

**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 №2.4**

**Дисциплина: «Основы кроссплатформенного программирования»**

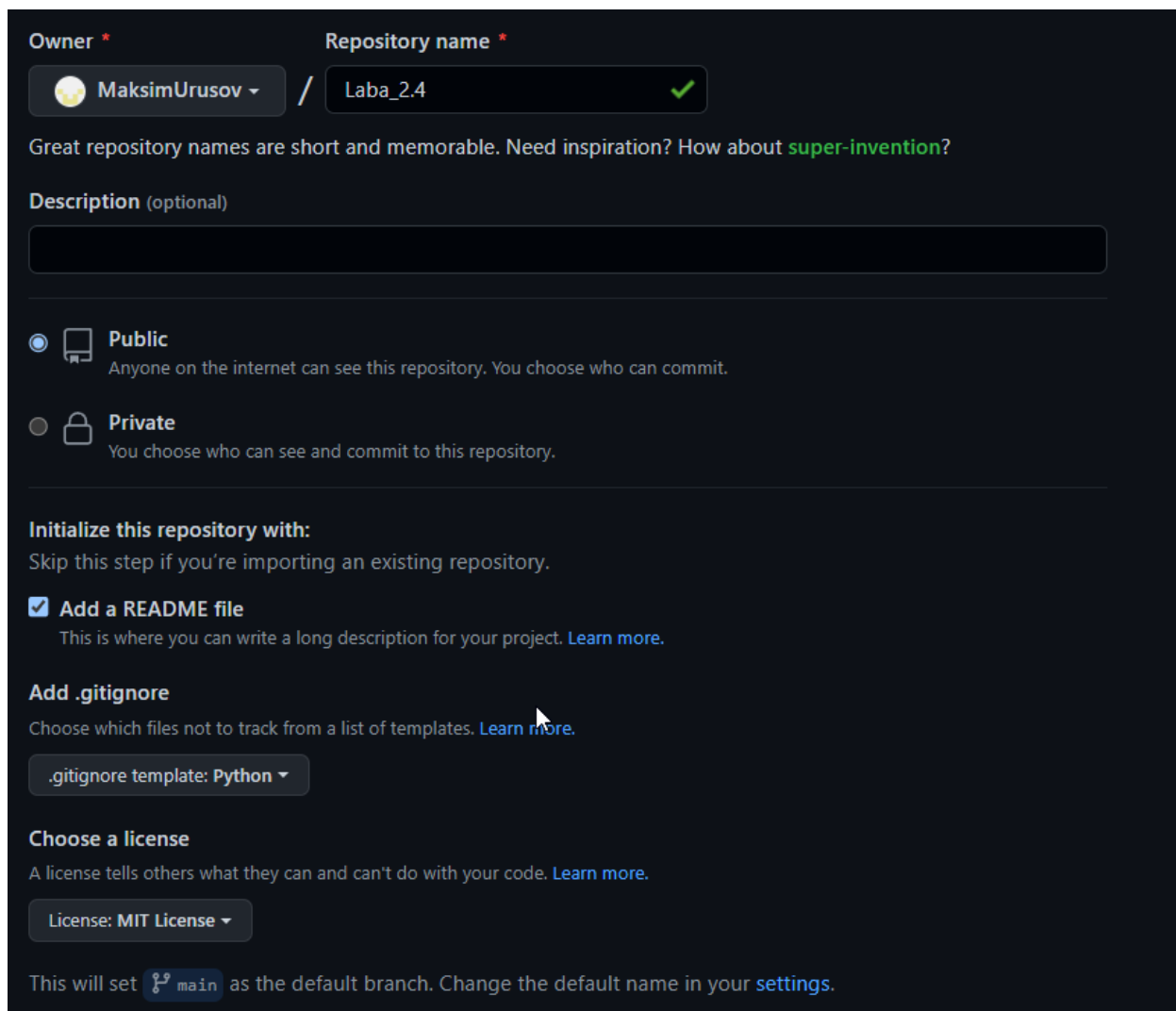
**Тема: «Работа со списками в языке Python»**

**Выполнил: студент 1  
курса  
группы ИВТ-б-о-21-1  
Урусов Максим  
Андреевич**

**Ставрополь 2022**

## Выполнение работы:

1. Создал репозиторий в GitHub «rep 2.4» 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.



