

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций
«Исследование возможностей Git для работы с
локальными репозиториями»**

**Отчет по лабораторной работе № 2.1
по дисциплине «Основы программной инженерии»**

Выполнил студент группы
ПИЖ-б-о-21-1

Трушева В. О. .«20» сентября
2022г.

Подпись студента _____

Работа защищена «
» _____ 20__ г.

Проверила Воронкин Р.А.

(подпись)

Ставрополь 2022

Методика и порядок выполнения работы

1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.

Рисунок 1 – Создание репозитория

3. Выполните клонирование созданного репозитория.

```
PS D:\fgit> git clone https://github.com/pynikcmd/4_LR.git
Cloning into '4_LR'...
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.
PS D:\fgit> |
```

Рисунок 2 – Клонирование репозитория на компьютер

4. Дополните файл .gitignore необходимыми правилами для работы с IDE PyCharm.

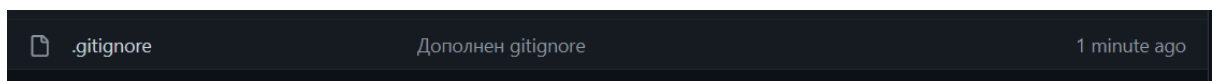


Рисунок 3 – Дополнен файл .gitignore

5. Организуйте свой репозиторий в соответствие с моделью ветвления git-flow.

```
PS D:\fgit\4_LR> 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:/fgit/4_LR/.git/hooks]
PS D:\fgit\4_LR>
```

Рисунок 4 – Модель ветвления git-flow

6. Создайте проект PyCharm в папке репозитория.

7. Решите следующие задачи с помощью языка программирования Python3 и IDE PyCharm:

8. 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:

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

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

```
1 name = input("What is your name? ")
2 years = input("How old are you? ")
3 living = input("Where are you live? ")
4
5 print("My name is ", name)
6 print("I am ", years)
7 print("I live in ", living)
```

Рисунок 5 – Код программы

```
D:\fgit\4_LR\venv\Scripts\python.exe
What is your name? Nika
How old are you? 21
Where are you live? Stavropol
My name is Nika
I am 21
I live in Stavropol
```

Рисунок 6 – Результат работы программы

9. 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.

```
print("Solve example: 4 * 100 - 54")
otvet = 4 * 100 - 54
primer = int(input("Enter the answer: "))
print("Right answer: {0} \nYour answer: {1}".format(otvet, primer))
```

Рисунок 7 – Код программы

```
D:\fgit\4_LR\venv\Scripts\pyt
Solve example: 4 * 100 - 54
Enter the answer: 5
Right answer: 346
Your answer: 5
```

Рисунок 8 – Результат работы программы

```
Python
PS D:\fgit\4_LR\venv> python arithmetic.py
Solve example: 4 * 100 - 54
Enter the answer: 45
Right answer: 346
Your answer: 45
PS D:\fgit\4_LR\venv> |
```

Рисунок 9 – Результат работы программы в командной строке

10. 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.

```
ch_1 = int(input("Enter number: "))
ch_2 = int(input("Enter number: "))
ch_3 = int(input("Enter number: "))
ch_4 = int(input("Enter number: "))

otvet = (ch_1 + ch_2) / (ch_3 + ch_4)

print("Answer: %.2f" % (otvet))
```

Рисунок 10 – Код программы

```
D:\fgit\4_LR\venv\Script
Enter number: 4
Enter number: 5
Enter number: 1
Enter number: 2
Answer: 3.00
Process finished with ex
```

Рисунок 11 – Результат работы программы

```
PS D:\fgit\4_LR\venv> python numbers.py
Enter number: 4
Enter number: 9
Enter number: 2
Enter number: 6
Answer: 1.62
PS D:\fgit\4_LR\venv> |
```

Рисунок 12 – Результат работы программы в командной строке

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

Вариант - 5

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

```
# Вариант - 5

a = int(input("Enter number: "))
b = int(input("Enter number: "))
c = int(input("Enter number: "))

print("Volume: {0} \nSide surface area: {1}".format(a*b*c, (a+b)*2*c))
```

Рисунок 13 – Код программы

```

D:\fgit\4_LR\venv\Scripts
Enter number: 5
Enter number: 4
Enter number: 2
Volume: 40
Side surface area: 36

```

Рисунок 14 – Результат работы программы

```

ANSWER: 1702
PS D:\fgit\4_LR\venv> python individual.py
Enter number: 4
Enter number: 6
Enter number: 5
Volume: 120
Side surface area: 100
PS D:\fgit\4_LR\venv>

```

Рисунок 15 – Результат работы программы в командной строке

12. Выполните коммит файлов user.py, arithmetic.py, numbers.py и individual.py в репозиторий git в ветку для разработки.

```

PS D:\fgit\4_LR> git add PyCharm
PS D:\fgit\4_LR> git commit -m "Add directory PyCharm"
[main 4c2d0d1] Add directory PyCharm
5 files changed, 33 insertions(+)
create mode 100644 PyCharm/arithmetic.py
create mode 100644 PyCharm/individual.py
create mode 100644 PyCharm/numbers.py
create mode 100644 PyCharm/user.py

```

Рисунок 16 – Коммит файлов

13. Выполните слияние ветки для разработки с веткой master.

```

PS D:\fgit\4_LR> git merge develop
Merge made by the 'ort' strategy.
PyCharm/arithmetic.py | 4 ++++
PyCharm/individual.py | 7 +++++++
PyCharm/numbers.py    | 8 +++++++
PyCharm/user.py       | 7 +++++++
4 files changed, 26 insertions(+)
create mode 100644 PyCharm/arithmetic.py
create mode 100644 PyCharm/individual.py
create mode 100644 PyCharm/numbers.py
create mode 100644 PyCharm/user.py

```

Рисунок 17 – Слияние веток

14. Отправьте сделанные изменения на сервер GitHub.

