

**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 №2.5**  
Дисциплина: «Основы кроссплатформенного  
программирования» Тема: «Работа с картами в языке Python»

Выполнил: студент 1 курса  
группы ИВТ-б-о-21-1  
Богдашов Артём  
Владимирович

Ставрополь 2022

## Выполнение работы:

Создал репозиторий в GitHub «rep 2.5» в который добавил .gitignore, который дополнил правила для работы с IDE PyCharm с ЯП Python, выбрал лицензию MIT, клонировал его на лок. сервер и организовал в соответствии с моделью ветвления git-flow.

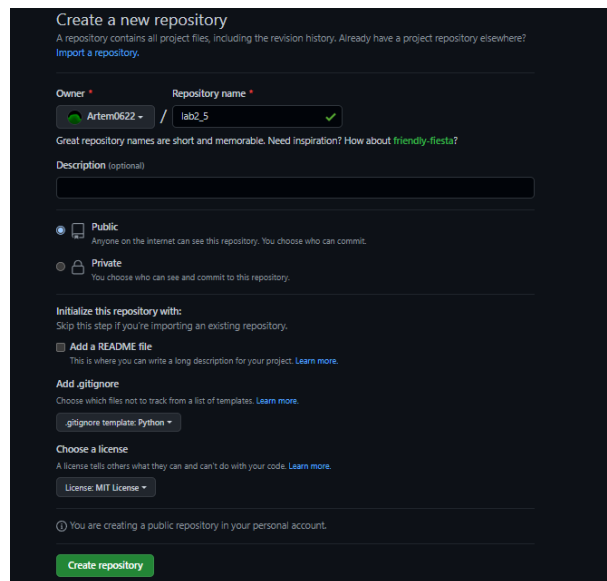


Рисунок 1.1 Создание репозитория

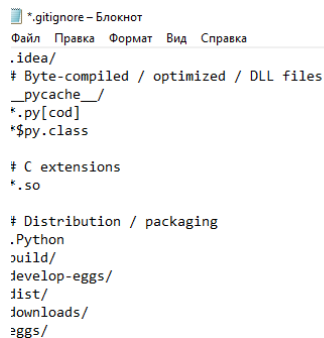
```
C:\Games\Программы\СУ\1.2КС\Программирование\2.5>git clone https://github.com/Artem0622/lab2_5.git
Cloning into 'lab2_5'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
```

Рисунок 1.2 Клонирование репозитория

```
C:\Games\Программы\СУ\1.2КС\Программирование\2.5\lab2_5>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? [C:/Games/Программы/СУ/1.2КС/Программирование/2.5/lab2_5/.git/hooks]
```

Рисунок 1.3 Организация репозитория в соответствии с моделью ветвления git-flow



```

*.gitignore - Блокнот
Файл  Правка  Формат  Вид  Справка
.idea/
# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
*.py.class

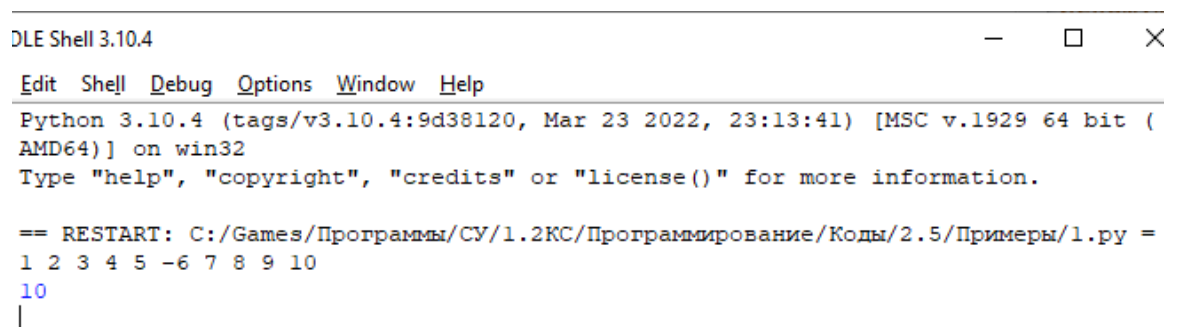
# C extensions
*.so

# Distribution / packaging
.Python
build/
develop-eggs/
dist/
downloads/
eggs/

