

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

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №8
дисциплины «Основы программной инженерии»

Выполнил:
Матвеев Александр Иванович
2 курс, группа ПИЖ-б-о-22-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка и
сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: Работа с кортежами в языке Python

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

Ход работы.

1. Создание нового репозитория с лицензией MIT.

Create a new repository

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

Required fields are marked with an asterisk ().*

Owner * SashkaHacker / **Repository name *** laba3
✅ laba3 is available.

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

Description (optional)
Выполнение лабораторной работы №3 по дисциплине: основы программной инженерии

☒ **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:

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

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

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

This will set `main` as the default branch. Change the default name in your [settings](#).

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

Create repository

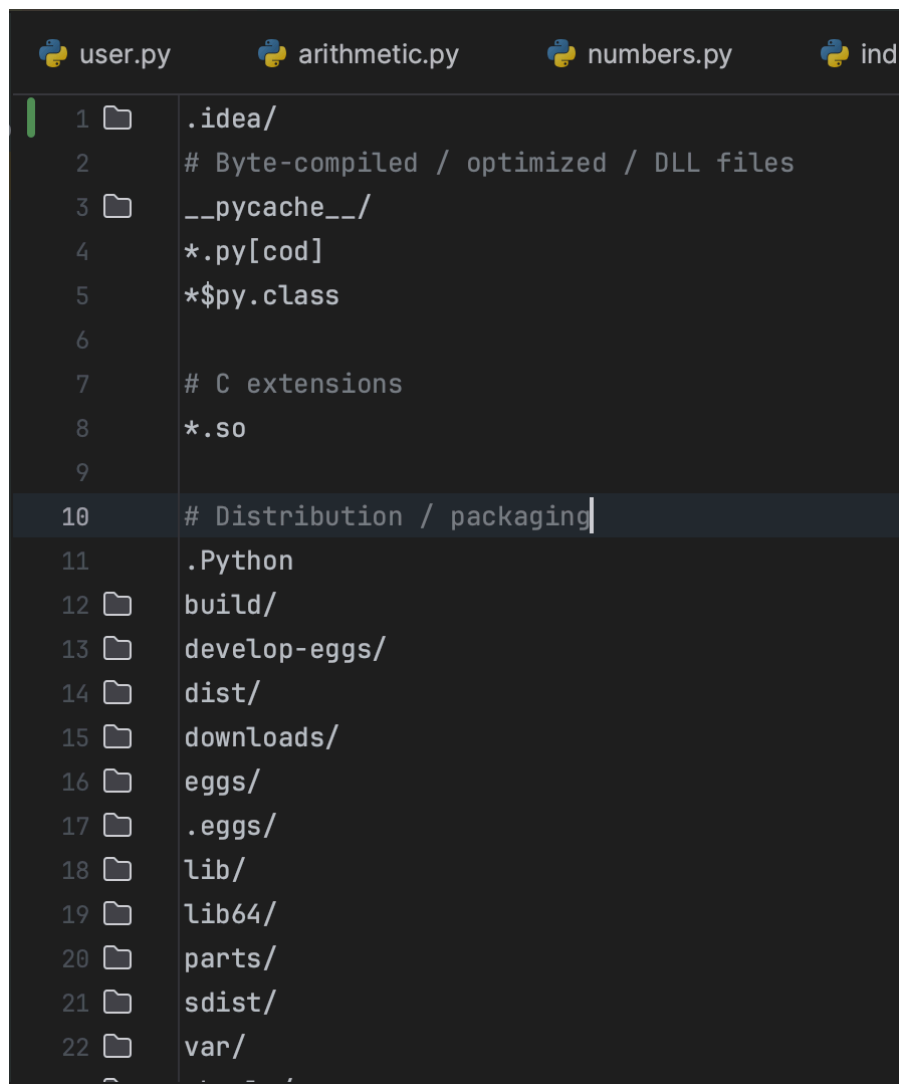
Рисунок 1 – Создание репозитория

2. Клонировал репозиторий на рабочий ПК.

```
Sashka@DESKTOP-U4RPSBI MINGW64 ~/Documents/GitHub/laba6 (develop)
$ git push origin develop
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/SashkaHacker/laba6/pull/new/develop
remote:
To https://github.com/SashkaHacker/laba6.git
 * [new branch]      develop -> develop
```

