Московский государственный технический

университет им. Н.Э. Баумана

Выполнил:

Студент группы ИУ5-33Б

Соколов Илья Владимирович

Подпись и дата:

Курс «Базовые компоненты интернет-технологий»

Отчёт по рубежному контролю 2

Факультет «Информатика и система управления»

Кафедра ИУ5 «Система обработки информации и управления»

Проверил:

Преподаватель каф. ИУ5

Гапанюк Юрий Евгеньевич

Подпись и дата:

Код:

Main.py

from random import randint

# используется для сортировки

from operator import itemgetter

class Composition:

    '''Класс Музыкальная Композиция

    '''

    def \_\_init\_\_(self,id,name,duration,comp\_id):

        self.id=id

        self.name=name

        self.duration=duration

        self.comp\_id=comp\_id

    def get\_duration(self):

        return self.duration

class Orcestra:

    def \_\_init\_\_(self,id,name) -> None:

        '''Принимает уникальный индекс орекстра,

           его название, кол-во сыгранных композиций

        '''

        self.id=id

        self.name=name

    def \_\_repr\_\_(self):

        return f'Name: {self.name}, id: {self.id}'

class Orcestra\_to\_Composition:

    '''Обеспечивает связь М:М

       Между классами Оркестр и Композиция

    '''

    def \_\_init\_\_(self,Orcestra\_id,Composition\_id):

        self.orc\_id=Orcestra\_id

        self.comp\_id=Composition\_id

    def get\_orcestra\_id(self):

        return self.orc\_id

    def get\_id\_tuple(self):

        return (self.orc\_id,self.comp\_id)

orcestras=[

            Orcestra(1,"Лондонский симфонический оркестр"),

            Orcestra(2,"Московский оркестр"),

            Orcestra(3,"Белорусский классический оркестр"),

            Orcestra(4,"Токийский новый оркестр")

         ]

composition=[

            Composition(1,'Undertale','6:20',4),

            Composition(2,'Seasons. Vivaldi','9:00',1),

            Composition(3,'Asgore','4:10',4),

            Composition(4,'Hopes and dreams','3:20',4),

            Composition(5,'Щелкунчик','20:03',2),

            Composition(6,'Гимн Белоруси','2:13',3)

]

orc\_to\_comp=[Orcestra\_to\_Composition(1,1),Orcestra\_to\_Composition(1,2),

Orcestra\_to\_Composition(3,3),Orcestra\_to\_Composition(3,6),Orcestra\_to\_Composition(2,2)

    ]

def main():

    one\_to\_many=[(c.name,'Длительность(мин,сек): {}'.format(c.duration),o.name)

            for o in orcestras

            for c in composition

            if o.id==c.comp\_id]

    print("B1. Сортировка значений по названию в алфавитном порядке")

    print(\*sorted(one\_to\_many,key=itemgetter(0)),sep='\n')

    print()

if \_\_name\_\_=='\_\_main\_\_':

    main()

rk2\_tests.py

import pytest

from main import Orcestra,Composition,Orcestra\_to\_Composition

def test\_Composition():

    obj=Composition(1,'Asgore',"2:40",23)

    assert obj.get\_duration()=='2:40'

def test\_Orc():

    name='Tokyo new orcestra'

    obj=Orcestra(1,name)

    assert obj.\_\_repr\_\_()==f'Name: {name}, id: {1}'

def test\_Orc\_to\_Comp():

    obj=Orcestra\_to\_Composition(1,42)

    assert obj.get\_id\_tuple()==(1,42)

Результаты:



