

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 №10**  
**дисциплины «Основы программной инженерии»**

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

---

(подпись)

Доцент кафедры инфокоммуникаций  
Воронкин Роман Александрович

---

(подпись)

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

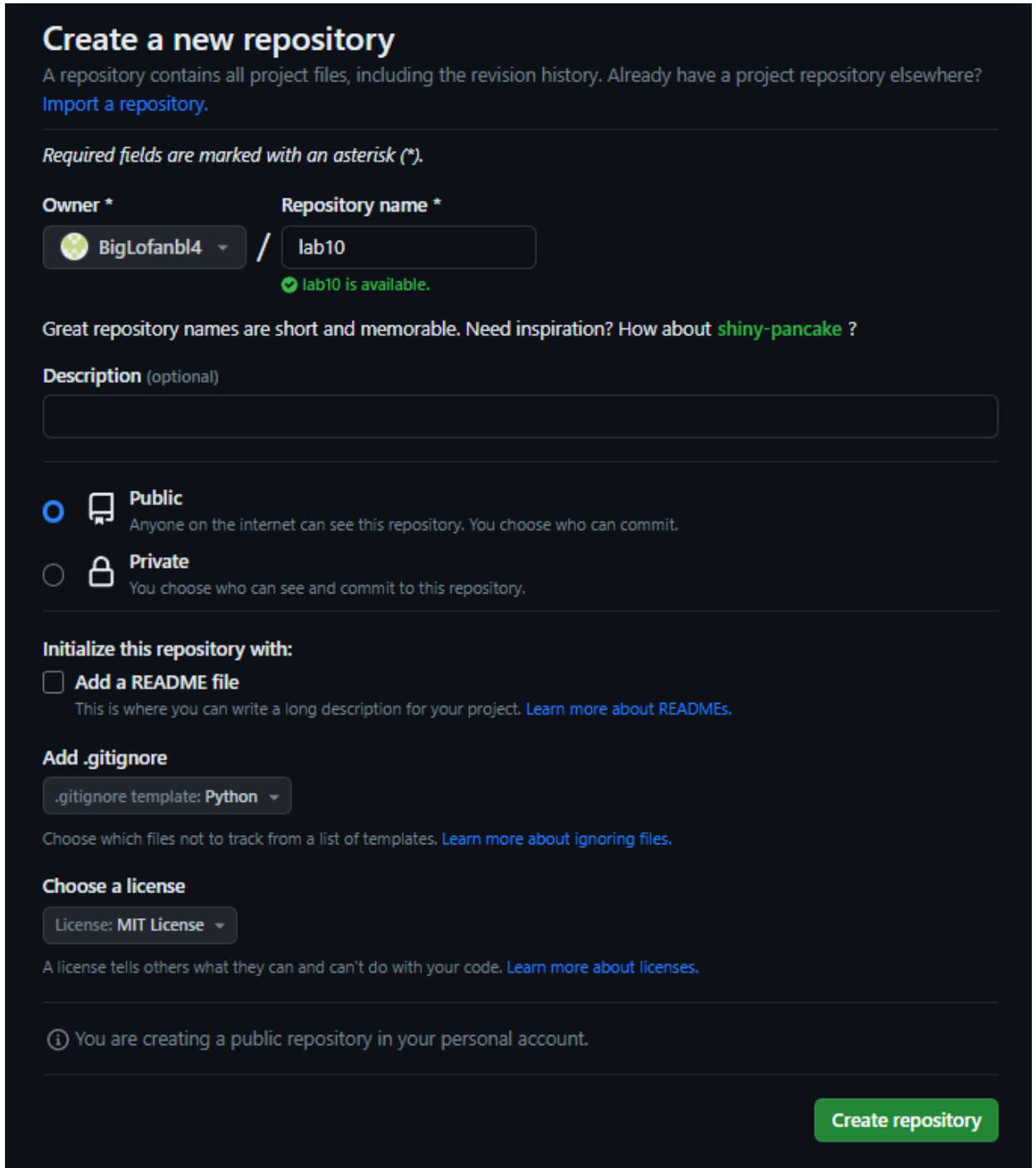
Ставрополь, 2023 г.

**Тема:** работа с множествами в языке Python.

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

Порядок выполнения работы

1. Создание репозитория.



The image shows the 'Create a new repository' page on GitHub. The page has a dark theme. At the top, it says 'Create a new repository' in white. Below that, a subtitle explains that a repository contains all project files and revision history. There's a link to 'Import a repository'. A note states 'Required fields are marked with an asterisk (\*)'. The 'Owner' field is set to 'BigLofanbl4' and the 'Repository name' field is 'lab10'. A green checkmark indicates 'lab10 is available'. Below this, there's a suggestion for repository names: 'Great repository names are short and memorable. Need inspiration? How about shiny-pancake?'. The 'Description' field is optional and empty. There are two radio button options: 'Public' (selected) and 'Private'. Under 'Initialize this repository with:', there's an unchecked checkbox for 'Add a README file'. Below that, there's a section for 'Add .gitignore' with a dropdown menu set to 'Python'. Then, there's a 'Choose a license' section with a dropdown menu set to 'MIT License'. At the bottom, there's an information icon and a message: 'You are creating a public repository in your personal account.' A green 'Create repository' button is at the bottom right.

