

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Отчет о лабораторной работе №10 по дисциплине основы программной
инженерии**

Выполнила:

Емельянова Яна

Александровна, 2 курс,

группа ПИЖ-б-о-20-1,

Проверил:

Доцент кафедры инфокоммуникаций,

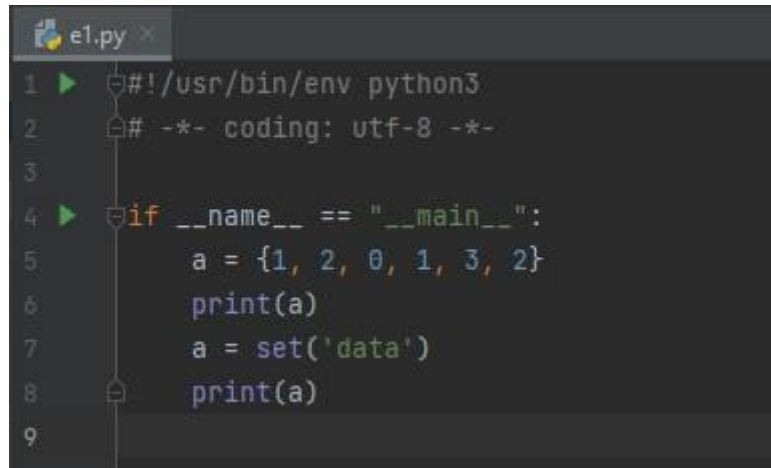
Воронкин Р.А.

Ставрополь, 2021 г.

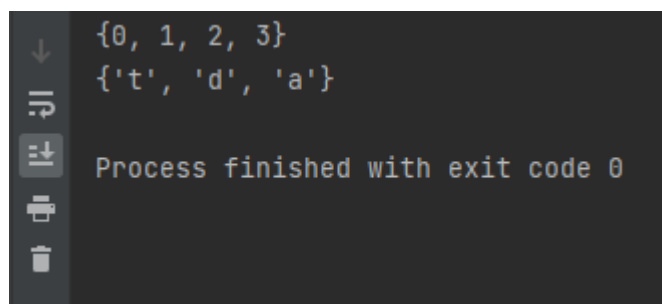
1. Работа с множествами в языке Python

Примеры из методических указаний

Создание



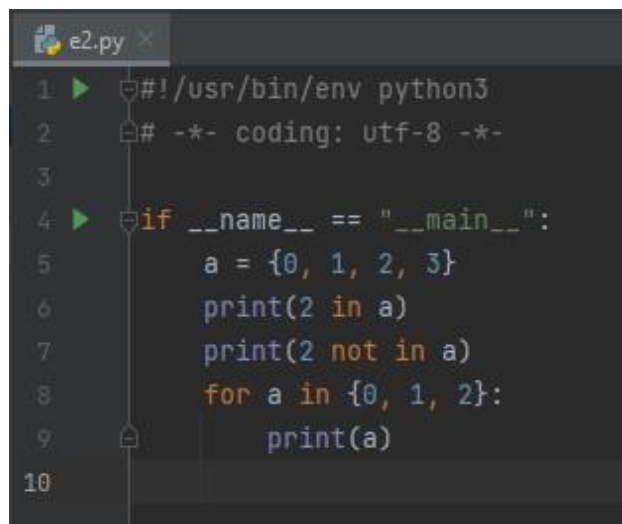
```
1  > #!/usr/bin/env python3
2  > # -*- coding: utf-8 -*-
3
4  > if __name__ == "__main__":
5      a = {1, 2, 0, 1, 3, 2}
6      print(a)
7      a = set('data')
8      print(a)
9
```



```
{0, 1, 2, 3}
{'t', 'd', 'a'}

Process finished with exit code 0
```

Использование



```
1  > #!/usr/bin/env python3
2  > # -*- coding: utf-8 -*-
3
4  > if __name__ == "__main__":
5      a = {0, 1, 2, 3}
6      print(2 in a)
7      print(2 not in a)
8      for a in {0, 1, 2}:
9          print(a)
10
```

```
True
False
0
1
2

Process finished with exit code 0
```

Set Comprehensions

```
e3.py x
1 ▶ #!/usr/bin/env python3
2   # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == "__main__":
5     a = {i for i in [1, 2, 0, 1, 3, 2]}
6     print(a)
7
```

```
{0, 1, 2, 3}

Process finished with exit code 0
```

Изменение множеств

```
e4.py x
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == "__main__":
5     a = {0, 1, 2, 3}
6     print(len(a))
7     a.add(4)
8     print(a)
9     a = {0, 1, 2, 3}
10    a.remove(3)
11    print(a)
12    a = {0, 1, 2, 3}
13    a.clear()
14    print(a)
15
```

```
↓ 4
{0, 1, 2, 3, 4}
↺ {0, 1, 2}
↻ set()
🖨
🗑 Process finished with exit code 0
```

Порядок элементов в множестве

```
e5.py x
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == "__main__":
5     a = {0, 1, 12, 'b', 'ab', 3, 2, 'a'}
6     print(a)
7     b = {0, 1, 12, 3, 2}
8     print(b)
9     print(list(b))
10
```

```
↓ {0, 1, 2, 'ab', 3, 'b', 'a', 12}  
⌵ {0, 1, 2, 3, 12}  
⇅ [0, 1, 2, 3, 12]  
🖨 Process finished with exit code 0  
🗑
```

Операции над множествами

```
е6.py ×  
1 ▶ #!/usr/bin/env python3  
2 # -*- coding: utf-8 -*-  
3  
4 ▶ if __name__ == "__main__":  
5     a = {0, 1, 2, 3}  
6     b = {4, 3, 2, 1}  
7     c = a.union(b)  
8     print(c)  
9     a.update(b)  
10    print(a)  
11    a = {0, 1, 2, 3}  
12    b = {4, 3, 2, 1}  
13    c = a.intersection(b)  
14    print(c)  
15    a = {0, 1, 2, 3}  
16    b = {4, 3, 2, 1}  
17    c = a.difference(b)  
18    print(c)  
19
```

```
↓ {0, 1, 2, 3, 4}  
⌵ {0, 1, 2, 3, 4}  
⇅ {1, 2, 3}  
🖨 {0}  
🗑 Process finished with exit code 0
```

Отношения между множествами

```

e7.py x
1 ▶ #!/usr/bin/env python3
2   # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == "__main__":
5     a = {0, 1, 2, 3, 4}
6     b = {3, 2, 1}
7     print(a.issubset(b))
8     print(a.issuperset(b))
9

```

```

↓
True
Process finished with exit code 0

```

Тип frozenset

```

e8.py x
1 ▶ #!/usr/bin/env python3
2   # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == "__main__":
5     a = frozenset({"hello", "world"})
6     print(a)
7

```

```

↓
frozenset({'hello', 'world'})
Process finished with exit code 0

```

Преобразование множеств

```
e9.py x
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == "__main__":
5     a = {'set', 'str', 'dict', 'list'}
6     b = ','.join(a)
7     print(b)
8     print(type(b))
9     a = {('a', 2), ('b', 4)}
10    b = dict(a)
11    print(b)
12    print(type(b))
13    a = {1, 2, 0, 1, 3, 2}
14    b = list(a)
15    print(b)
16    print(type(b))
17
```

```
↓ set, str, list, dict
⌵ <class 'str'>
⌵ {'a': 2, 'b': 4}
⌵ <class 'dict'>
⌵ [0, 1, 2, 3]
⌵ <class 'list'>

Process finished with exit code 0
```

