

## Задания для самостоятельной работы обучающегося и методические рекомендации по их выполнению

### СРО №8. Словари

#### Задания:

##### 1. Задание

Разработайте 7 программ, в которых используется информация о ряде государств Европы и Азии: название, столица, часть света, численность населения (млн чел.) и площадь территории (тыс. кв. км).

#### Программы должны решать следующие задачи:

- 1) определение столицы по названию государства;
- 2) определение названия государства по городу-столице;
- 3) вывод всей информации о заданном государстве;
- 4) вывод названий всех стран, расположенных в заданной части света;
- 5) расчет плотности населения (в тыс. чел. на 1 кв. км) для каждого государства;
- 6) определение количества государств, расположенных в заданной части света;

#### Методические рекомендации к выполнению:

Элементами такой коллекции выступают пары объектов, каждая из которых включает в себя ключ и значение. Это позволяет идентифицировать ее элементы не по числовому индексу (как списки), а по произвольному, т.е. в качестве идентификатора можно использовать и число, и текстовую строку. Значениями в словаре, в отличие от ключей, могут быть объекты любого типа – числа, строки, кортежи, списки и даже другие словари.

В Python словари пишутся в фигурных скобках {}.

Создание словаря:

1) присвоить свободной переменной произвольное количество пар объектов. Элементы необходимо поместить в фигурные скобки, а между ключом и значением должен стоять символ двоеточия.

```
d = {} # создание пустого словаря
```

```
thisdict = {  
    "brand": "Ford",  
    "model":  
    "Mustang",  
    "year":  
    1964  
}
```

```
print(thisdict) # {'brand': 'Ford', 'model': 'Mustang', 'year': 1964}
```

2) использовать конструктор dict(). В таком случае быть ключом может только строка, как это показано в следующем примере кода.

```
d = dict() # создание пустого словаря  
thisdict = dict(brand="Ford", model="Mustang",  
year=1964)print(thisdict) # {'brand': 'Ford', 'model': 'Mustang',  
'year': 1964}
```

Обратите внимание, ключевые слова не являются строками, а вместо двоеточия для задания используется «равно».

Для того чтобы ввести в коллекцию новую пару объектов необходимо указать новый ключ в квадратных скобках, а также соответствующее ему значение.

```
thisdict = {  
    "brand": "Ford",  
    "model":  
    "Mustang",  
    "year":  
    1964  
}  
thisdict["color"] = "red"  
print(thisdict)    # {'brand': 'Ford', 'model': 'Mustang', 'year': 1964, 'color': 'red'}
```

Как и с другими наборами данных, производить заполнение словарей можно при помощи генераторов.

```
a = {a: a * a for a in range(5)}  
print(a)    # {0: 0, 1: 1, 2: 4, 3: 9, 4: 16}
```

Для работы со словарями доступны функции, меняющие их содержимое и выполняющие различные операции над ними. Его можно конвертировать в другие типы данных, например, в строку (потребуется функция str).

```
a = {1: "one", 2: "two", 3:  
    "three"}  
b = str(a)  
print(b)    # {1: 'one', 2: 'two', 3: 'three'}
```

В Python словари могут быть *вложенными*, то есть выступать частью другого более крупного словаря.

```
a = {  
    "First": {  
        1: "one",  
        2: "two",  
        3: "three"  
    },  
    "Second": {  
        4: "four",  
        5: "five"  
    }  
}  
print(a) # {'First': {1: 'one', 2: 'two', 3: 'three'}, 'Second': {4: 'four', 5: 'five'}}
```

В примере, создается словарь a, включающий в себя два других словаря (First и Second), т.е. в свою очередь, содержат несколько пар ключей и значений.