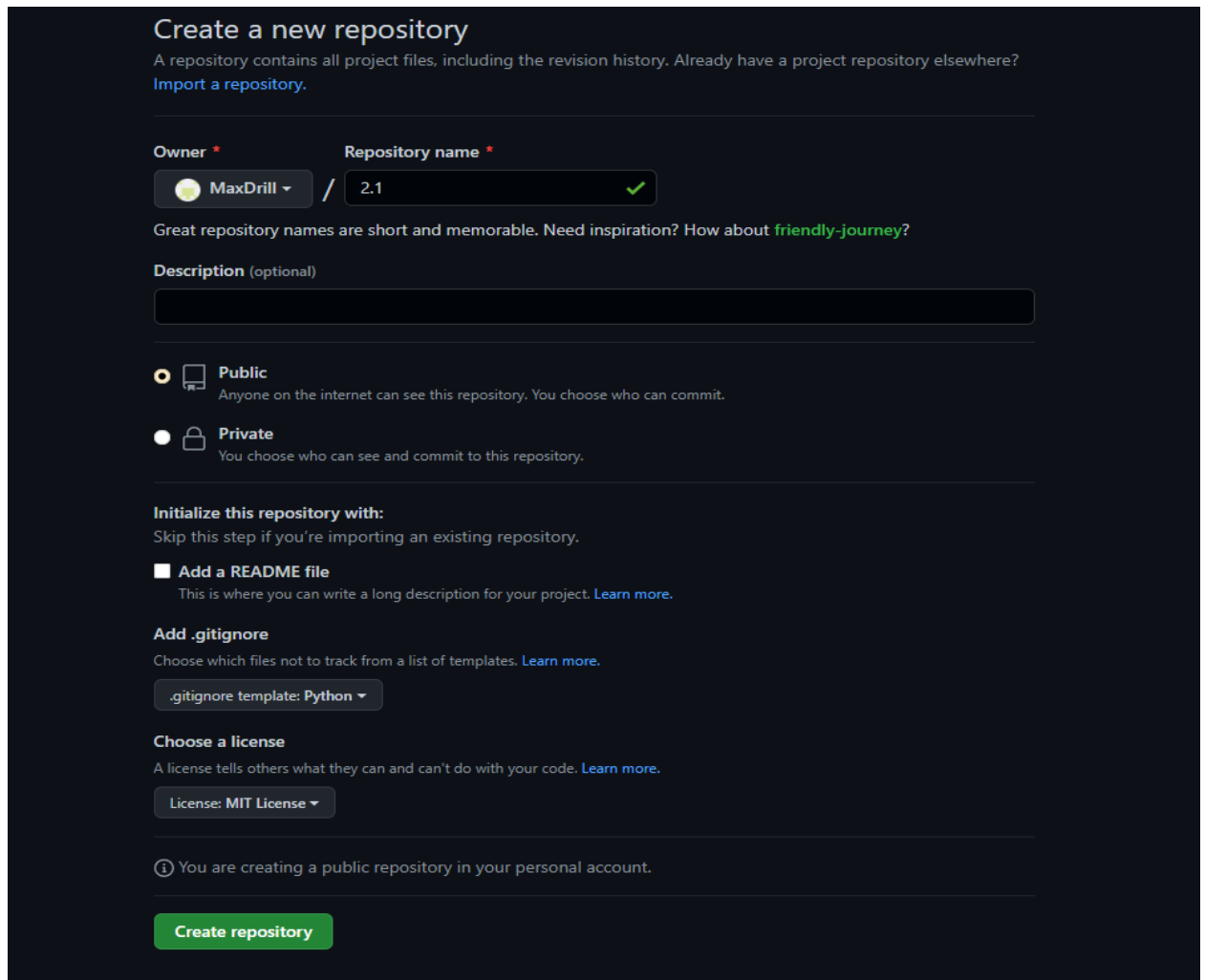
Дисциплина: «Основы программной инженерии»

Тема: «Основы ветвления GIT»

Выполнил:
студент 2 курса
группы Пиж-б-о-21-1
Коныжев Максим
Викторович

Ставрополь 2022

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



Create a new repository
A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * MaxDrill / **Repository name *** 2.1 ✓

Great repository names are short and memorable. Need inspiration? How about [friendly-journey?](#)

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)
.gitignore template: Python

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)
License: MIT License

ⓘ You are creating a public repository in your personal account.

