Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Автоматизированные системы обработки информации и управления»

Дисциплина «Разработка интернет-приложений»

Отчёт по рубежному контролю №1

Выполнил:   Проверил: Студент группы ИУ5-53Б   Преподаватель  
Аникин Ф.А.     Гапанюк Ю.Е.

Москва, 2020 г.

**Постановка задачи**

**Вариант В.**

1. «Класс» и «Школьник» связаны соотношением один-ко-многим. Выведите список всех школьников, у которых фамилия начинается с буквы «А», и номера их классов.
2. «Класс» и «Школьник» связаны соотношением один-ко-многим. Выведите список классов с минимальным рейтингом школьников в каждом классе, отсортированный по минимальному рейтингу.
3. «Класс» и «Школьник» связаны соотношением многие-ко-многим. Выведите список всех связанных школьников и классов, отсортированный по школьникам, сортировка по классам произвольная.

**Текст программы**

# используется для сортировки

from operator import itemgetter

class Student:

"""Школьник"""

def \_\_init\_\_(self, id, name, rating, class\_id):

self.id = id

self.name = name

self.rating = rating

self.class\_id = class\_id

class Class:

"""Класс"""

def \_\_init\_\_(self, id, number):

self.id = id

self.number = number

class StudClass:

"""

'StudClass' для реализации

связи многие-ко-многим

"""

def \_\_init\_\_(self, class\_id, student\_id):

self.class\_id = class\_id

self.student\_id = student\_id

# Школяры

students = [

Student(1, 'Якубов', 14.88, 2),

Student(2, 'Власов', 2, 1),

Student(3, 'Аникин', 13.37, 2),

Student(4, 'Троцкий', 3, 1),

Student(5, 'Ленин', 19.17, 3),

Student(6, 'Сталин', 19.45, 3),

Student(7, 'Абдуллаев', 42.0, 2),

Student(8, 'Горбачёв', 1.991, 1)

]

# Классы

classes = [

Class(1, 5),

Class(2,228),

Class(3,420)

]

studsclasses = [

StudClass(1,2),

StudClass(1,4),

StudClass(1,8),

StudClass(2,1),

StudClass(2,3),

StudClass(2,7),

StudClass(3,5),

StudClass(3,6)

]

def main():

"""Основная функция"""

# Соединение данных один-ко-многим

one\_to\_many = [(s.name, s.rating, c.number)

for s in students

for c in classes

if s.class\_id == c.id]

# Соединение данных многие-ко-многим

many\_to\_many\_temp = [(c.number, sc.class\_id, sc.student\_id)

for c in classes

for sc in studsclasses

if c.id == sc.class\_id]

many\_to\_many = [(s.name, student\_name)

for student\_name, student\_id, class\_id in many\_to\_many\_temp

for s in students if s.id == class\_id]

print('Задание B1')

res\_11 = list(filter(lambda x: x[0].startswith('А'), one\_to\_many))

print(res\_11)

print('\nЗадание B2')

res\_12\_unsorted = []

for c in classes:

meow = list(filter(lambda i: i[2] == c.number, one\_to\_many))

if len(meow) > 0:

count = [rating for \_, rating, \_ in meow]

count\_min = min(count)

res\_12\_unsorted.append((c.number, count\_min))

res\_12 = sorted(res\_12\_unsorted, key=itemgetter(1), reverse=False)

print(res\_12)

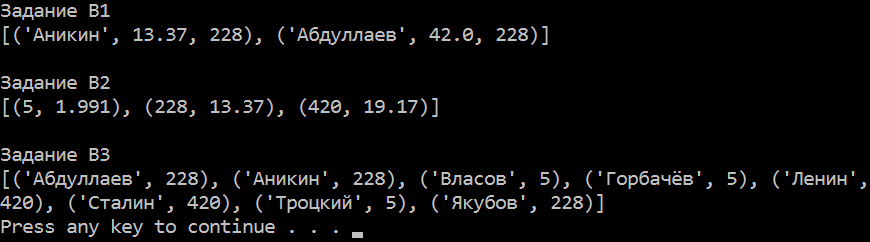
print('\nЗадание B3')

res\_13 = sorted(many\_to\_many, key=itemgetter(0))

print(res\_13)

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

main()

**Пример выполнения работы программы**