

**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 №7**

Дисциплина: «Основы кроссплатформенного программирования»

Тема: «Работа со списками в языке Python»

Выполнил: студент 1 курса

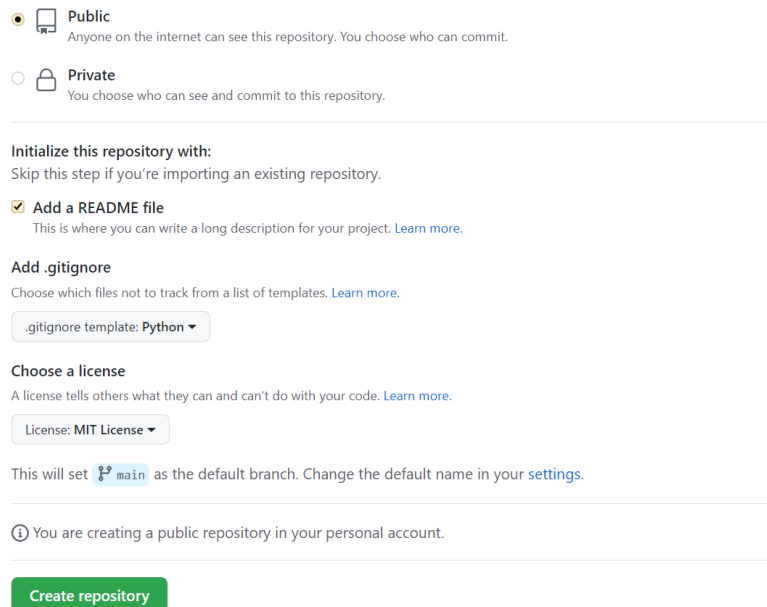
группы ИВТ-б-о-21-1

Уланбекова Айканыш Уланбековна

Ставрополь 2022

## Выполнение работы:

1. Создала репозиторий в GitHub «rep 2.4» в который добавила .gitignore, который дополнила правила для работы с IDE PyCharm с ЯП Python, выбрала лицензию MIT, клонировала его на лок. сервер и организовала в соответствии с моделью ветвления git-flow.



☒ **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**

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 1.1 Создание репозитория

```
C:\Users\User\Desktop\lab7>git clone https://github.com/aikanyshkaukanbekova/lab7.git
Cloning into 'lab7'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
```

Рисунок 1.2 Клонирование репозитория

```
C:\Users\User\Desktop\lab7\lab7>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:/Users/User/Desktop/lab7/lab7/.git/hooks]
```

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



The image shows a text editor window titled ".gitignore - Блокнот". The menu bar includes "Файл", "Правка", "Формат", "Вид", and "Справка". The content of the file is as follows:

```
.idea/  
# Created by https://www.toptal.com/developers/gitignore/api/python,pycharm  
# Edit at https://www.toptal.com/developers/gitignore?templates=python,pycharm  
  
### PyCharm ###  
# Covers JetBrains IDEs: IntelliJ, RubyMine, PhpStorm, AppCode, PyCharm, CLion, Android Studio, WebStorm and Rider  
# Reference: https://intellij-support.jetbrains.com/hc/en-us/articles/206544839  
  
# User-specific stuff  
.idea/**/workspace.xml  
.idea/**/tasks.xml  
.idea/**/usage.statistics.xml  
.idea/**/dictionaries  
.idea/**/shelf  
  
# AWS User-specific  
.idea/**/aws.xml  
  
# Generated files  
.idea/**/contentModel.xml  
  
# Sensitive or high-churn files  
.idea/**/dataSources/
```

Рисунок 1.4 Изменение .gitignore

2. Создала проект PyCharm в папке репозитория, проработала примеры ЛР.