Create repository

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

```
C:\Users\UESR\Desktop>git clone https://github.com/MaxDrill/2.1.git
Cloning into '2.1'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.

C:\Users\UESR\Desktop>
```

Рисунок 2 – Клонирование репозитория

```
C:\Users\UESR\Desktop>cd 2.1  
  
C:\Users\UESR\Desktop\2.1>git branch ForWork  
  
C:\Users\UESR\Desktop\2.1>git status  
On branch main  
Your branch is up to date with 'origin/main'.  
  
nothing to commit, working tree clean  
  
C:\Users\UESR\Desktop\2.1>git checkout ForWork  
Switched to branch 'ForWork'  
  
C:\Users\UESR\Desktop\2.1>
```

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

```
a = input("What is your name? ")  
b = input("How old are you? ")  
c = input("Where are you live? ")  
  
print("This is ", a)  
print("It is ", b)  
print("(S)he live in ",c)
```

```

F:\Python\UPI\venv\Scripts\python.exe F:/Python/UPI/use
What is your name? max
How old are you? 19
Where are you live? Stavropol
This is max
It is 19
(5)he live in Stavropol

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 = 4*100-54
print ("the correct solution is: ",corsol)
print ("your answer is: ",sol)

```

```

what is the answer?
4*100-54 =
the correct solution is: 346
your answer is:

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.

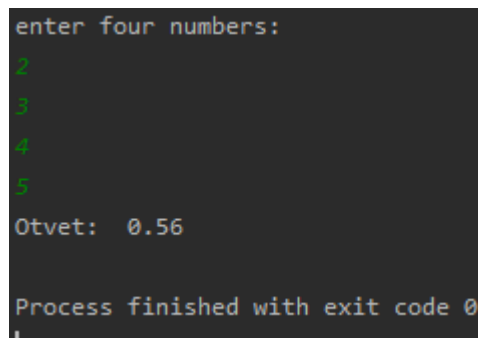
Код программы:

```
print("enter four numbers: ")

num1 = int(input())
num2 = int(input())
num3 = int(input())
num4 = int(input())

sum1 = num1 + num2
sum2 = num3 + num4
div = sum1 / sum2

print("Otvét: ",format(div,".2"))
```



```
enter four numbers:
2
3
4
5
Otvét: 0.56

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 основания и высота равнобедренной трапеции. Найти периметр трапеции.

Код программы:

```

from math import sqrt
a = float(input("Длина большего основания:"))
b = float(input("Длина меньшего основания:"))
h = float(input("Высота:"))

print("P =", a + b + 2 * sqrt(sqrt(h) + sqrt(a - b) / 4))

```

```

Длина большего основания:10
Длина меньшего основания:6
Высота:5
P = 19.308212796964423

Process finished with exit code 0
|

```

Рисунок 7 – Результат выполнения программы

```

C:\Users\UESR\Desktop\2.1>git commit -m "add of folders"
[ForWork b917de2] add of folders
10 files changed, 68 insertions(+)
create mode 100644 proj/.idea/.gitignore
create mode 100644 proj/.idea/inspectionProfiles/profiles_settings.xml
create mode 100644 proj/.idea/misc.xml
create mode 100644 proj/.idea/modules.xml
create mode 100644 proj/.idea/proj.iml
create mode 100644 proj/.idea/vcs.xml
create mode 100644 proj/arithmetic.py
create mode 100644 proj/individual.py
create mode 100644 proj/numbers.py
create mode 100644 proj/user.py
C:\Users\UESR\Desktop\2.1>

```

Рисунок 8 – Коммит изменений ветки ForWork

```

C:\Users\UESR\Desktop\2.1>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

C:\Users\UESR\Desktop\2.1>git merge ForWork
Updating fc27e44..b917de2
Fast-forward
 proj/.idea/.gitignore          | 2 ++
 proj/.idea/inspectionProfiles/profiles_settings.xml | 6 ++++++
 proj/.idea/misc.xml            | 4 +++++
 proj/.idea/modules.xml         | 8 ++++++++
 proj/.idea/proj.iml            |13 ++++++++
 proj/.idea/vcs.xml             | 6 ++++++
 proj/arithmetric.py            | 4 +++++
 proj/individual.py             | 6 ++++++
 proj/numbers.py                |12 ++++++++
 proj/user.py                   | 7 ++++++
10 files changed, 68 insertions(+)
create mode 100644 proj/.idea/.gitignore
create mode 100644 proj/.idea/inspectionProfiles/profiles_settings.xml
create mode 100644 proj/.idea/misc.xml
create mode 100644 proj/.idea/modules.xml
create mode 100644 proj/.idea/proj.iml
create mode 100644 proj/.idea/vcs.xml
create mode 100644 proj/arithmetric.py
create mode 100644 proj/individual.py
create mode 100644 proj/numbers.py
create mode 100644 proj/user.py

```

Рисунок 8 – Слияние ветки ForWork с веткой main

```

C:\Users\UESR\Desktop\2.1>git push
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 4 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (15/15), 2.15 KiB | 439.00 KiB/s, done.
Total 15 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/MaxDrill/2.1.git
   fc27e44..b917de2  main -> main

C:\Users\UESR\Desktop\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 отобразится 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 (frozen set).

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

е, 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 пи.

`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())`.