**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 \*** **Repository name \***

BigLofanbl4 / lab10

lab10 is available.

Great repository names are short and memorable. Need inspiration? How about shiny-pancake ?

**Description** (optional)

☐ **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.](#)

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

**Create repository**

Рисунок 1 – Создание репозитория

## 2. Примеры из лабораторной работы.

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

if __name__ == "__main__":
    u = set("abcdefghijklmnopqrstuvwxyz")

    a = {"b", "c", "h", "o"}
    b = {"d", "f", "g", "o", "v", "y"}
    c = {"d", "e", "j", "k"}
    d = {"a", "b", "f", "g"}

    x = (a.intersection(b)).union(c)
    print(f"x = {x}")

    # Найдем дополнения множеств
    bn = u.difference(b)
    cn = u.difference(c)

    y = (a.difference(d)).union(cn.difference(bn))
    print(f"y = {y}")
```

Рисунок 2.1 – Код примера

```
x = {'j', 'e', 'k', 'o', 'd'}
y = {'g', 'v', 'o', 'y', 'f', 'c', 'h'}
```

Рисунок 2.2 – Вывод программы из примера

3. Задание 1: подсчитайте количество гласных в строке, введенной с клавиатуры с использованием множеств.

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

if __name__ == "__main__":
    vowel = set("ауоиэыяюеё")
    user_input = input("Введите текст: ")
    c = len([i for i in user_input.lower() if i in vowel])

    # c = 0
    # for i in user_input.lower():
    #     if i in vowel:
    #         c += 1

    print(f"Количество гласных букв = {c}")
```

Рисунок 3.1 – Код программы

```
Введите текст: А вот бы сходить на речку!
Количество гласных букв = 8
```

Рисунок 3.2 – Вывод программы

4. Задание 2: определите общие символы в двух строках, введенных с клавиатуры.

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

if __name__ == "__main__":
    str1 = set(input("Введите первую строку: "))
    str2 = set(input("Введите вторую строку: "))

    print(
        f"Общие символы в введенных строках {str1.intersection(str2)}"
    )
```

Рисунок 4.1 – Код программы

```
Введите первую строку: Привет мир!
Введите вторую строку: Пока мир!
Общие символы в введенных строках {' ', 'м', '!', 'р', 'и', 'П'}
```

Рисунок 4.2- Вывод программы

## 5. Индивидуальное задание вариант 14.

14.

$$X = (A \cup B) \cap C; \quad Y = (\bar{A} \cap D) \cup (C/B).$$

$$A = \{c, m, n, o, q\}; \quad B = \{c, d, m, w\}; \quad C = \{m, n, q\}; \quad D = \{c, m, p\};$$

Рисунок 5.1 – Условие задания

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

# 14 вариант

if __name__ == "__main__":
    u = set("abcdefghijklmnopqrstuvwxyz")

    a = {"c", "m", "n", "o", "q"}
    b = {"c", "d", "m", "w"}
    c = {"m", "n", "q"}
    d = {"c", "m", "p"}

    x = (a.union(b)).intersection(c)

    an = u.difference(a)
    y = (an.intersection(d)).union(c.difference(b))

    print(f"x = {x}")
    print(f"y = {y}")
```

Рисунок 5.2 – Код программы

```
x = {'m', 'n', 'q'}
y = {'n', 'p', 'q'}
```

Рисунок 5.3 – Вывод программы

### Ответы на контрольные вопросы

1. Множеством в языке программирования Python называется неупорядоченная совокупность уникальных значений.
2. С помощью функции set либо с помощью фигурных скобок:  $a = \{ \}$  или  $a = set()$ .
3. С помощью оператора in.
4. С помощью цикла for.

5. Set comprehension - это метод создания множеств в Python, используя элементы из других итерируемых объектов.

6. С помощью метода add.

7. Удаление одного методы: remove – удаляет элемент если он есть в множестве, иначе ошибка, discard – удалят элемент, но не вызывает ошибку, pop – удаляет первый элемент множества и возвращает его, но так как множества неупорядоченная коллекция, удаляет случайны элемент и возвращает его. Удаление всех – метод clear.

8. Объединение – union, пересечение – intersection, разность – difference.

9. Подмножество – метод issubset, надмножество – метод issuperset.

10. Создание неизменяемого множества.

11. При помощи функций str, list, dict.