

Лекция №7

Контрольная работа

- Вопросы по теории
- Практика

Вопросы по теории

1. Каким будет результат деления $7// -3$?
2. Какие типы являются изменяемыми из tuple, list, dictionary, set?
3. Допустимо ли такое выражение: $1 \leq a < 10$ and $1 \leq b < 20$
4. Что будет в результате в переменной a?

```
a = [[]]*3  
a[0].append(3)  
a[1] = 1
```
5. Что такое функциональный объект (функтор) и как его создать?

Вопросы по теории

6. Что выведется на стандартный ввод/вывод при выполнении следующего кода:

```
class TheClass:
    def __call__(self, n, *args, **kwargs):
        n += 1
        args[0].extend((2, 3, 4))
        kwargs['the_list'].remove(5)

value_n = 1
list_of_args = [1, 3, 5]
the_class = TheClass()
the_class(value_n, list_of_args, the_list=list_of_args)
print(value_n, list_of_args)
```

7. Можно ли писать else в try-excerpt блоках?

8. Как добавить новый атрибут в объект класса и как предотвратить такую возможность?

Вопросы по теории

```
class MyClass(object):  
    __slots__ = ('a', 'b', 'c')  
  
    def __init__(self):  
        self.a = 'value a'  
        self.b = 'value b'  
        self.c = 'value c'  
        self.d = 'value d'  
  
    def __str__(self):  
        return ", ".join([self.a, self.b, self.c, self.d])  
  
if __name__ == "__main__":  
    try:  
        my_object = MyClass()  
        my_object.d = 'value d'  
        print(my_object)  
    except Exception as e:  
        print("Error! Message: {}".format(e))
```

Практика

1. Написать реализацию встроенной функции len: функция принимает список, возвращает его длину. (3 балла)
2. Написать реализацию функции reversed: функция принимает список, возвращает его же, располагая элементы в обратном порядке. (3 балла)
3. * Написать реализацию функции range: она может принимать от одного до трех аргументов, при этом аргументами должны быть целые числа (int). range(**старт**, **стоп**, **шаг**) - так выглядит стандартный вызов функции range() в Python. По умолчанию старт равняется нулю, шаг - единице. Возвращает список целых чисел в форме [**старт**, **старт** + **шаг**, **старт** + **шаг** * 2...]. Если шаг положительное число, последним элементом списка будет наибольшее (**старт** + i * **шаг**) меньшее числа **стоп**. Если шаг отрицательное число, то последний элемент будет наименьшее (**старт** + i * **шаг**) большее числа **стоп**. Предусмотреть случаи ошибочных аргументов (например, **шаг** == 0). (5 баллов)



Практика

4. Написать функцию `to_title`: принимает строку, ищет пробелы, первые буквы после них и после начала строки делает заглавными. (3 балла)

```
>>> to_title('orlov Ilya evgenyevich')
'Orlov Ilya Evgenyevich'
```

5. Написать функцию `count_symbol`: считает сколько раз символ встречается в строке. (3 балла)

```
>>> count_symbol("Hi, Elvis, I am here!", "i")
2
```

6. * Написать реализацию функции `format`. (5 баллов, с re – 7 баллов)

```
>>> myformat('{1}, {0}, {2}', 'a', 'b', 'c')
'b, a, c'
>>> myformat('coordinates: {}, {}', '37.4N', '18.3W')
'coordinates: 37.4N, 18.3W'
```

Практика

7. Написать функцию `copyfile`: функция принимает два аргумента – имена файлов `source` и `destination`, открывает `source`, читает его, открывает `destination`, пишет в него. Если `source` не найден или `destination` уже существует, то выбрасываются соответствующие исключения. Нужно проверить выполнение функции как для правильных аргументов, так и для приводящих к исключениям. (3 балла)
8. * Написать функцию `copydir` - копирование директории с использованием `copyfile`, а также проверки на существование `source` и `destination`. (5 баллов)

Практика

9. Создайте класс User, в котором будут следующие поля: name (имя), age (возраст), методы setName, getName, setAge, getAge.

Создайте класс Worker, который наследуется от класса User и имеет дополнительное поле salary (зарплата), а также методы getSalary и setSalary.

Создайте объект этого класса name='John', age=25, salary=1000.

Создайте второй объект этого класса 'Jack', age=26, salary=2000.

Найдите сумму зарплат объектов John и Jack. (3 балла)

10. * Создайте класс Money (деньги) для работы с денежными суммами.

Число должно быть представлено двумя полями: для рублей и для копеек. Дробная часть (копейки) при выводе на экран должна быть отделена от целой части запятой. Необходимо реализовать сложение, вычитание, деление сумм и операции сравнения. Также нужно добавить атрибут текущий курс по доллару и метод перевода по текущему курсу в доллары (для справки по перегрузке операторов

<https://pythonworld.ru/osnovy/peregruzka-operatorov.html>). (7 баллов)