

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт перспективной инженерии
Департамент цифровых, робототехнических систем и электроники

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №4
дисциплины
«Основы кроссплатформенного программирования»
Вариант 1

Выполнил:
Бакулин Вадим Романович
2 курс, группа ИТС-б-о-23-1,
11.03.02
«Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные
системы и сети»,
очная форма обучения

(подпись)

Проверил:
Доцент департамента цифровых,
робототехнических систем и
электроники
Воронкин Р.А.

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г.

Лабораторная работа 1.3 «Основы языка Python»

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

Ссылка на GitHub: <https://github.com/zepteloid/LR4>

Ход работы:

1. Создан общедоступный репозиторий на GitHub

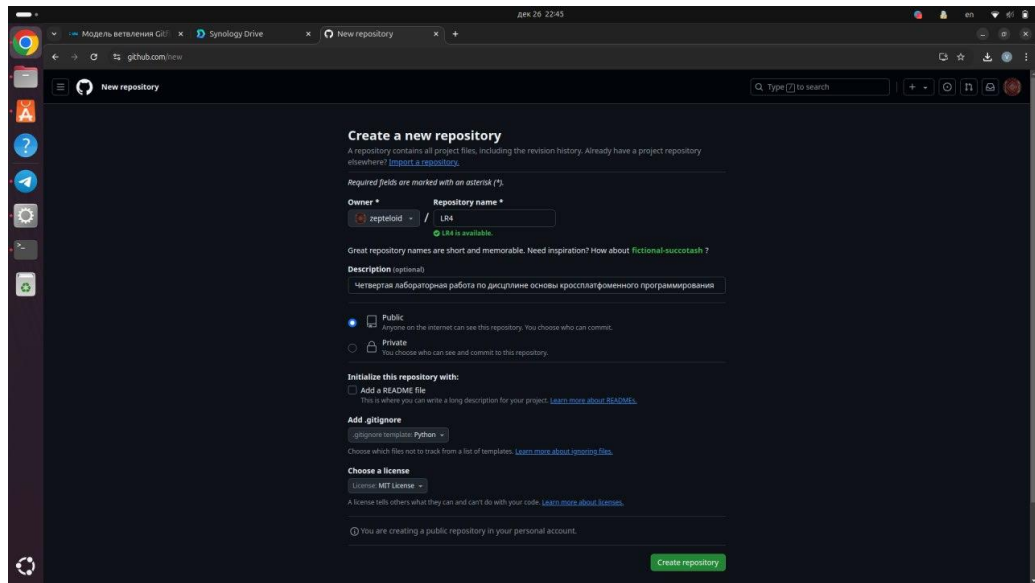


Рисунок 1. Общедоступный репозиторий

2. Выполнил задание «user.py»:

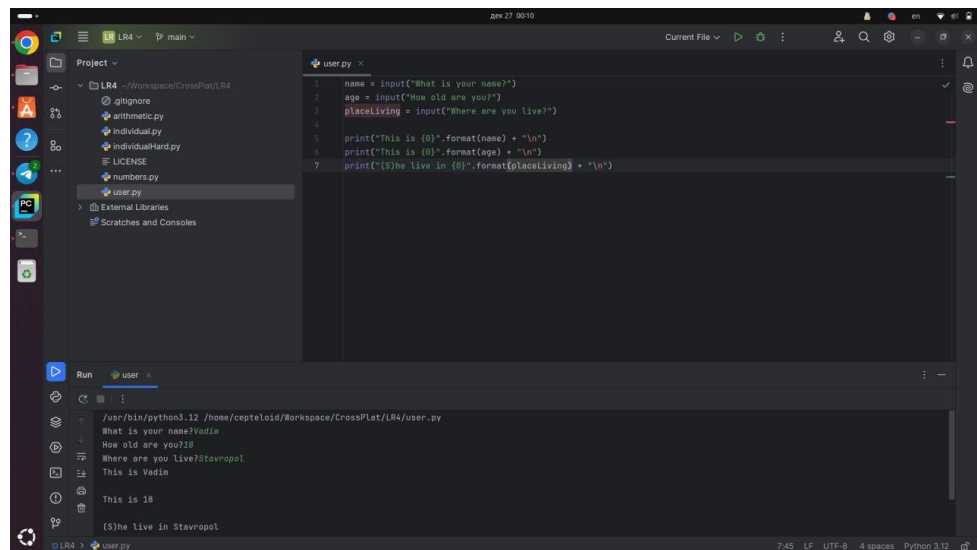


Рисунок 2. Программа задания «user»