1.1 Пример 1 (рис. 1, 2).

```

ex1.py x
1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4  ▶  if __name__ == "__main__":
5      # Определим универсальное множество
6      u = set("abcdefghijklmnopqrstuvwxyz")
7      a = {"b", "c", "h", "o"}
8      b = {"d", "f", "g", "o", "v", "y"}
9      c = {"d", "e", "j", "k"}
10     d = {"a", "b", "f", "g"}
11     x = (a.intersection(b)).union(c)
12     print(f"x = {x}")
13     # Найдем дополнения множеств
14     bn = u.difference(b)
15     cn = u.difference(c)
16     y = (a.difference(d)).union(cn.difference(bn))
17     print(f"y = {y}")
18

```

Рисунок 1 – Код примера

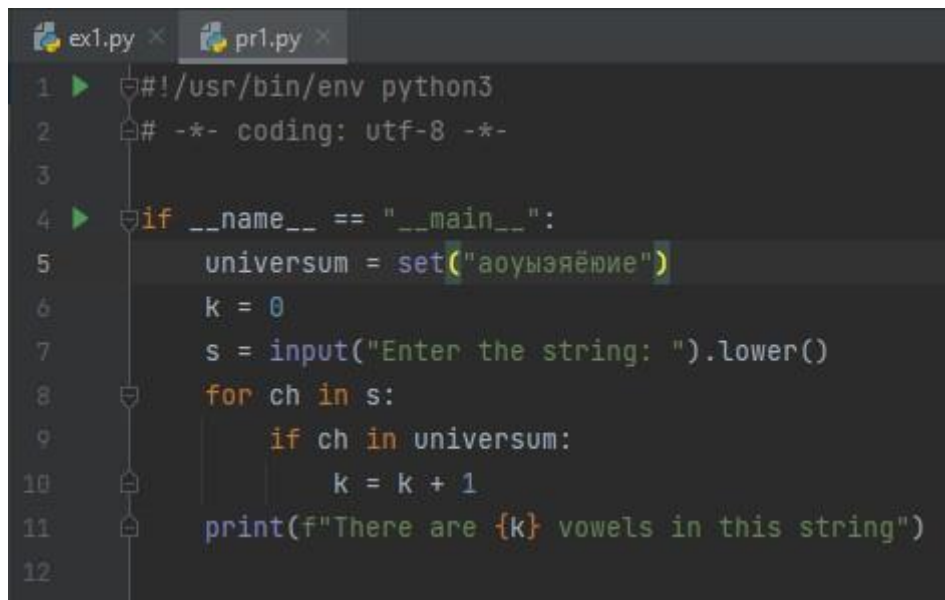
```

↓
⌵
⌵
x = {'d', 'k', 'j', 'e', 'o'}
y = {'v', 'c', 'h', 'y', 'f', 'o', 'g'}
Process finished with exit code 0

```

Рисунок 2 – Результат выполнения программы

1.2 Задание №1 (рис 3, 4 ,5).

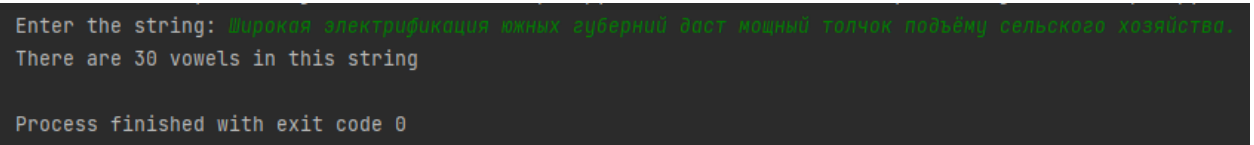


```

1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4  ▶  if __name__ == "__main__":
5      universum = set("аоуыэяёюие")
6      k = 0
7      s = input("Enter the string: ").lower()
8      for ch in s:
9          if ch in universum:
10             k = k + 1
11      print(f"There are {k} vowels in this string")
12

```

Рисунок 3 – Код программы



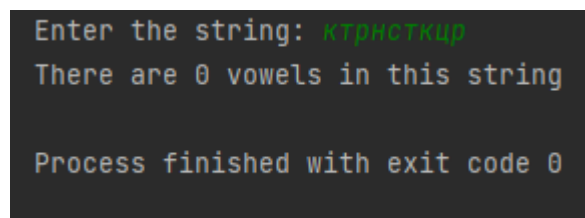
```

Enter the string: Широкая электрификация южных губерний даст мощный толчок подъёму сельского хозяйства.
There are 30 vowels in this string

Process finished with exit code 0