```

Рисунок 1.4 Изменение .gitignore

**Пример 1.** Ввести кортеж A из 10 элементов, найти сумму элементов, меньших по модулю 5, и вывести ее на экран. Использовать в программе вместо списков кортежи.



```

DLE Shell 3.10.4
Edit Shell Debug Options Window Help
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

== RESTART: C:/Games/Программы/СУ/1.2КС/Программирование/Коды/2.5/Примеры/1.py ==
1 2 3 4 5 -6 7 8 9 10
10
|

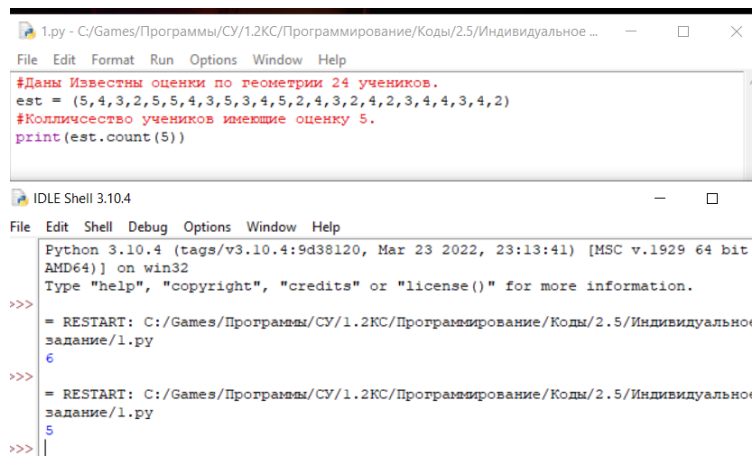
```

Рисунок 2.1 Результат выполнения программы.

## Индивидуальное задание:

### Вариант 3

Известны оценки по геометрии каждого из 24 учеников класса. В начале списка перечислены все пятерки, затем все остальные оценки. Сколько учеников имеет по геометрии оценку «5»? Условный оператор не использовать.



```
1.py - C:/Games/Программы/СУ/1.2КС/Программирование/Коды/2.5/Индивидуальное ...
File Edit Format Run Options Window Help

#Даны Известны оценки по геометрии 24 учеников.
est = (5,4,3,2,5,5,4,3,5,3,4,5,2,4,3,2,4,2,3,4,4,3,4,2)
#Количество учеников имеющие оценку 5.
print(est.count(5))

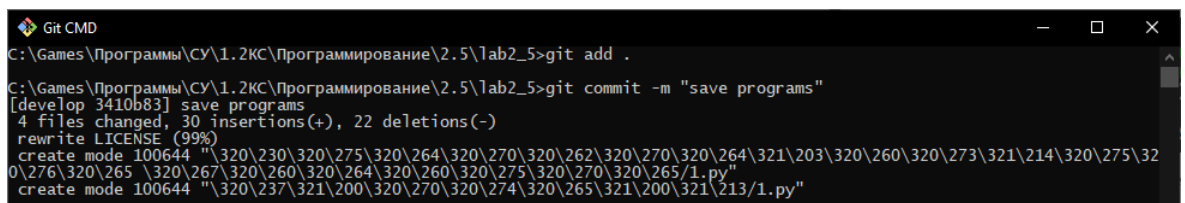
IDLE Shell 3.10.4
File Edit Shell Debug Options Window Help

Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit
AMD64] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> = RESTART: C:/Games/Программы/СУ/1.2КС/Программирование/Коды/2.5/Индивидуальное
задание/1.py
6
>>> = RESTART: C:/Games/Программы/СУ/1.2КС/Программирование/Коды/2.5/Индивидуальное
задание/1.py
5
>>> |
```

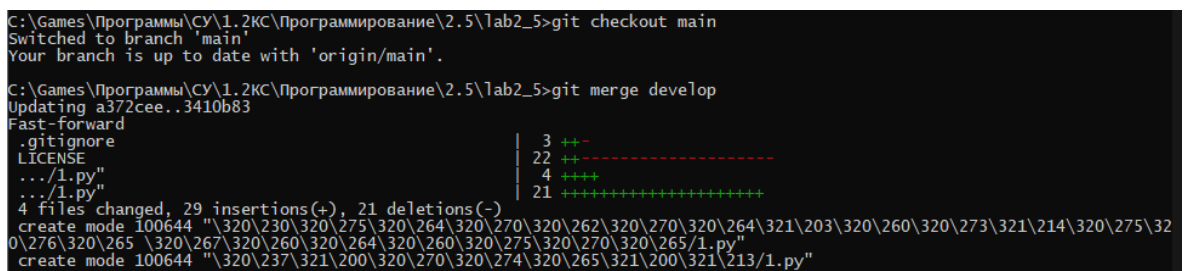
Рисунок 3 Результат выполнения программы

Сделал коммит, выполнил слияние с веткой main, и запустил изменения в уд. репозиторий.



```
Git CMD
C:\Games\Программы\СУ\1.2КС\Программирование\2.5\lab2_5>git add .
C:\Games\Программы\СУ\1.2КС\Программирование\2.5\lab2_5>git commit -m "save programs"
[develop 3410b83] save programs
4 files changed, 30 insertions(+), 22 deletions(-)
rewrite LICENSE (99%)
create mode 100644 "\320\230\320\275\320\264\320\270\320\262\320\270\320\264\321\203\320\260\320\273\321\214\320\275\32
0\276\320\265 \320\267\320\260\320\264\320\260\320\275\320\270\320\265\1.py"
create mode 100644 "\320\237\321\200\320\270\320\274\320\265\321\200\321\213\1.py"
```