3. Выполнил задание «arithmatic»:

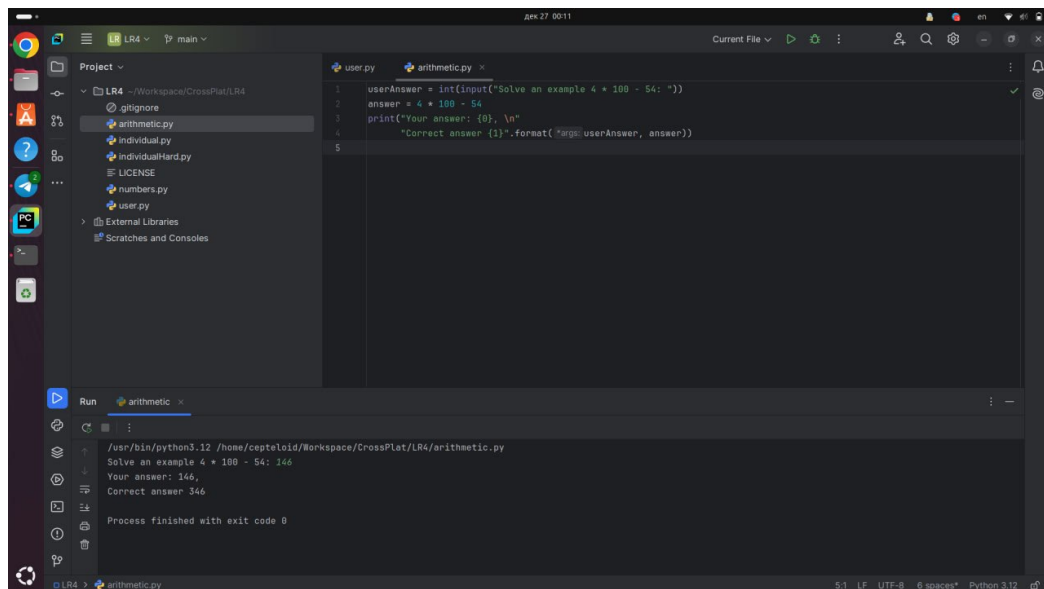


Рисунок 3. Программа задания «arithmetic»

4. Выполнил задание «numbers»

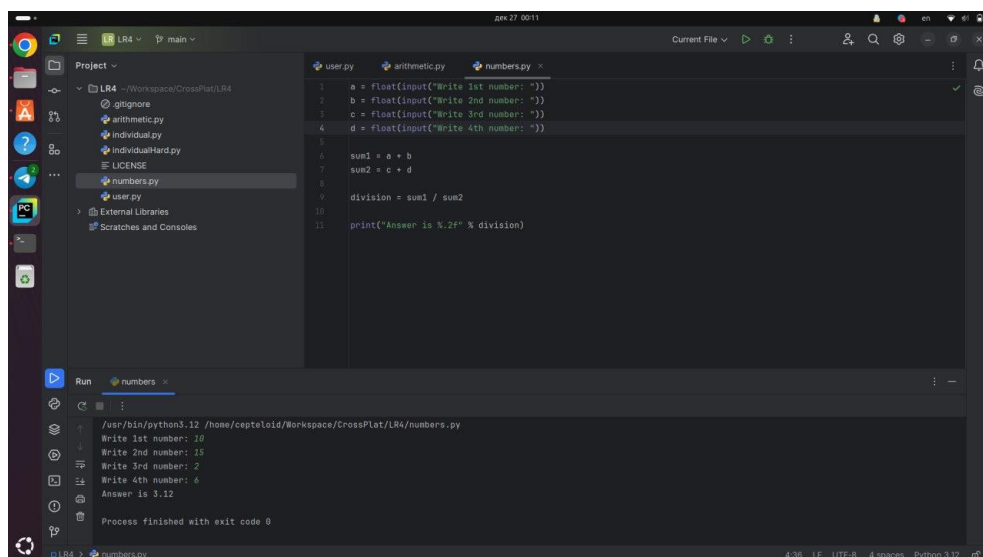


Рисунок 4. Программа задания «numbers»