The screenshot shows the GitHub repository creation interface. At the top, the 'Owner' is 'MaksimUrusov' and the 'Repository name' is 'Laba\_2.4', which is marked as valid with a green checkmark. Below this, there is a suggestion for repository names and a 'Description' field. The 'Public' option is selected, indicating that anyone on the internet can see the repository. Under 'Initialize this repository with:', the 'Add a README file' checkbox is checked. The 'Add .gitignore' section shows a dropdown menu with 'Python' selected. The 'Choose a license' section shows 'MIT License' selected. At the bottom, it states that 'main' will be set as the default branch.

Рисунок 1.1 Создание репозитория

```
Cloning into 'rep_2.4'...
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:\Users\den-n\L\Laba-2.4>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/den-n/L/Laba-2.4/.git/hooks]

```

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

2. Создание проект PyCharm в папке репозитория, проработал примеры ЛР.

```

1 5 4 6 8 7 9 5 1 3
9

```

Рисунок2.1 Результат работы 1 примера

```

2 3 6 5 4 8 9 4 8 7
13

```

Рисунок2.2 Результат работы 2 примера

3. (16 вариант). Выполнил 2 индивидуальных задания.

```

import sys

if __name__ == '__main__':
    A = list(map(int, input().split()))
    if not A:
        print("Заданный список пуст", file=sys.stderr)
        exit(1)

    pol = 0

    for id, val in enumerate(A):
        if val > 0:
            pol += val

    sum = 0

    for i in range(len(A) - 1, 0, -1):
        if A[i] != 0:
            sum += A[i]
        else:
            break
    print(pol, sum)

```

Рисунок 3.1 первая программа

```

1 2 3 4 5 6 7 8 9 10
Сумма равна - 30
Кол-во равно - 5

```

Рисунок 3.1 Результат работы 1 программы

```

import sys

if __name__ == '__main__':
    A = list(map(int, input().split()))
    if not A:
        print("Заданный список пуст", file=sys.stderr)
        exit(1)

    pol = 0

    for id, val in enumerate(A):
        if val > 0:
            pol += val

    sum = 0

    for i in range(len(A) - 1, 0, -1):
        if A[i] != 0:
            sum += A[i]
        else:
            break
    print(pol, sum)

```

Рисунок 3.3 вторая программа

```

C:\Users\den-n\L\Laba-2.4\prim\Scripts\python.exe C:/Users/den-n/L/Laba-2.4/ind/ind_2.py
1 -1 2 -2 3 -3 4 -4 0 1 2 3 4
20 10

Process finished with exit code 0
|

```

Рисунок 3.4 Результат работы второй программы

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

```

[develop 3356fc8] added programs
1 file changed, 150 insertions(+), 3 deletions(-)

C:\Users\adamk\OneDrive\Рабочий стол\rep_2.4>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\den-n\L\Laba-2.4>git branch
  develop
* main
C:\Users\den-n\L\Laba-2.4>git merge develop
Updating a196389..f60defa
Fast-forward
 .idea/.gitignore      | 3 +++
 .idea/Laba-2.4.iml    |10 +++++++
 .idea/inspectionProfiles/profiles_settings.xml | 6 +++++
 .idea/misc.xml        | 4 +++
 .idea/modules.xml     | 8 ++++++
 .idea/vcs.xml         | 6 +++++
 ind/ind_1.py          |22 ++++++++++++++++++
 ind/ind_2.py          |28 +++++++++++++++++++++
 prim/prim_1.py        |20 ++++++++
 prim/prim_2.py        |34 +++++++++++++++++++++
10 files changed, 141 insertions(+)
create mode 100644 .idea/.gitignore
create mode 100644 .idea/Laba-2.4.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 ind/ind_1.py
create mode 100644 ind/ind_2.py
create mode 100644 prim/prim_1.py
create mode 100644 prim/prim_2.py
C:\Users\den-n\L\Laba-2.4>

```

Рисунок 4.2 Слияние веток main и develop

```

D:\git\Laba_2.4>git push
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 8 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (13/13), 4.84 KiB | 4.84 MiB/s, done.
Total 13 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/LenkaGolovach/Laba_2.4.git
 1c5f9d6..eb73e21  main -> main

```

Рисунок 4.3 Пуш изменений на удалённый сервер

**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.