Рисунок 2 – Клонирование репозитория

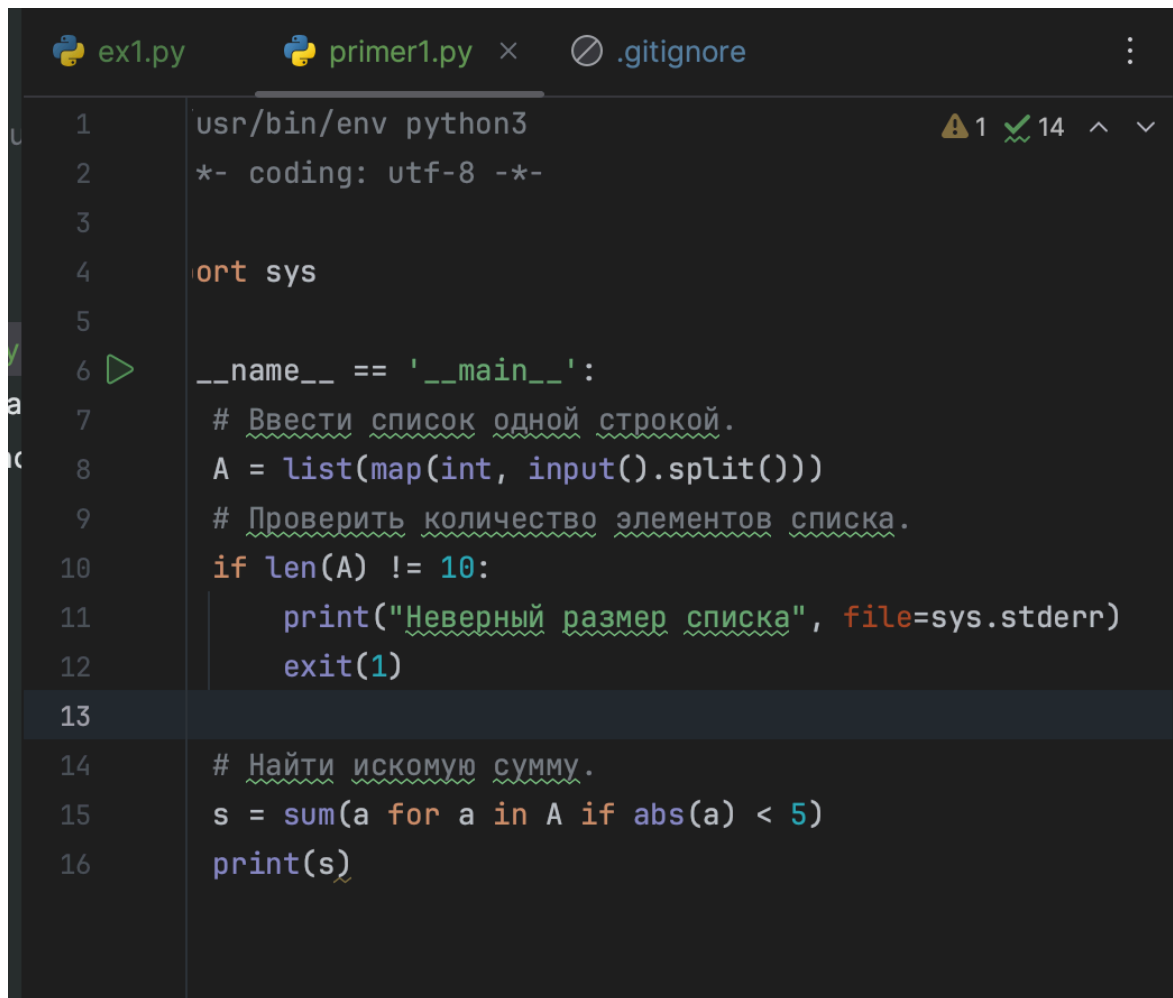
3. Дополнил файл .gitignore необходимыми инструкциями.



```
user.py arithmetic.py numbers.py ind
1  .idea/
2  # Byte-compiled / optimized / DLL files
3  __pycache__/
4  *.py[cod]
5  *$py.class
6
7  # C extensions
8  *.so
9
10 # Distribution / packaging
11 .Python
12 build/
13 develop-eggs/
14 dist/
15 downloads/
16 eggs/
17 .eggs/
18 lib/
19 lib64/
20 parts/
21 sdist/
22 var/
```

