РК-1 Курс ПиКЯП Турсков Евгений ИУ5-36Б

Вариант Д26

**Текст программы:**

**main.py**

**from** operator **import** itemgetter  
  
  
**class** StudentGroup:  
 """Студенческая группа"""  
  
 **def** \_\_init\_\_(self, id, name):  
 self.id = id  
 self.name = name  
  
  
**class** Course:  
 """Учебный курс"""  
  
 **def** \_\_init\_\_(self, id, title, avg\_grade, group\_id):  
 self.id = id  
 self.title = title  
 self.avg\_grade = avg\_grade  
 self.group\_id = group\_id  
  
  
**class** GroupCourse:  
 """  
 'Группы на курсе' для реализации   
 связи многие-ко-многим  
 """  
  
 **def** \_\_init\_\_(self, group\_id, course\_id):  
 self.group\_id = group\_id  
 self.course\_id = course\_id  
  
  
# Студенческие группы  
groups = [  
 StudentGroup(1, 'Группа ИТ-101'),  
 StudentGroup(2, 'Группа Матем-201'),  
 StudentGroup(3, 'Группа Физ-301'),  
 StudentGroup(4, 'Армейская Группа СВО-243'),  
]  
  
# Учебные курсы  
courses = [  
 Course(1, 'Программирование', 85, 1),  
 Course(2, 'Математический анализ', 90, 2),  
 Course(3, 'Физика', 75, 3),  
 Course(4, 'История', 80, 4),  
 Course(5, 'Изучение языков', 88, 2),  
 Course(6, 'Философия', 93, 3),  
]  
  
# Связи многие-ко-многим  
groups\_courses = [  
 GroupCourse(1, 1),  
 GroupCourse(2, 2),  
 GroupCourse(3, 3),  
 GroupCourse(4, 4),  
 GroupCourse(2, 5),  
 GroupCourse(3, 6),  
 GroupCourse(4, 6),  
 GroupCourse(3, 1),  
]  
  
  
**def** main():  
 """Основная функция"""  
  
 # Соединение данных один-ко-многим  
 one\_to\_many = [(c.title, c.avg\_grade, g.name)  
 **for** g **in** groups  
 **for** c **in** courses  
 **if** c.group\_id == g.id]  
  
 # Соединение данных многие-ко-многим  
 many\_to\_many\_temp = [(g.name, gc.group\_id, gc.course\_id)  
 **for** g **in** groups  
 **for** gc **in** groups\_courses  
 **if** g.id == gc.group\_id]  
  
 many\_to\_many = [(c.title, c.avg\_grade, group\_name)  
 **for** group\_name, group\_id, course\_id **in** many\_to\_many\_temp  
 **for** c **in** courses **if** c.id == course\_id]  
  
 print('Запрос Д1')  
 res\_d1 = [(title, group\_name) **for** title, avg\_grade, group\_name **in** one\_to\_many **if** title.endswith("ов")]  
 print(res\_d1)  
  
 print('\nЗапрос Д2')  
 res\_d2\_unsorted = []  
 **for** g **in** groups:  
 g\_courses = list(filter(**lambda** i: i[2] == g.name, one\_to\_many))  
 **if** len(g\_courses) > 0:  
 g\_grades = [avg\_grade **for** \_, avg\_grade, \_ **in** g\_courses]  
 g\_avg\_grade = sum(g\_grades) / len(g\_grades)  
 res\_d2\_unsorted.append((g.name, g\_avg\_grade))  
  
 res\_d2 = sorted(res\_d2\_unsorted, key=itemgetter(1), reverse=**True**)  
 print(res\_d2)  
  
 print('\nЗапрос Д3')  
 res\_d3 = {}  
 **for** g **in** groups:  
 **if** g.name.startswith("А"):  
 g\_courses = list(filter(**lambda** i: i[2] == g.name, many\_to\_many))  
 g\_courses\_titles = [x **for** x, \_, \_ **in** g\_courses]  
 res\_d3[g.name] = g\_courses\_titles  
  
 print(res\_d3)  
  
  
**if** \_\_name\_\_ == '\_\_main\_\_':  
 main()

**Результаты выполнения:**

**Запрос Д1**

**Изучение языков - Армейская Группа СВО-243**

**Теория алгоритмов - Группа Физ-301**

**Запрос Д2**

**Армейская Группа СВО-243 - 90.0**

**Астрономическая Группа 504 - 87.0**

**Группа Физ-301 - 85.66666666666667**

**Группа ИТ-101 - 84.5**

**Группа Матем-201 - 79.0**

**Археологическая Группа А-102 - 76.0**

**Запрос Д3**

**Армейская Группа СВО-243: Математический анализ, Изучение языков, Инвестирование**

**Астрономическая Группа 504: Экономический анализ, Философия**

**Археологическая Группа А-102: Введение в биологию, Теория алгоритмов**

**Process finished with exit code 0**