Задачи повышенной сложности

Вариант – 5

5. Даны целые числа h, m, s ($0 < h \leq 23, 0 \leq m \leq 59, 0 \leq s \leq 59$), указывающие момент времени: « h часов, m минут, s секунд». Определить угол (в градусах) между положением часовой стрелки в начале суток и в указанный момент времени.

```
h = int(input("hour: "))
m = int(input("minute: "))
s = int(input("second: "))
h %= 12
m %= 60
s %= 60
g = 360/12*h + 30/60*m + 30/3600*s
print("Gradus: %.3f" % g)
```

Рисунок 18 – Код программы

```
D:\fgit\4_LR\PyCharm\venv\S
hour: 5
minute: 18
second: 5
Gradus: 159.042
```

Рисунок 19 – Результат работы программы

Контрольные вопросы

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

Для установки интерпретатора Python первое, что нужно сделать – это

скачать дистрибутив. Запустить скачанный установочный файл. Выбрать способ установки. Отметить необходимые опции установки (доступно при выборе Customize installation). Выберете место установки (доступно при выборе Customize installation).

При установке для Linux, в случае ошибки необходимо либо собрать Python из исходников либо взять из репозитория.

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

Anaconda является дистрибутивом языков программирования таких как Python и R для науки о данных и машинного обучения, а Python — это язык программирования высокого уровня общего назначения. Этот пакет включает в себя интерпретатор языка Python (есть версии 2 и 3), набор наиболее часто используемых библиотек и удобную среду разработки и исполнения, запускаемую в браузере.

3. Как осуществить проверку работоспособности пакета Anaconda?

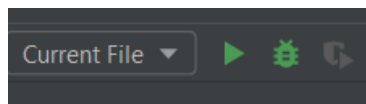
Для выполнения проверки работоспособности Anaconda необходимо вначале запустить командный процессор с поддержкой виртуальных окружений Anaconda. В Windows это можно сделать выбрав следующий пункт главного меню системы Пуск Anaconda3 (64-bit) Anaconda Prompt. В появившейся командной строке необходимо 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?

При создании нового проекта в PyCharm есть возможность выбрать путь до проекта и интерпретатор.

5. Как осуществить запуск программы с помощью IDE PyCharm?

Правой кнопкой в любом месте или по файлу слева и выбрать из появившегося меню пункт «Run». Или в верхнем меню нажать на зеленый треугольник:



Или сочетание клавиш “Ctrl+Shift+F10”

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

Интерактивный режим позволяет вводить команды, которые сразу же будут обрабатываться, это можно использовать в качестве калькулятора. Пакетный режим позволяет запустить файл с исходным кодом.

7. Почему язык программирования Python называется языком динамической типизации?

Значит, что переменная в разное время может ссылаться на данные разного типа.

8. Какие существуют основные типы в языке программирования Python?

1. None (неопределенное значение переменной)
2. Логические переменные (Boolean Type)
3. Числа (Numeric Type)
 1. int – целое число
 2. float – число с плавающей точкой

- 3. `complex` – комплексное число
- 4. Списки (Sequence Type)
 - 1. `list` – список
 - 2. `tuple` – кортеж
 - 3. `range` – диапазон
- 5. Строки (Text Sequence Type)
 - 1. `Str`
- 6. Бинарные списки (Binary Sequence Types)
 - 1. `bytes` – байты
 - 2. `bytearray` – массивы байт
 - 3. `memoryview` – специальные объекты для доступа к внутренним данным объекта через `protocol buffer`
- 7. Множества (Set Types)
 - 1. `set` – множество
 - 2. `frozenset` – неизменяемое множество
- 8. Словари (Mapping Types)
 - 1. `dict` – словарь

9. Как создаются объекты в памяти? Каково их устройство? В чем заключается процесс объявления новых переменных и работа операции присваивания?

Для того, чтобы объявить и сразу инициализировать переменную необходимо написать её имя, потом поставить знак равенства и значение, с которым эта переменная будет создана. Объект, в данном случае – это абстракция для представления данных, данные – это числа, списки, строки и т.п. Каждый объект имеет три атрибута – это идентификатор, значение и тип. Идентификатор – это уникальный признак объекта, позволяющий отличать объекты друг от друга, а значение – непосредственно информация, хранящаяся в памяти, которой управляет интерпретатор.

10. Как получить список ключевых слов в Python?

Список ключевых слов можно получить непосредственно в программе, для этого нужно подключить модуль `keyword` и воспользоваться командой `keyword.kwlist`.

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

Для того, чтобы посмотреть на объект с каким идентификатором ссылается данная переменная, можно использовать функцию `id()`.

Тип переменной можно определить с помощью функции `type()`.

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 остаток от деления не учитывается.

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.

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 комплексносопряженного 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()`?

Через параметр `sep` можно указать отличный от пробела разделитель строк. Параметр `end` позволяет указывать, что делать, после вывода строки.

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`.

F-Строки являются упрощенной версией метода `format()`. F-strings являются строковыми литералами с «f» в начале и фигурные скобки, содержащие выражения, которые в дальнейшем будут заменены своими значениями. Пример: `print(f"Hello, {name}. You are {2*8}.")` //name – переменная

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

Сначала выполняется функция `input()`. Она возвращает строку, которую функция `int()` или `float()` сразу преобразует в число. Только после этого происходит присваивание переменной, то есть она сразу получает численное значение.