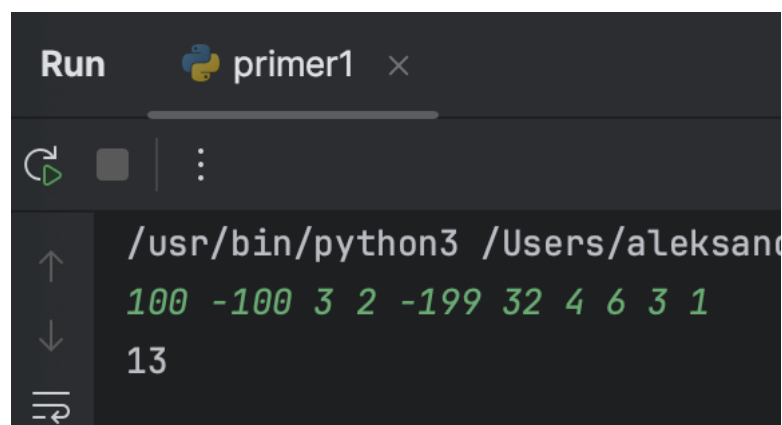
Рисунок 3 – Файл .gitignore

4. Проработка задания №1 из методических указаний.



```
1  /usr/bin/env python3
2  -*- coding: utf-8 -*-
3
4  import sys
5
6  __name__ == '__main__':
7      # Ввести список одной строкой.
8      A = list(map(int, input().split()))
9      # Проверить количество элементов списка.
10     if len(A) != 10:
11         print("Неверный размер списка", file=sys.stderr)
12         exit(1)
13
14     # Найти искомую сумму.
15     s = sum(a for a in A if abs(a) < 5)
16     print(s)
```

Рисунок 4 – Код программы

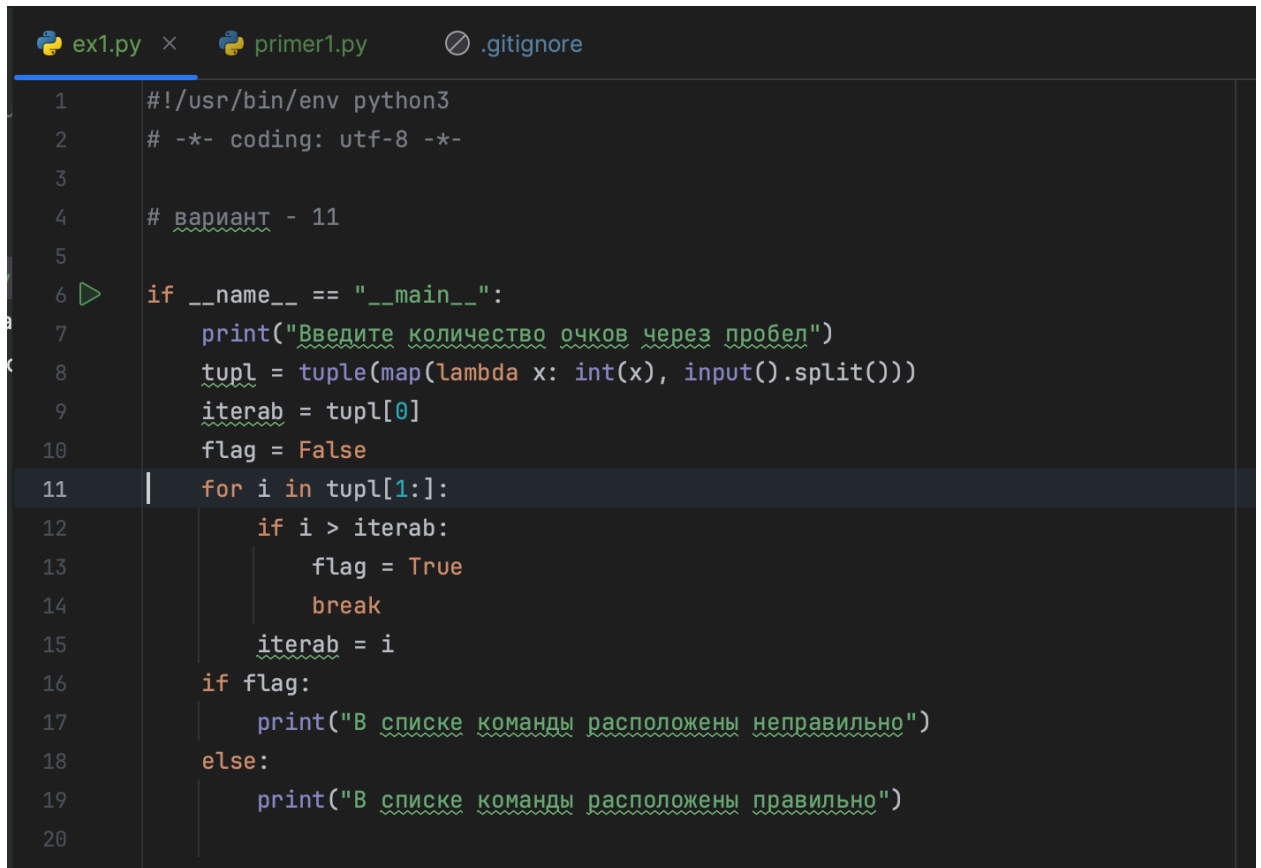


```
Run primer1 x
100 -100 3 2 -199 32 4 6 3 1
13
```

