



**Федеральное государственное бюджетное
образовательное учреждение
высшего образования
«Московский государственный технический
университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

Факультет «Информатика и вычислительная техника»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка интернет-приложений»

Отчет по РК№1
«Функциональные возможности языка Python»

Выполнил:
студент группы ИУ5-52Б

Олейников И.И.

Подпись и дата:

Проверил:
преподаватель каф.
ИУ5

Гапанюк Ю.Е.

Подпись и дата:

Москва, 2021 г.

Вариант Б, 17

Текст программы

Main

```
from conductor import Conductor
from orchestra import Orchestra
from controlDB import ControlDB

def main():
    names_conductor = ['Константин', 'Евгений', 'Михаил', 'Илья']

    bd = ControlDB()
    # Создаем дирижеров
    bd.set_conductors(*names_conductor)

    # Добавляем оркестры к дирижерам
    bd.set_orchestra("ансамбль Константина - 1", 7, 0)
    bd.set_orchestra("ансамбль Константина - 2", 16, 0)
    bd.set_orchestra("ансамбль Константина - 3", 5, 0)
    bd.set_orchestra("ансамбль Евгения - 1", 24, 1)
    bd.set_orchestra("ансамбль Евгения - 2", 12, 1)
    bd.set_orchestra("ансамбль Михаила - 1", 17, 2)
    bd.set_orchestra("ансамбль Ильи - 1", 4, 3)
    bd.set_orchestra("ансамбль Ильи - 2", 28, 3)

    print(bd.getSort()) #Задание 1
    print(bd.get_cost()) #Задание 2

    # для M-M
    bd.set_many_orchestra("ан К,И - 1", 7, 0, 3)
    bd.set_many_orchestra("ан К,Е - 2", 16, 0, 1)
    bd.set_many_orchestra("ан К,М - 3", 5, 0, 2)
    bd.set_many_orchestra("ан Е,М - 1", 24, 1, 2)
    bd.set_many_orchestra("ан Е,И - 2", 12, 1, 3)
    bd.set_many_orchestra("ан М,И - 1", 17, 2, 3)

    print(bd.get_orcetst_cond()) #Задание 3

if __name__ == '__main__':
    main()
```

cm_timer

```
import time
from contextlib import contextmanager

class cm_timer_1:
    def __enter__(self):
        self.start_time = time.time()

    def __exit__(self, exc_type, exc_val, exc_tb):
        print("Время выполнения {} секунд".format(time.time() -
self.start_time))
        if exc_val:
            raise
```

```

@contextmanager
def cm_timer_2():
    start_time = time.time()
    yield
    print("Время выполнения {} секунд".format(time.time() - start_time))

def test_cm_timer():
    print('Вызываем конт. мен. как класс')
    with cm_timer_1():
        time.sleep(1)

    print('Вызываем конт. мен. как contextmanager')
    with cm_timer_2():
        time.sleep(1)

```

controlBD

```

from conductor import Conductor
from orchestra import Orchestra

class ControlDB:
    def __init__(self):
        self.list_conductors = []
        self.list_orchestra = []

    def getSort(self) -> list:
        return sorted(self.list_orchestra, key=lambda x: x.get_name()) #
сортируем

    def get_cost(self) -> list:
        return list(map(lambda x: '{} - кол-ом
{}'.format(x, x.get_n()), sorted(self.list_conductors, key=lambda x:
x.get_n())) # сортируем

    def get_orcetst_cond(self):
        l = []
        for i in self.list_orchestra:
            if len(i.get_id_conductor()) > 1 and i.get_name()[2]=='ан':
                sr = 'Оркестр: {} дирижеры: {}'.format(i.get_name())
                for j in i.get_id_conductor():
                    sr += str(j.get_id_conductor()) + ';'
                l.append(sr)
        return l

    def set_conductors(self, *arg):
        for i in arg:
            self.list_conductors.append(Conductor(i))

    def set_orchestra(self, name:str, size:int, id_cond:int):
        self.list_orchestra.append(Orchestra(name, size,
[self.list_conductors[id_cond]]))
        self.list_conductors[id_cond].up_n()

    def set_many_orchestra(self, name:str, size:int, *id_cond:int):
        self.list_orchestra.append(Orchestra(name, size,
[self.list_conductors[x] for x in id_cond]))

```

orchestra

```
from orchestra_conductor import OrchestraConductor

class Orchestra:
    def __init__(self, name, size, id_conductor:list):
        #global num_orchestra
        #self.__id_orchestra = num_orchestra
        self.__id_orchestra = id(self)
        self.__name = name
        self.__size = size
        if len(id_conductor) == 1:
            self.__id_conductor = id_conductor
        elif len(id_conductor) > 1:
            self.__id_conductor = []
            for i in id_conductor:

self.__id_conductor.append(OrchestraConductor(self.__id_orchestra, i))

        #num_orchestra += 1

    def get_name(self):
        return self.__name

    def get_id_conductor(self):
        return self.__id_conductor

    def __repr__(self):
        return 'название: {}, дирижер: {}'.format(self.__name,
self.__id_conductor)
```

conductor

```
class Conductor:
    def __init__(self, name):
        self.__id_conductor = id(self)
        self.__name = name
        self.n = 0

    def __repr__(self):
        return self.__name

    def return_id(self):
        return self.__id_conductor

    def up_n(self) -> None:
        self.n += 1

    def get_n(self) -> int:
        return self.n
    return {'a': 1, 'b': 2, 'c': 3, 'd': 4}

@print_result
def test_4():
    return [1, 2, 'a', 28]

def all_test():
    test_1()
    test_2()
    test_3()
    test_4()
```

orchestraConductor

```
class orchestraConductor:
    def __init__(self, id_orchestra, id_conductor):
        self.__id_orchestra = id_orchestra
        self.__id_conductor = id_conductor

    def get_id_conductor(self):
        return self.__id_conductor
```

Результат выполнения

```
[название: ансамбль Евгения - 1, дирижер: [Евгений], название: ансамбль Евгения - 2, дирижер: [Евгений], название: ансамбль Ильи - 1, дирижер: [Илья], названи
['Михаил - кол-ом 1', 'Евгений - кол-ом 2', 'Илья - кол-ом 2', 'Константин - кол-ом 3']
['Оркестр: ан К,И - 1 дирижеры: Константин; Илья; ', 'Оркестр: ан К,Е - 2 дирижеры: Константин; Евгений; ', 'Оркестр: ан Е,М - 1 дирижеры: Евгений; Михаил; ']

Process finished with exit code 0
```

1:[название: ансамбль Евгения - 1, дирижер: [Евгений], название: ансамбль Евгения - 2, дирижер: [Евгений], название: ансамбль Ильи - 1, дирижер: [Илья], название: ансамбль Ильи - 2, дирижер: [Илья], название: ансамбль Константина - 1, дирижер: [Константин], название: ансамбль Константина - 2, дирижер: [Константин], название: ансамбль Константина - 3, дирижер: [Константин], название: ансамбль Михаила - 1, дирижер: [Михаил]]

2:['Михаил - кол-ом 1', 'Евгений - кол-ом 2', 'Илья - кол-ом 2', 'Константин - кол-ом 3']

3:['Оркестр: ан К,И - 1 дирижеры: Константин; Илья; ', 'Оркестр: ан К,Е - 2 дирижеры: Константин; Евгений; ', 'Оркестр: ан Е,М - 1 дирижеры: Евгений; Михаил; ']