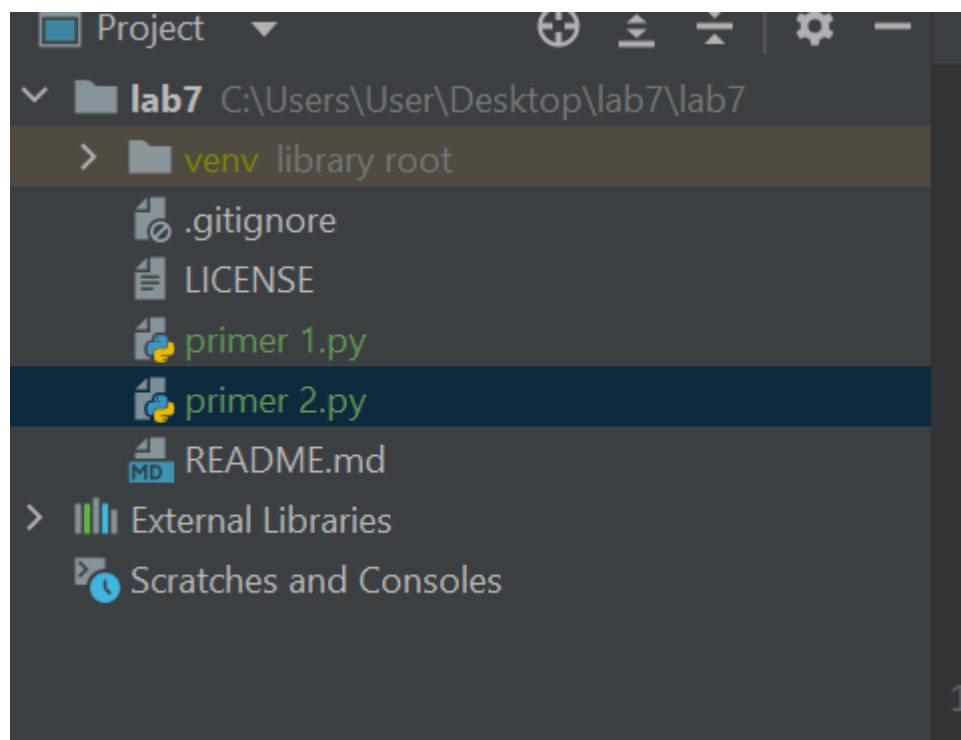


Рисунок 2.1 Создание проекта в PyCharm



The image shows the output of a program, with numbers displayed in green on a black background:

```
1 5 4 6 8 7 9 5 1 3  
9
```

Рисунок 2.2 Рез-т выполнения программы

```
2 3 6 5 4 8 9 4 8 7
13
```

Рисунок 2.3 Рез-т выполнения программы

### 3. Выполнила 2 индивидуальных задания. Вариант 23

```
Понедельник
Введите значение утренней температуры: 1
Введите значение дневной температуры: 2
Введите значение вечерней температуры: 1

Вторник
Введите значение утренней температуры: -5
Введите значение дневной температуры: -1
Введите значение вечерней температуры: -10

Среда
Введите значение утренней температуры: 2
Введите значение дневной температуры: 3
Введите значение вечерней температуры: 1

Четверг
Введите значение утренней температуры: 10
Введите значение дневной температуры: 16
Введите значение вечерней температуры: 8

Пятница
Введите значение утренней температуры: 25
Введите значение дневной температуры: 31
Введите значение вечерней температуры: 22

Суббота
Введите значение утренней температуры: -45
Введите значение дневной температуры: -100
Введите значение вечерней температуры: -200

Воскресенье
Введите значение утренней температуры: 1
Введите значение дневной температуры: 1
Введите значение вечерней температуры: 1

Средние температуры за утро, день и вечер соответственно: [1, -5, 2, 10, 26, -115, 1]
```

Рисунок 3.1 Вывод программы индивидуального задания 1

```

Введите количество чисел в списке: 5
Введите числа в список:
0
1
-9
5
0
Произведение элементов списка с четными номерами = 5
Сумму элементов списка, расположенных между первым и последним нулевыми элементами -3
Отсортированный список: [5, 1, 0, 0, -9]

```

Рисунок 3.2 Вывод программы индивидуального задания 2

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

```

C:\Users\User\Desktop\lab7\lab7>git add .
C:\Users\User\Desktop\lab7\lab7>git commit -m "d"
[develop db22d73] d
4 files changed, 108 insertions(+)
create mode 100644 individyalnoe/indiv 1.py
create mode 100644 individyalnoe/indiv 2.py
create mode 100644 primer/primer 1.py
create mode 100644 primer/primer 2.py
C:\Users\User\Desktop\lab7\lab7>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

```

Рисунок 4.1 коммит изменений и переход на ветку main

```

C:\Users\adamk\OneDrive\Рабочий стол\rep_2.4>git merge develop
Updating b61f559..3356fc8
Fast-forward
 .gitignore | 153 ++++++
 1 file changed, 150 insertions(+), 3 deletions(-)

```

Рисунок 4.2 Слияние ветки main с develop

```

C:\Users\User\Desktop\lab7\lab7>git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 1.87 KiB | 1.87 MiB/s, done.
Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/aikanyshkaukanbekova/lab7.git
ab8bcdc..265ec02 main -> main