5. Выполнил индивидуальное задание:

Индивидуальные задания

1. Даны два числа. Найти среднее арифметическое и среднее геометрическое их модулей.

Рисунок 5. Индивидуальное задание в соответствии с вариантом

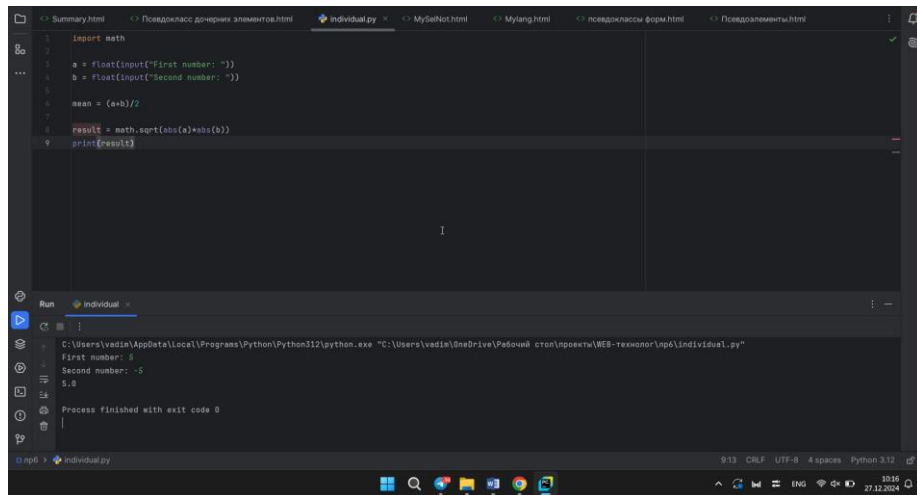


Рисунок 6. Программа для индивидуального задания

6. Закоммитил изменения

```

cepteloid@cepteloid-BoDE-WXX9:~/Workspace/CrossPlat/LR4$ git commit -m "commit files"
[develop 97beb80] commit files
11 files changed, 103 insertions(+)
create mode 100644 .idea/.gitignore
create mode 100644 .idea/LR4.iml
create mode 100644 .idea/inspectionProfiles/profiles_settings.xml
create mode 100644 .idea/misc.xml
create mode 100644 .idea/modules.xml
create mode 100644 .idea/vcs.xml
create mode 100644 arithmetic.py
create mode 100644 individual.py
create mode 100644 individualHard.py
create mode 100644 numbers.py
create mode 100644 user.py

```

Рисунок 7. Коммит созданных файлов

7. Выполнил слияние ветки разработки с основной веткой

```

cepteloid@cepteloid-BoDE-WXX9:~/Workspace/CrossPlat/LR4$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
cepteloid@cepteloid-BoDE-WXX9:~/Workspace/CrossPlat/LR4$ git merge develop
Updating 7be9057..97beb80
Fast-forward
 .idea/.gitignore          | 3 +++
 .idea/LR4.iml             | 8 ++++++++
 .idea/inspectionProfiles/profiles_settings.xml | 6 ++++++
 .idea/misc.xml            | 7 ++++++
 .idea/modules.xml         | 8 ++++++++
 .idea/vcs.xml             | 6 ++++++
 arithmetic.py             | 4 ++++
 individual.py             | 30 +++++++++++++++++++++++++++++++++++++
 individualHard.py         | 13 ++++++++
 numbers.py               | 11 ++++++++
 user.py                   | 7 ++++++
11 files changed, 103 insertions(+)
create mode 100644 .idea/.gitignore
create mode 100644 .idea/LR4.iml
create mode 100644 .idea/inspectionProfiles/profiles_settings.xml
create mode 100644 .idea/misc.xml
create mode 100644 .idea/modules.xml
create mode 100644 .idea/vcs.xml
create mode 100644 arithmetic.py
create mode 100644 individual.py
create mode 100644 individualHard.py
create mode 100644 numbers.py
create mode 100644 user.py