```

Рисунок 4 – Пример выполнения программы



```

Enter the string: ктрнсткцр
There are 0 vowels in this string

Process finished with exit code 0

```

Рисунок 5 – Вывод программы при отсутствии гласных букв

1.3 Задание №3 (рис. 6, 7).

```
ex1.py x pr1.py x pr2.py x
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == "__main__":
5     a = set(i for i in input("Enter the first string: ").lower())
6     b = set(i for i in input("Enter the second string: ").lower())
7     print(f"The intersection of these strings is: {a.intersection(b)}")
8
```

Рисунок 6 – Код программы

```
Enter the first string: французские
Enter the second string: электрификация
The intersection of these strings is: {'ц', 'и', 'ф', 'е', 'а', 'к', 'р'}
```

Рисунок 7 – Пример выполнения программы

1.3 Индивидуальное задание вариант 9

```
ind_zad.py x
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 ▶ if __name__ == "__main__":
6     u = set("abcdefghijklmnopqrstuvwxyz")
7     a = {"a", "e", "f", "i"}
8     b = {"a", "b", "k", "n"}
9     c = {"e", "f", "n", "o", "w", "x"}
10    d = {"a", "d", "e", "o", "p", "t", "u"}
11    x = (a.union(b)).intersection(d)
12    print(f"x = {x}")
13    an = u.difference(a)
14    bn = u.difference(b)
15    y = (an.intersection(bn)).difference(c.union(d))
16    print(f"y = {y}")
17
```

```
x = {'e', 'a'}
y = {'j', 'm', 'c', 'v', 'z', 'l', 'g', 's', 'q', 'y', 'r', 'h'}
Process finished with exit code 0
```

2. Ответы на контрольные вопросы

1. Множеством в языке программирования Python называется неупорядоченная совокупность уникальных значений. В качестве элементов этого набора данных могут выступать любые неизменяемые объекты, такие как числа, символы, строки.

2. Пример 1: `a = {1, 2, 0, 1, 3, 2}`

Пример 2: `a = set('data')`

3. С помощью оператора `in`.

4. С помощью цикла `for`.

5. Для создания множества можно в Python воспользоваться генератором, позволяющих заполнять списки, а также другие наборы данных с учетом неких условий.

6. С помощью метода `add()`, `a.add(<element>)`

7. Для удаления элементов из множества используются следующие функции в Python:

- `remove` — удаление элемента с генерацией исключения в случае, если такого элемента нет;
- `discard` — удаление элемента без генерации исключения, если элемент отсутствует;
- `pop` — удаление первого элемента, генерируется исключение при попытке удаления из пустого множества.
- `clear` — удаление всех элементов множества

8. Пример объединения: `c = a.union(b)`

Пример пересечения: `c = a.intersection(b)`

Пример разности: `c = a.difference(b)`

9. Определение подмножества: `a.issubset(b)`

Определение надмножества: `a.issuperset(b)`

10. Множество, содержимое которого не поддается изменению имеет тип `frozenset`. Значения из этого набора нельзя удалить, как и добавить новые. Поскольку содержимое `frozenset` должно всегда оставаться статичным, перечень функций, с которыми такое множество может взаимодействовать, имеет ограничения.

11. Для преобразования множества в строку используется конкатенация текстовых значений, которую обеспечивает функция `join`.

Пример: `b = ','.join(a)`

Чтобы получить из множества словарь, следует передать функции `dict` набор из нескольких пар значений, в каждом из которых будет находиться ключ.

Пример:

```
a = {('a', 2), ('b', 4)}
```

```
b = dict(a)
```

Для получения списка: используется вызов `list`, получающий в качестве аргумента множество `a`.

Пример:

```
a = {1, 2, 0, 1, 3, 2}
```

```
b = list(a)
```