

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 №2\_4**  
**дисциплины**  
**«Основы программной инженерии»**

Выполнил:

Разворотников Денис Сергеевич  
2 курс, группа ПИЖ-б-о-21-1,  
09.03.04 «Программная инженерия»,  
направленность (профиль) «Разработка  
и сопровождение программного  
обеспечения», очная форма обучения

---

(подпись)

Проверил:

---

(подпись)

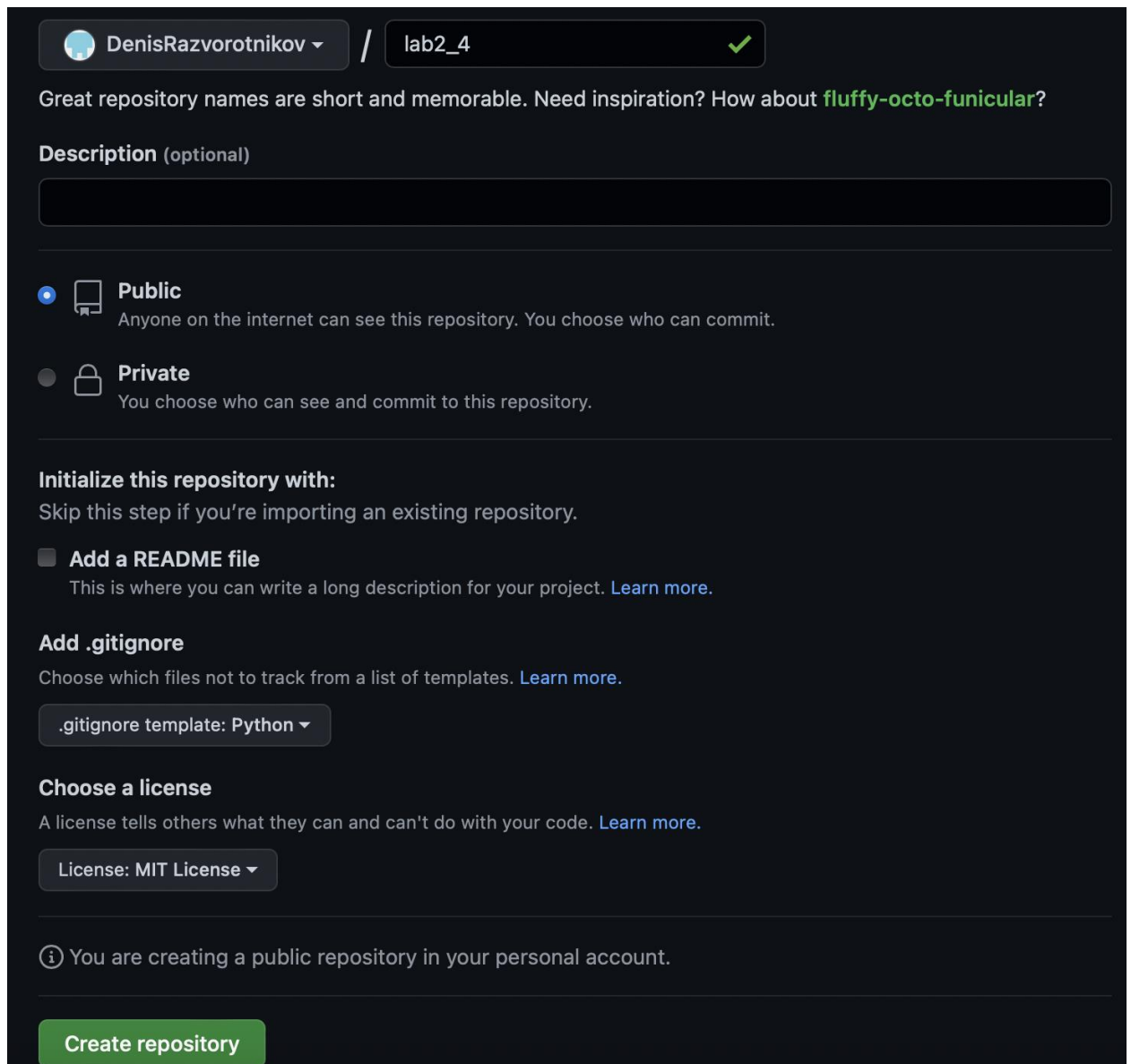
Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь 2022

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

Ход работы:

1. Был создан репозиторий в Github в который были добавлены правила gitignore для работы IDE PyCharm, была выбрана лицензия MIT, сам репозиторий был клонирован на локальный сервер и был организован в соответствии с моделью ветвления git-flow.



