

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

(подпись)

Руководитель практики:
Богданов С.С., ассистент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: Работа с множествами в языке Python.

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

Порядок выполнения работы

1. Создал репозиторий GitHub.

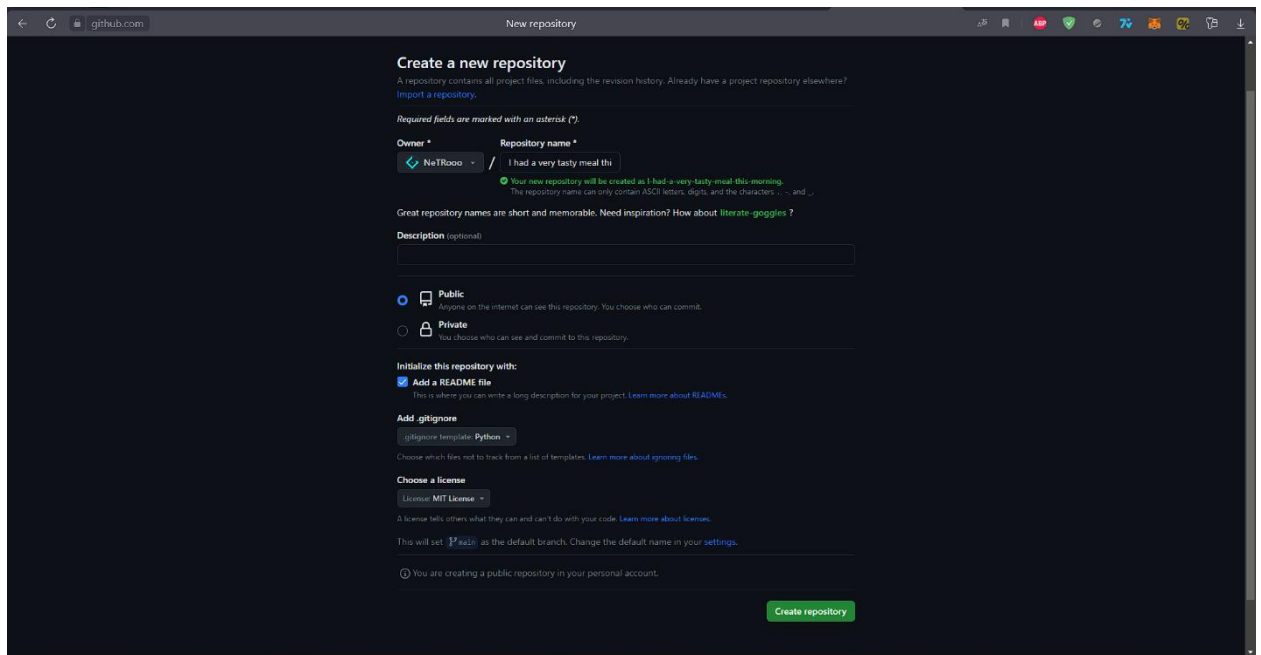
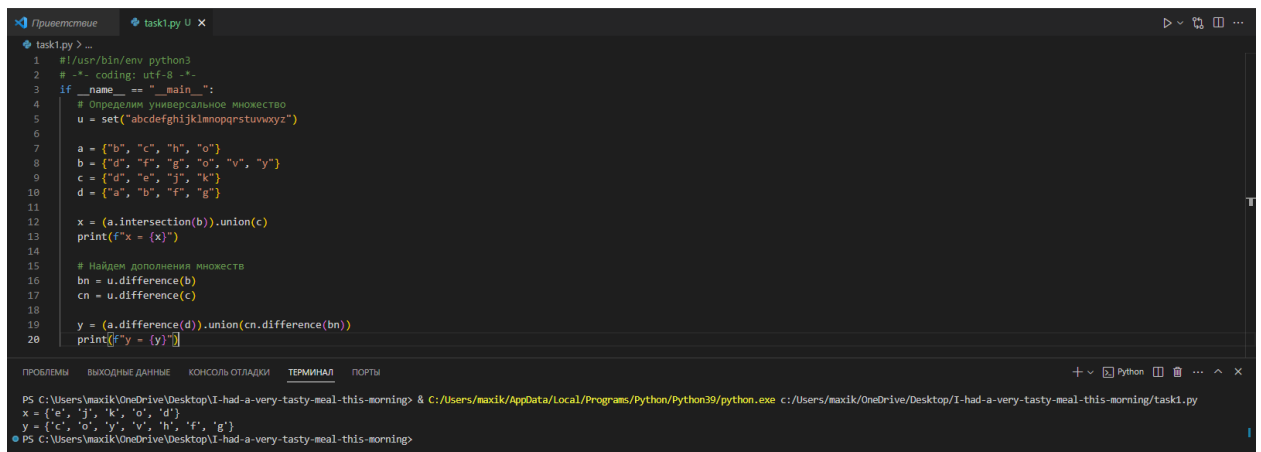