Рисунок 5 – Пример работы программы

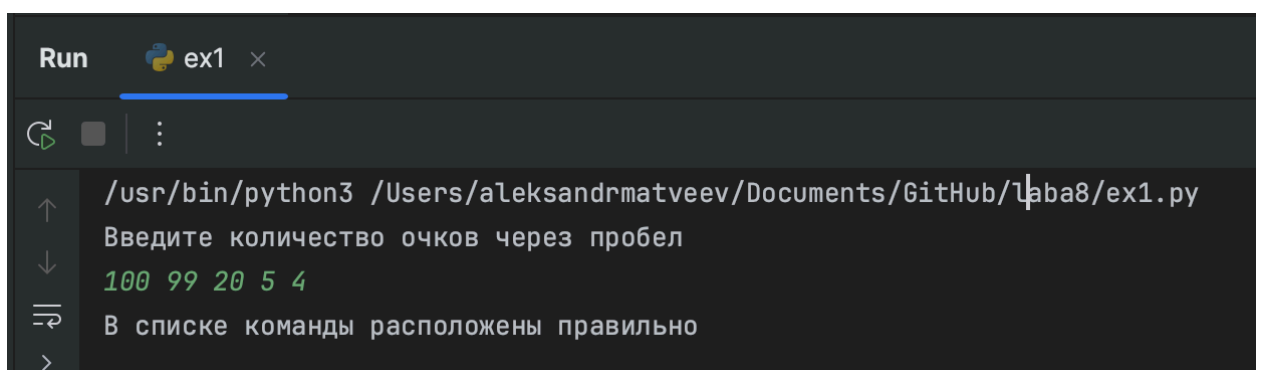
5. Выполнение индивидуального задания №1 (Вариант-11).

Условие: имеются данные о сумме очков, набранных в чемпионате каждой из футбольных команд. Определить, перечислены ли команды в списке в соответствии с занятыми ими местами в чемпионате.