```

Рисунок 4.3 Пуш изменений на удаленный сервер

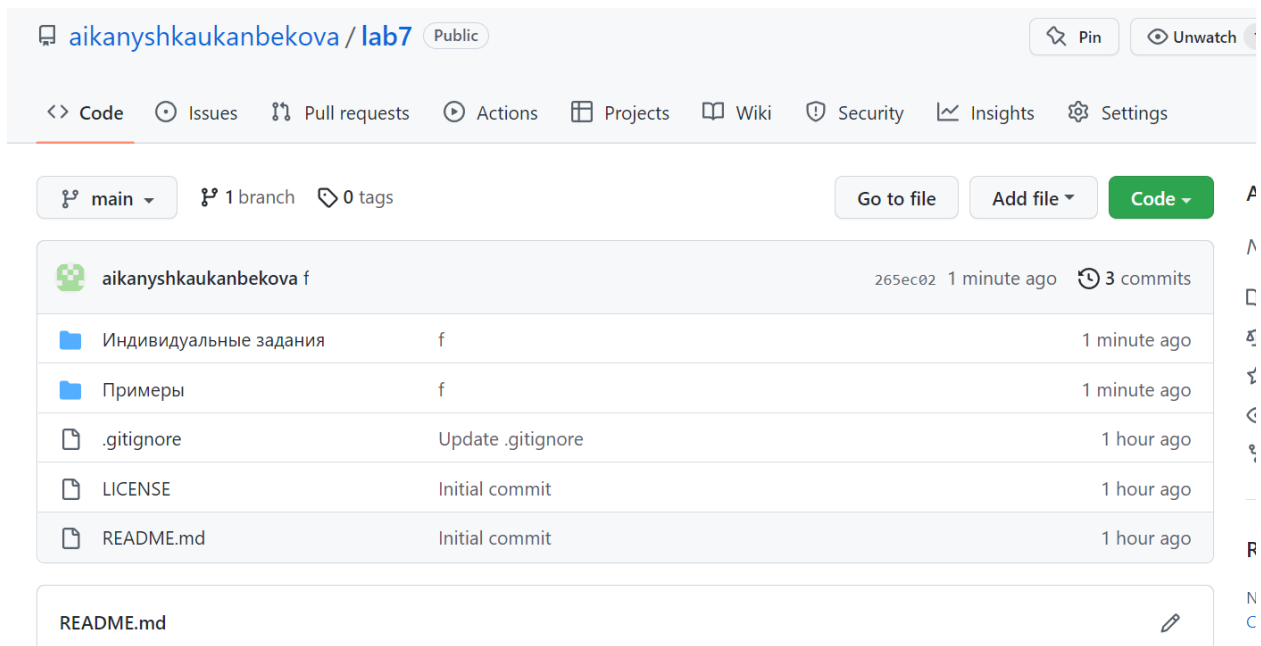


Рисунок 4.4 Изменения на удаленном сервере

### Контр. вопросы и ответы на них:

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

Список (list) – это структура данных для хранения объектов различных типов.

#### 2. Как осуществляется создание списка в Python?

Для создания списка нужно заключить элементы в квадратные скобки.

#### 3. Как организовано хранение списков в оперативной памяти?

Список является изменяемым типом данных. При его создании в памяти резервируется область, которую можно условно назвать некоторым “контейнером”, в котором хранятся ссылки на другие элементы данных в памяти. В отличие от таких типов данных как число или строка, содержимое “контейнера” списка можно менять.

#### 4. Каким образом можно перебрать все элементы списка?

```
for elem in my_list:
```

#### 5. Какие существуют арифметические операции со списками?

+, \*

#### 6. Как проверить есть ли элемент в списке?

Для того, чтобы проверить, есть ли заданный элемент в списке Python необходимо использовать оператор `in`.

**7. Как определить число вхождений заданного элемента в списке?**

`list.count('элемент')`

**8. Как осуществляется добавление (вставка) элемента в список?**

Метод `insert` можно использовать, чтобы вставить элемент в список.

**9. Как выполнить сортировку списка?**

`list.sort()`

**10. Как удалить один или несколько элементов из списка?**

Удалить элемент можно, написав его индекс в методе `pop`.

**11. Что такое списковое включение и как с его помощью осуществлять обработку списков?**

List Comprehensions чаще всего на русский язык переводят как абстракция списков или списковое включение, является частью синтаксиса языка, которая предоставляет простой способ построения списков.

**12. Как осуществляется доступ к элементам списков с помощью срезов?**

`list[<начало среза>:<конец среза>:<шаг>]`

**13. Какие существуют функции агрегации для работы со списками?**

Для работы со списками Python предоставляет следующие функции:

- `len(L)` - получить число элементов в списке `L`.
- `min(L)` - получить минимальный элемент списка `L`.
- `max(L)` - получить максимальный элемент списка `L`.
- `sum(L)` - получить сумму элементов списка `L`, если список `L`

содержит только числовые значения

**14. Как создать копию списка?**

Для создания копии списка необходимо использовать либо метод `copy`, либо использовать оператор среза

**15. Самостоятельно изучите функцию `sorted` языка Python. В чем ее отличие от метода `sort` списков?**

Отличие заключается в том, что метод `list.sort()` определён только для списков, в то время как `sorted()` работает со всеми итерируемыми объектами.