Рисунок 1- Создание репозитория

2. Проработал примеры из лабораторной работы.

```
PS C:\Users\maxik\OneDrive\Desktop> git clone https://github.com/NeTRooo/I-had-a-very-tasty-meal-this-morning.git
Cloning into 'I-had-a-very-tasty-meal-this-morning'...
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.
PS C:\Users\maxik\OneDrive\Desktop> |
```

Рисунок 2 – Выполнил клонирование созданного репозитория



```
1 #!/usr/bin/env python3
2 #-*- coding: utf-8 -*-
3 if __name__ == "__main__":
4     # Определим универсальное множество
5     u = set("abcdefghijklmnopqrstuvwxyz")
6
7     a = {"b", "c", "h", "o"}
8     b = {"d", "f", "g", "o", "v", "y"}
9     c = {"d", "o", "j", "k"}
10    d = {"a", "b", "f", "g"}
11
12    x = (a.intersection(b)).union(c)
13    print(f"x = {x}")
14
15    # Найдем дополнения множеств
16    bn = u.difference(b)
17    cn = u.difference(c)
18
19    y = (a.difference(d)).union(cn.difference(bn))
20    print(f"y = {y}")
```

PS C:\Users\maxik\OneDrive\Desktop\I-had-a-very-tasty-meal-this-morning> & C:/Users/maxik/AppData/Local/Programs/Python/Python39/python.exe c:/Users/maxik/OneDrive/Desktop/I-had-a-very-tasty-meal-this-morning/task1.py

x = {'c', 'j', 'k', 'o', 'd'}

y = {'c', 'o', 'y', 'v', 'h', 'f', 'g'}

PS C:\Users\maxik\OneDrive\Desktop\I-had-a-very-tasty-meal-this-morning>

Рисунок 3 – Определить результат выполнений операция над множествами (задание №1)



```
1 #!/usr/bin/env python3
2 #-*- coding: utf-8 -*-
3 def count_vowels(input_string):
4     vowels = set("aeiouAEIOU")
5     vowel_count = 0
6     for char in input_string:
7         if char in vowels:
8             vowel_count += 1
9     return vowel_count
10
11 if __name__ == "__main__":
12     input_string = input("Введите строку: ")
13
14     result = count_vowels(input_string)
15     print("Количество гласных в строке:", result)
16
```

PS C:\Users\maxik\OneDrive\Desktop\I-had-a-very-tasty-meal-this-morning> & C:/Users/maxik/AppData/Local/Programs/Python/Python39/python.exe c:/Users/maxik/OneDrive/Desktop/I-had-a-very-tasty-meal-this-morning/ind1.py

Введите строку: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore

Количество гласных в строке: 35

PS C:\Users\maxik\OneDrive\Desktop\I-had-a-very-tasty-meal-this-morning>

Рисунок 4 – Решите задачу: подсчитайте количество гласных в строке, введенной с клавиатуры с использованием множеств. (задание №2)



```
1 #!/usr/bin/env python3
2 #-*- coding: utf-8 -*-
3 if __name__ == "__main__":
4     string1 = input("Enter the first string: ")
5     string2 = input("Enter the second string: ")
6
7     set_of_chars1 = set(string1)
8     set_of_chars2 = set(string2)
9
10    common_chars = set_of_chars1.intersection(set_of_chars2)
11
12    if common_chars:
13        print("Common characters:", common_chars)
14    else:
15        print("No common characters.")
```

PS C:\Users\maxik\OneDrive\Desktop\I-had-a-very-tasty-meal-this-morning> & C:/Users/maxik/AppData/Local/Programs/Python/Python39/python.exe c:/Users/maxik/OneDrive/Desktop/I-had-a-very-tasty-meal-this-morning/ind2.py

Enter the first string: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore

Enter the second string: et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud

Common characters: {'e', 'o', 's', 'a', 'g', 'a', 'i', 'n', 't', 'l', 'd', 'r', 'u', 't'}

PS C:\Users\maxik\OneDrive\Desktop\I-had-a-very-tasty-meal-this-morning>

Рисунок 5 – Решите задачу: определите общие символы в двух строках, введенных с клавиатуры. (задание №3)

13. $X = (A/B) \cap (C \cup D); \quad Y = (A \cap B)/(C \cup D).$
14. $A = \{b, e, g, h, k, s\}; \quad B = \{c, g, p, q\}; \quad C = \{f, g, s, x, y, z\}; \quad D = \{a, c, d, g, u, v, z\};$
14. $X = (A \cup B) \cap C; \quad Y = (\bar{A} \cap D) \cup (C/B).$ (15)

Рисунок 6 – Условия задачи

```

1 #!/usr/bin/env python3
2 #-*- coding: utf-8 -*-
3 if __name__ == "__main__":
4     a = {'c', 'm', 'n', 'o', 'g'}
5     b = {'c', 'g', 'p', 'q'}
6     c = {'f', 'g', 's', 'x', 'y', 'z'}
7     d = {'a', 'c', 'd', 'g', 'u', 'v', 'z'}
8     all_ = a.union(b).union(c).union(d)
9
10    x = (a.union(b)).intersection(c)
11    y = (all_.difference(a).intersection(d)).union(c.difference(d))
12
13    print(f"x = {x}")
14    print(f"y = {y}")
  