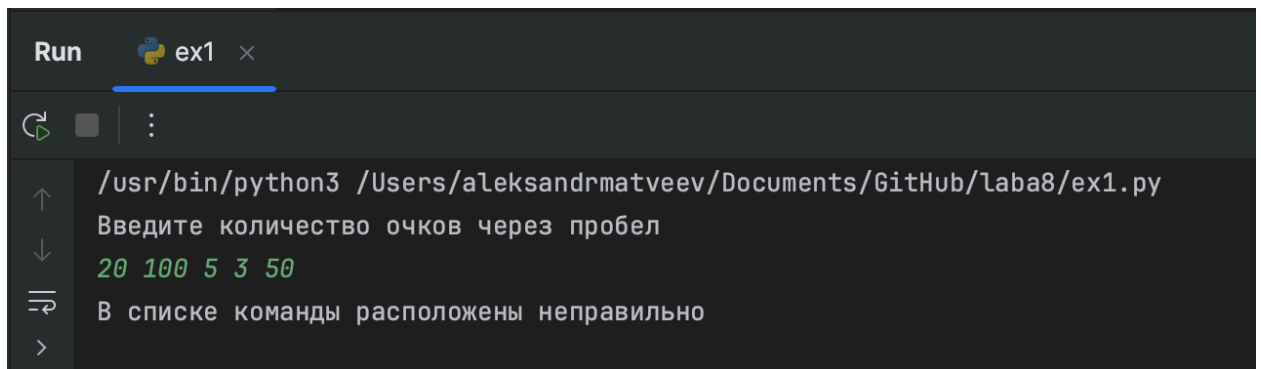
```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  # вариант - 11
5
6  if __name__ == "__main__":
7      print("Введите количество очков через пробел")
8      tupl = tuple(map(lambda x: int(x), input().split()))
9      iterab = tupl[0]
10     flag = False
11     for i in tupl[1:]:
12         if i > iterab:
13             flag = True
14             break
15         iterab = i
16     if flag:
17         print("В списке команды расположены неправильно")
18     else:
19         print("В списке команды расположены правильно")
20
```

Рисунок 6 – Код программы



```
Run  ex1 x
┌───┴───┐
└─┬──┬─┘
  ↑  ↓  ||
  /usr/bin/python3 /Users/aleksandrmatveev/Documents/GitHub/1aba8/ex1.py
  Введите количество очков через пробел
  100 99 20 5 4
  В списке команды расположены правильно
  >
```

Рисунок 7 – Пример выполнения программы



```
Run ex1 x
/usr/bin/python3 /Users/aleksandrmatveev/Documents/GitHub/lab8/ex1.py
Введите количество очков через пробел
20 100 5 3 50
В списке команды расположены неправильно
```

Рисунок 8 – Пример выполнения программы

Контрольные вопросы:

1. Что такое кортежи в языке Python?

О: это неизменяемая структура данных для хранения объектов различных типов.

2. Каково назначение кортежей в языке Python?

О: кортежи предназначены в первую очередь для хранения различных типов данных, но в отличие от списков, они защищены от изменений. Именно из-за этой особенности кортежей они используются в Python. Но также помимо этого они занимают меньше места и работают быстрее, чем списки.

3. Как осуществляется создание кортежей?

О: для создания кортежа достаточно: `a = (1, 2, 3)`; `a = (1,)`; `a = tuple([1, 2, 3])`

4. Как осуществляется доступ к элементам кортежа?

О: для обращения к элементу кортежа стоит использовать индекс.

5. Зачем нужна распаковка (деструктуризация) кортежа?

О: нужно для того, чтобы получить каждый элемент в отдельной переменной, особенно полезно, если функция возвращает несколько элементов (то она вернет кортеж)

6. Какую роль играют кортежи в множественном присваивании?

О: если функция возвращает несколько переменных, то она возвращает кортеж.

7. Как выбрать элементы кортежа с помощью среза?

О: путем создания диапазона в квадратных скобках [start:stop]

8. Как выполняется конкатенация и повторение кортежей?

О: конкатенация выполняется при помощи оператора +, повторение при помощи оператора *.

9. Как выполняется обход элементов кортежа?

О: непосредственно запустить цикл for по кортежу, либо при помощи цикла while и индекса элементов.

10. Как проверить принадлежность элемента кортежу?

О: при помощи оператора in.

11. Какие методы работы с кортежами Вам известны?

О: index(), count(), min(), max(), sum()

12. Допустимо ли использование функций агрегации таких как len() , sum() и т. д. при работе с кортежами?

О: да, так написано прямо в документации.

13. Как создать кортеж с помощью спискового включения.

О: tuple([input() for i in range(n)])