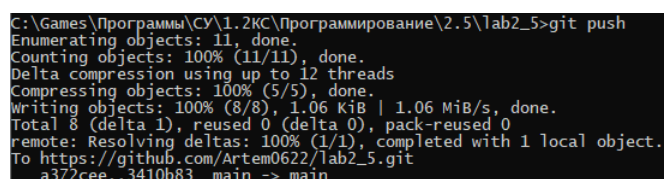
Рисунок 4.1 коммит изменений и переход на ветку main



```
C:\Games\Программы\СУ\1.2КС\Программирование\2.5\lab2_5>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

C:\Games\Программы\СУ\1.2КС\Программирование\2.5\lab2_5>git merge develop
Updating a372cee..3410b83
Fast-forward
 .gitignore      | 3  ++
 LICENSE        | 22  ++-----
 .../1.py"      | 4   ++++
 .../1.py"      | 21  ++++++
4 files changed, 29 insertions(+), 21 deletions(-)
create mode 100644 "\320\230\320\275\320\264\320\270\320\262\320\270\320\264\321\203\320\260\320\273\321\214\320\275\32
0\276\320\265 \320\267\320\260\320\264\320\260\320\275\320\270\320\265\1.py"
create mode 100644 "\320\237\321\200\320\270\320\274\320\265\321\200\321\213\1.py"
```

Рисунок 4.2 Слияние ветки main с develop



```
C:\Games\Программы\СУ\1.2КС\Программирование\2.5\lab2_5>git push
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (8/8), 1.06 KiB | 1.06 MiB/s, done.
Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Artem0622/lab2_5.git
a372cee..3410b83 main -> main
```

Рисунок 4.3 Пуш изменений на удаленный сервер

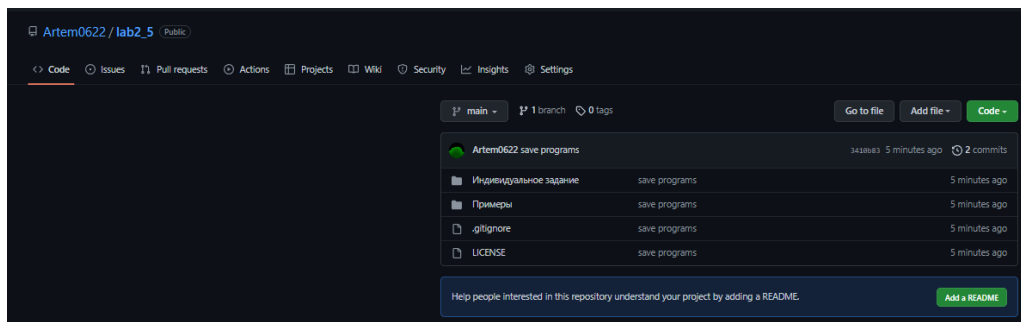


Рисунок 4.4 Изменения на удаленном сервере

## Контрольные вопросы:

### 1. Что такое кортежи в языке Python?

Кортеж (tuple) – это неизменяемая структура данных, которая по своему подобию очень похожа на список.

### 2. Каково назначение кортежей в языке Python?

Существует несколько причин, по которым стоит использовать кортежи вместо списков. Одна из них – это обезопасить данные от случайного изменения. Если мы получили откуда-то массив данных, и у нас есть желание поработать с ним, но при этом непосредственно менять данные мы не собираемся, тогда, это как раз тот случай, когда кортежи придутся как нельзя кстати. Кортежи в памяти занимают меньший объем по сравнению со списками. Кортежи работают быстрее, чем списки

### 3. Как осуществляется создание кортежей?

```
a = ()
```

```
b = tuple()
```

### 4. Как осуществляется доступ к элементам кортежа?

Доступ к элементам кортежа осуществляется также как к элементам списка – через указание индекса.

### 5. Зачем нужна распаковка (деструктуризация) кортежа?

Обращение по индексу, это не самый удобный способ работы с кортежами. Дело в том, что кортежи часто содержат значения разных типов, и помнить, по какому индексу что лежит – очень непросто.

## **6. Какую роль играют кортежи в множественном присваивании?**

Используя множественное присваивание, можно провернуть интересный трюк: обмен значениями между двумя переменными.

## **7. Как выбрать элементы кортежа с помощью среза?**

С помощью операции взятия среза можно получить другой кортеж.

Общая форма операции взятия среза для кортежа следующая

$$T2 = T1[i:j]$$

здесь

- $T2$  – новый кортеж, который получается из кортежа  $T1$ ;
- $T1$  – исходный кортеж, для которого происходит срез;
- $i, j$  – соответственно нижняя и верхняя границы среза.

Фактически берутся ко вниманию элементы, лежащие на позициях  $i, i+1, \dots, j-1$ . Значение  $j$  определяет позицию за последним элементом среза.

## **8. Как выполняется конкатенация и повторение кортежей?**

Для кортежей можно выполнять операцию конкатенации, которая обозначается символом  $+$ .

$$T3 = T1 + T2$$

## **9. Как выполняется обход элементов кортежа?**

Элементы кортежа можно последовательно просмотреть с помощью операторов цикла `while` или `for`.

## **10. Как проверить принадлежность элемента кортежу?**

Проверка вхождения элемента в кортеж - оператор `in`.

## **11. Какие методы работы с кортежами Вам известны?**

`index()`, `count()`.

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

Доступно.

## **13. Как создать кортеж с помощью спискового включения.**

Так же как и список.