```

```

PS C:\Users\maxik\OneDrive\Desktop\I-had-a-very-tasty-meal-this-morning> & C:\Users\maxik\AppData\Local\Programs\Python\Python39\python.exe c:\Users\maxik\OneDrive\Desktop\I-had-a-very-tasty-meal-this-morning\ind3.py
x = {'g'}
y = {'d', 'z', 'x', 'a', 'y', 'u', 'v', 'f', 's'}
PS C:\Users\maxik\OneDrive\Desktop\I-had-a-very-tasty-meal-this-morning>
  
```

Рисунок 7 – Решите задачу: определите общие символы в двух строках, введенных с клавиатуры. (инд. задание №1)

```

PS C:\Users\maxik\OneDrive\Desktop\I-had-a-very-tasty-meal-this-morning> git add .
PS C:\Users\maxik\OneDrive\Desktop\I-had-a-very-tasty-meal-this-morning> git commit -m "last commit"
[dev 81fb2e2] last commit
3 files changed, 58 insertions(+)
create mode 100644 ind1.py
create mode 100644 ind2.py
create mode 100644 task1.py
PS C:\Users\maxik\OneDrive\Desktop\I-had-a-very-tasty-meal-this-morning> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\maxik\OneDrive\Desktop\I-had-a-very-tasty-meal-this-morning> git merge dev
Updating 1e133c4..81fb2e2
Fast-forward
 ind1.py | 15 +++++
 ind2.py | 15 +++++
 task1.py | 20 +++++
 3 files changed, 58 insertions(+)
 create mode 100644 ind1.py
 create mode 100644 ind2.py
 create mode 100644 task1.py
PS C:\Users\maxik\OneDrive\Desktop\I-had-a-very-tasty-meal-this-morning> git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.20 KiB | 1.20 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/NeTRooo/I-had-a-very-tasty-meal-this-morning.git
 1e133c4..81fb2e2 main -> main
PS C:\Users\maxik\OneDrive\Desktop\I-had-a-very-tasty-meal-this-morning> git checkout dev
Switched to branch 'dev'
Your branch is ahead of 'origin/dev' by 1 commit.
(use "git push" to publish your local commits)
PS C:\Users\maxik\OneDrive\Desktop\I-had-a-very-tasty-meal-this-morning> git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/NeTRooo/I-had-a-very-tasty-meal-this-morning.git
 1e133c4..81fb2e2 dev -> dev
PS C:\Users\maxik\OneDrive\Desktop\I-had-a-very-tasty-meal-this-morning>
  
```

Рисунок 8 – Создание и отправка коммита

Вывод: в ходе выполнения лабораторной работы были получены навыки работы с множествами в языке Python.

Ссылка на репозиторий: <https://github.com/NeTRooo/I-had-a-very-tasty-meal-this-morning>

Ответы на контрольные вопросы

1. Что такое списки в языке Python?

Списки в языке Python - это упорядоченные изменяемые коллекции элементов.

2. Каково назначение кортежей в языке Python?

Кортежи в языке Python используются для создания неизменяемых упорядоченных коллекций элементов.

3. Как осуществляется создание кортежей?

Кортежи создаются с использованием круглых скобок, например, `my_tuple = (1, 2, 3)`.

4. Как осуществляется доступ к элементам кортежа?

Доступ к элементам кортежа осуществляется по индексу, например, `element = my_tuple[0]`.

5. Зачем нужна распаковка (деструктуризация) кортежа?

Распаковка кортежа (деструктуризация) позволяет присваивать значения элементам кортежа одной строкой.

6. Какую роль играют кортежи в множественном присваивании?

Кортежи играют ключевую роль в множественном присваивании, где значения присваиваются сразу нескольким переменным.

7. Как выбрать элементы кортежа с помощью среза?

Выбор элементов кортежа с помощью среза осуществляется, например, `subset = my_tuple[1:3]`.

8. Как выполняется конкатенация и повторение кортежей?

Конкатенация кортежей выполняется оператором `+`, а повторение - оператором `*`.

9. Как выполняется обход элементов кортежа?

Обход элементов кортежа осуществляется с использованием цикла, например, `for item in my_tuple:`.

10. Как проверить принадлежность элемента кортежу.

Принадлежность элемента кортежу можно проверить с использованием оператора `in`

11. Какие методы работы с кортежами Вам известны?

Некоторые методы работы с кортежами включают `count()` для подсчета элементов и `index()` для поиска индекса элемента.

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

Да, функции агрегации, такие как `len()`, `sum()`, могут использоваться с кортежами.

13. Как создать кортеж с помощью спискового включения?

Кортеж можно создать с помощью спискового включения, например, `my_tuple = tuple(x for x in my_list)`.