The screenshot shows the GitHub 'Create repository' form. At the top, the user 'DenisRazvorotnikov' is selected, and the repository name 'lab2\_4' is entered with a green checkmark. Below this, a tip suggests repository names should be short and memorable, with an example 'fluffy-octo-funicular?'. The 'Description' field is empty. Under 'Visibility', the 'Public' option is selected, indicating that anyone on the internet can see the repository. The 'Private' option is also visible. The 'Initialize this repository with:' section has a checkbox for 'Add a README file'. Below this, there is a section for 'Add .gitignore' with a dropdown menu showing '.gitignore template: Python'. The 'Choose a license' section has a dropdown menu showing 'License: MIT License'. At the bottom, a green button labeled 'Create repository' is visible. A note at the bottom states: 'You are creating a public repository in your personal account.'

Рисунок 1.1 – Создание репозитория

```
Cloning into 'lab2_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.
denisrazvorotnikov@Air-Denis desktop %
```

Рисунок 1.2 – Клонирование репозитория

2. Был создана папка PyCharm в которой хранятся примеры из лабораторной работы.

```
/Users/denisrazvorotnikov/PY/bin/python /U
1 3 5 6 3 5 6 3 6 7
10

Process finished with exit code 0
```

Рисунок 2.1 – Результат работы первого примера

```
/Users/denisrazvorotnikov/PY/bin/pyt
5 -3 4 5 8 7 -5
1

Process finished with exit code 0
```

Рисунок 2.2 – Результат работы второго примера

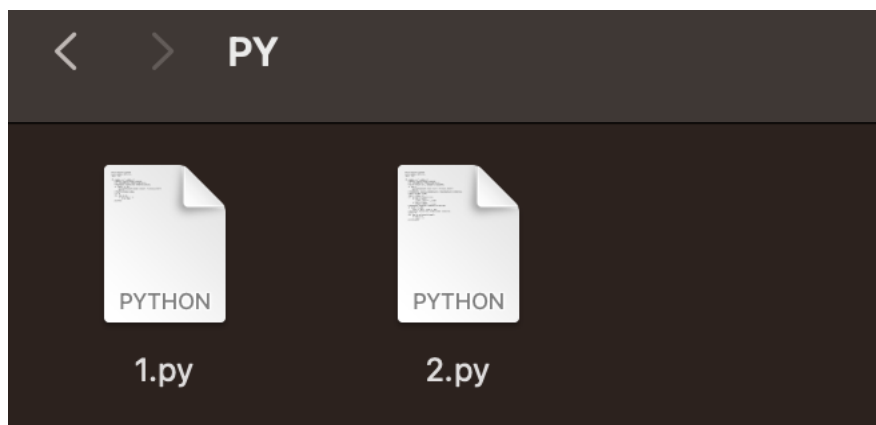


Рисунок 2.3 – Папка с примерами из лабораторной работы

3. Было выполнено два индивидуальных задания согласно 5 варианту

#### Задание 1

Определить средний рост девочек и мальчиков одного класса. В классе учится  $n$  учеников.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import sys

if __name__ == "__main__":
    n = int(input("Общее кол-во учеников = "))
    m = int(input("Кол-во мальчиков = "))
    d = n - m
    sm = 0
    sd = 0
    rm = list(map(float, input('Рост мальчиков: ').split()))
    rd = list(map(float, input('Рост девочек: ').split()))
    if len(rm) == m:
        for i, item in enumerate(rm):
            sm += item
    else:
        print(
            'Неверное кол-во',
            file=sys.stderr
        )
    if len(rd) == d:
        for i, item in enumerate(rd):
            sd += item
    else:
        print(
            'Неверное кол-во',
            file=sys.stderr
        )
    r_sr_m = sm / m
    r_sr_d = sd / d
    print('Средний рост мальчиков: {:.1f}\nСредний рост девочек: {:.1f}'.format(r_sr_m, r_sr_d))
```

