

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 «Программирование на Python»
Вариант 5.

Выполнила:
Михеева Елена Александровна
2 курс, группа ИВТ-б-з-20-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной
техники и автоматизированных
систем», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А., канд. техн. наук,
доцент, доцент 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 с множествами при написании программ с помощью языка программирования Python версии 3.x.

Порядок выполнения работы.

1. Был проработан примеры лабораторной работы.

```
x = {'j', 'k', 'o', 'e', 'd'}
y = {'v', 'h', 'c', 'o', 'y', 'g', 'f'}
(base) mikheeva@MacBook-Pro-Elena Python_2.7 %
```

Рисунок 1. Выполнение программы примера 1

2. Было выполнено задание: подсчитайте количество гласных в строке, введённой с клавиатуры с использованием множеств.

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  def count_vowels(input_string):
6      vowels = {'a', 'e', 'i', 'o', 'u', 'A', 'E', 'I', 'O', 'U'}
7      vowel_count = input_string.intersection(vowels)
8
9      return vowel_count
10
11
12  if __name__ == "__main__":
13      user_input = input("Введите строку: ")
14      input_set = set(user_input)
15      result = count_vowels(input_set)
16      print("Количество гласных в строке:", len(result))
17      print(result)
18
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Введите строку: Hello word A
Количество гласных в строке: 3
{'A', 'e', 'o'}
(base) mikheeva@MacBook-Pro-Elena Python_2.7 %
```

Рисунок 3. Программа для решения задачи

3. Было выполнено задание: определите общие символы в двух строках, введенных с клавиатуры.

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  def common_characters(string1, string2):
6      set1 = set(string1)
7      set2 = set(string2)
8      common_chars = set1 & set2
9      return common_chars
10
11
12  if __name__ == "__main__":
13      input_str1 = input("Введите первую строку: ")
14      input_str2 = input("Введите вторую строку: ")
15
16      common_chars_set = common_characters(input_str1, input_str2)
17
18      if common_chars_set:
19          print("Общие символы в двух строках:", ', '.join(common_chars_set))
20      else:
21          print("Нет общих символов в двух строках.")
22

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Введите первую строку: mamont
Введите вторую строку: moistazing
Общие символы в двух строках: a, t, m, o, n
(base) mikheeva@MacBook-Pro-Elena Python_2.7 %

Рисунок 4. Программа для решения второй задачи

4. Приступили к выполнению индивидуального задания:

$$A = \{c, e, h, n\}; \quad B = \{e, f, k, n, x\}; \quad C = \{b, c, h, p, r, s\}; \quad D = \{b, e, g\};$$

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

(6)

5.

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == "__main__":
5      u = set("abcdefghijklmnopqrstuvwxyz")
6      a = {"c", "e", "h", "n"}
7      b = {"e", "f", "k", "n", "x"}
8      c = {"b", "c", "h", "p", "r", "s"}
9      d = {"b", "e", "g"}
10
11      x = (a - b) & (c | d)
12      print(f"x = {x}")
13
14      not_b = u - b
15
16      y = (a & not_b) | (c - d)
17      print(f"y = {y}")
18

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

python3: error: SDK "macosx" cannot be located
x = {'h', 'c'}
y = {'h', 'p', 's', 'r', 'c'}
(base) mikheeva@MacBook-Pro-Elena Python_2.7 %

Рисунок 5. Индивидуальное задание №1

Ответы на контрольные вопросы.

1. Что такое множества в языке Python?

В языке Python множество (set) – это неупорядоченная коллекция уникальных элементов. Множества в Python поддерживают операции объединения, пересечения, разности и другие операции, свойственные математическим множествам.

2. Как осуществляется создание множеств в Python?

В Python множества можно создать с помощью фигурных скобок {} или с помощью функции set(). Множества могут содержать только уникальные элементы, поэтому дублирующиеся элементы будут автоматически удалены при создании множества.

3. Как проверить присутствие/отсутствие элемента в множестве?

В Python можно проверить присутствие или отсутствие элемента в множестве с помощью операторов in и not in.

4. Как выполнить перебор элементов множества?

В Python можно выполнить перебор элементов множества с помощью цикла for.

5. Что такое set comprehension?

Set comprehension (генератор множества) — это способ создания множества на основе итерации и условий. Он аналогичен list comprehension и использует синтаксис с фигурными скобками {}.

6. Как выполнить добавление элемента во множество?

В Python можно выполнить добавление элемента во множество с помощью метода add().

7. Как выполнить удаление одного или всех элементов множества?

В Python можно выполнить удаление одного элемента из множества с помощью метода remove() или discard(). Например:

```
my_set.remove(2) # Удаление элемента 2
```

```
my_set.discard(4) # Попытка удаления элемента 4 (если элемента нет, то ничего не происходит)
```

Чтобы удалить все элементы из множества, можно использовать метод `clear()`.

8. Как выполняются основные операции над множествами: объединение, пересечение, разность?

В Python основные операции над множествами выполняются с помощью соответствующих методов или операторов: Объединение множеств: можно использовать метод `union()` или оператор `|`. Пересечение множеств: можно использовать метод `intersection()` или оператор `&`. Разность множеств: можно использовать метод `difference()` или оператор `-`.

9. Как определить, что некоторое множество является надмножеством или подмножеством другого множества?

В Python можно определить, что некоторое множество является надмножеством или подмножеством другого множества с помощью методов `issuperset()` и `issubset()`.

10. Каково назначение множеств `frozenset`?

Множество `frozenset` в Python является неизменяемым (`immutable`) аналогом обычного множества `set`. Это означает, что после создания `frozenset` нельзя изменять его содержимое, добавлять или удалять элементы. `frozenset` полезен, когда требуется использовать множество в качестве ключа словаря или элемента другого множества, так как `frozenset` является хешируемым (`hashable`), в отличие от обычного множества `set`.

11. Как осуществляется преобразование множеств в строку, список, словарь?

В Python можно осуществить преобразование множества в строку, список и словарь с помощью соответствующих функций и методов. Преобразование в строку: Можно использовать функцию `str()` или метод `join()` для преобразования множества в строку. Преобразование в список: Можно использовать функцию `list()` для преобразования множества в список. Преобразование в словарь: Можно использовать функцию `dict.fromkeys()` для

преобразования множества в словарь, где элементы множества станут ключами словаря.