```

Рисунок 8. Слияние веток

8. Запустил изменения на удаленный сервер:

```
zepteloid@zepteloid-BoDE-WXX9:~/Workspace/CrossPlat/LR4$ git push
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 8 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (15/15), 2.53 KiB | 864.00 KiB/s, done.
Total 15 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:zepteloid/LR4.git
7be9057..97beb80  main -> main
```

Рисунок 9. Пуш на удаленный сервер

Ответы на контрольные вопросы:

1. Основные этапы установки Python в Windows и Linux:

Windows: Скачайте установочный файл с официального сайта Python, запустите его и следуйте инструкциям, не забыв отметить опцию "Add Python to PATH".

Linux: Используйте пакетный менеджер вашей дистрибуции, например, `sudo apt install python3` для Ubuntu.

2. Различие пакета Anaconda от пакета Python:

Anaconda включает в себя множество библиотек и инструментов для научных вычислений и анализа данных, таких как Jupyter Notebook и Conda.

Официальный Python поставляется только с основными библиотеками и инструментами.

3. Проверка работоспособности пакета Anaconda:

Запустите `conda list` в командной строке, чтобы увидеть установленные пакеты.

4. Задание интерпретатора Python в IDE PyCharm:

Перейдите в *File > Settings > Project: <ваш проект> > Python Interpreter* и выберите необходимый интерпретатор.

5. Запуск программы с помощью IDE PyCharm:

Нажмите кнопку *Run* или используйте сочетание клавиш *Shift + F10*.

6. Суть интерактивного и пакетного режимов работы Python:

Интерактивный режим: Прямой ввод и выполнение команд в интерактивной оболочке.

Пакетный режим: Выполнение команд из сохраненного файла скрипта.

7. Почему Python называется языком динамической типизации:

Переменные в Python могут изменять свой тип во время выполнения программы, и тип данных определяется автоматически.

8. Основные типы данных в Python:

int, float, str, list, tuple, dict, set, bool.

9. Создание объектов в памяти и объявление переменных:

Объекты создаются при присваивании значений переменным, а память под них выделяется автоматически.

10. Получение списка ключевых слов в Python:

Выполните команду `import keyword; print(keyword.kwlist)`.

11. Назначение функций `id()` и `type()`:

`id()` возвращает уникальный идентификатор объекта.

`type()` возвращает тип объекта.

12. Изменяемые и неизменяемые типы в Python:

Изменяемые: list, dict, set.

Неизменяемые: int, float, str, tuple.

13. Отличие операций деления и целочисленного деления:

`/` выполняет обычное деление.

`//` выполняет целочисленное деление.

14. Средства Python для работы с комплексными числами:

Встроенные функции и модуль `cmath`.

15. Назначение и функции библиотеки `math`:

Библиотека `math` предоставляет математические функции, такие как `sqrt`, `sin`, `cos`.

Модуль `cmath` аналогичен `math`, но для комплексных чисел.

16. Назначение параметров `sep` и `end` в функции `print()`:

`sep` задает разделитель между аргументами.

`end` задает окончание строки.

17. Назначение метода `format()` и другие средства форматирования строк:

Метод `format()` используется для форматирования строк.

Альтернативы: f-строки, оператор `%`.

18. Ввод значения целочисленной и вещественной переменной с консоли:

```
int_value = int(input("Введите целое число: "))
```

```
float_value = float(input("Введите вещественное число: "))
```

Вывод: в ходе выполнения лабораторной работы был исследован процесс установки и базовые конструкции и возможности языка Python.