```
/Users/denisrazvorotnikov/PY/bin/python
Общее кол-во учеников = 5
Кол-во мальчиков = 3
Рост мальчиков: 120 142 150
Рост девочек: 154 134
Средний рост мальчиков: 137.3
Средний рост девочек: 144.0

Process finished with exit code 0
```

Рисунок 3.1 – Результат работы программы

## Задание 2

В списке, состоящем из вещественных элементов, вычислить:

1. максимальный элемент списка;
2. сумму элементов списка, расположенных до последнего положительного элемента.

Сжать список, удалив из него все элементы, модуль которых находится в интервале  $[a, b]$ .

Освободившиеся в конце списка элементы заполнить нулями.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import sys

if __name__ == '__main__':
    a = list(map(float, input().split()))
    a1 = list(range(0, len(a)))
    in1 = int(input("range 1: "))
    in2 = int(input("range 2: "))
    max_ = a[0]
    pos = 0
    for i in a:
        if i > max_:
            max_ = i
    for i in a:
        if i > 0:
            pos += i
        else:
            break
    for i, item in enumerate(a):
        if (item >= in1) and (item <= in2):
            a.remove(item)
            a.append(0)
    for i, item in enumerate(a):
        print(item)
    print('\nmaximum is: ', max_, 'sum is', pos)
```

```
/Users/denisrazvorotnikov/PY/bin/python
1.2 2.2 5.1 2.4 -3.2 6.7
range 1: 2
range 2: 4
1.2
5.1
-3.2
6.7
0
0

maximum is: 6.7 sum is 10.9
```

Рисунок 3.2 – Результат работы программы

4. Был осуществлен коммит и слияние веток main и develop, также были запущены изменения на удаленный сервер.

```
[denisrazvorotnikov@Air-Denis lab2_4 % git add .
[denisrazvorotnikov@Air-Denis lab2_4 % git commit -m "Add files"
[develop 354204f] Add files
10 files changed, 138 insertions(+)
create mode 100644 PY/.idea/.gitignore
create mode 100644 PY/.idea/PY.iml
create mode 100644 PY/.idea/inspectionProfiles/profiles_settings.xml
create mode 100644 PY/.idea/misc.xml
create mode 100644 PY/.idea/modules.xml
create mode 100644 PY/.idea/vcs.xml
create mode 100644 PY/1.py
create mode 100644 PY/2.py
create mode 100644 PY/idz1.py
create mode 100644 PY/idz2.py
[denisrazvorotnikov@Air-Denis lab2_4 % git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
denisrazvorotnikov@Air-Denis lab2_4 %
```

Рисунок 4.1 – Коммит изменений и переход на ветку main

```

[denisrazvorotnikov@Air-Denis lab2_4 % git merge develop
Updating bc644bb..354204f
Fast-forward
 PY/.idea/.gitignore | 3 +++
 PY/.idea/PY.iml | 8 ++++++
 PY/.idea/inspectionProfiles/profiles_settings.xml | 6 +++++
 PY/.idea/misc.xml | 7 ++++++
 PY/.idea/modules.xml | 8 ++++++
 PY/.idea/vcs.xml | 6 +++++
 PY/1.py | 17 ++++++++
 PY/2.py | 28 ++++++++
 PY/idz1.py | 29 ++++++++
 PY/idz2.py | 26 ++++++++
10 files changed, 138 insertions(+)
create mode 100644 PY/.idea/.gitignore
create mode 100644 PY/.idea/PY.iml
create mode 100644 PY/.idea/inspectionProfiles/profiles_settings.xml
create mode 100644 PY/.idea/misc.xml
create mode 100644 PY/.idea/modules.xml
create mode 100644 PY/.idea/vcs.xml
create mode 100644 PY/1.py
create mode 100644 PY/2.py
create mode 100644 PY/idz1.py
create mode 100644 PY/idz2.py
[denisrazvorotnikov@Air-Denis lab2_4 % git push git@github.com:DenisRazvorotnikov
/lab2_4.git
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), 3.12 KiB | 1.56 MiB/s, done.
Total 15 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:DenisRazvorotnikov/lab2_4.git
 bc644bb..354204f main -> main
denisrazvorotnikov@Air-Denis lab2_4 % █

```

Рисунок 4.2 – Слияние веток и пуш изменений на удаленный сервер

Вывод: были приобретены навыки по работе со списками при написании программ с помощью языка программирования Python версии 3.x.

### Контрольные вопросы

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.