|  |  |
| --- | --- |
|  | Министерство науки и высшего образования Российской Федерации  Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»  (МГТУ им. Н.Э. Баумана) |
| ФАКУЛЬТЕТ «ИНЖЕНЕРНЫЙ БИЗНЕС И МЕНЕДЖМЕНТ»  КАФЕДРА «ПРОМЫШЛЕННАЯ ЛОГИСТИКА» (ИБМ-3)  Отчет по выполнению рубежного контроля №1 по дисциплине “Парадигмы и конструкции языков программирования”  38.03.05 Бизнес-Информатика (уровень бакалавриата)  Студент ИБМ3-34 Шевченко Г.А.  2024 г. | |

Код программы:  
# 1) Определение классов данных в соответствии с предметной областью "Студенческая группа" и "Кафедра" (Вариант 28, Шевченко Г.А)

# Класс "Кафедра"

class Department:

  def \_\_init\_\_(self, id, name):

    self.id = id         # ID записи о кафедре

    self.name = name       # Наименование кафедры

# Класс "Студенческая группа"

class StudentGroup:

  def \_\_init\_\_(self, id, name, num\_students, department\_id):

    self.id = id         # ID записи о студенческой группе

    self.name = name       # Наименование группы

    self.num\_students = num\_students # Количество студентов (количественный признак)

    self.department\_id = department\_id # ID записи о кафедре (для связи один-ко-многим)

# Класс для связи многие-ко-многим "Группы кафедры"

class DepartmentGroup:

  def \_\_init\_\_(self, department\_id, group\_id):

    self.department\_id = department\_id # ID записи о кафедре

    self.group\_id = group\_id      # ID записи о студенческой группе

# 2) Создание списков объектов классов с тестовыми данными

# Список кафедр

departments = [

  Department(1, "Кафедра ФН"),

  Department(2, "Кафедра ИУ"),

  Department(3, "Кафедра ИБМ"),

  Department(4, "Кафедра СМ")

]

# Список студенческих групп

student\_groups = [

  StudentGroup(1, "ФН1-13Б", 25, 1),

  StudentGroup(2, "ИУ7-24Б", 30, 2),

  StudentGroup(3, "ИБМ3-34Б", 28, 3),

  StudentGroup(4, "ИУ6-14Б", 28, 2),

  StudentGroup(5, "СМ12-43Б", 23, 4),

  StudentGroup(6,"ИБМ6-33Б",29,3),

  StudentGroup(7,"ФН3-41Б",23,1),

  StudentGroup(8,"СМ14-32Б",21,4)

  ]

# Связи многие-ко-многим между кафедрами и студенческими группами

department\_groups = [

  DepartmentGroup(1, 1),

  DepartmentGroup(1, 7),

  DepartmentGroup(2, 2),

  DepartmentGroup(2, 4),

  DepartmentGroup(3, 3),

  DepartmentGroup(3, 6),

  DepartmentGroup(4, 5),

  DepartmentGroup(4, 8),

  ]

# Создание словарей для быстрого доступа к данным по ID

department\_dict = {dept.id: dept for dept in departments}

group\_dict = {group.id: group for group in student\_groups}

# 3) Запросы

# Запрос №1: Выведите список всех связанных студенческих групп и кафедр, отсортированный по кафедрам, сортировка по студенческим группам произвольная.

# Создаем список кортежей (кафедра, студенческая группа)

dept\_group\_list = [(department\_dict[group.department\_id], group) for group in student\_groups]

# Сортируем список по названию кафедры

dept\_group\_list.sort(key=lambda x: x[0].name)

# Результат

print("Список всех связанных студенческих групп и кафедр, отсортированный по кафедрам:")

for dept, group in dept\_group\_list:

  print(f"Кафедра: {dept.name}, Студенческая группа: {group.name}")

# Запрос №2: Выведите список кафедр с общим количеством студентов в студенческих группах в каждой кафедре, отсортированный по общему количеству студентов.

# Суммируем количество студентов по кафедрам

dept\_student\_counts = {}

for group in student\_groups:

  dept\_id = group.department\_id

  dept\_student\_counts[dept\_id] = dept\_student\_counts.get(dept\_id, 0) + group.num\_students

# Преобразуем данные в список кортежей (кафедра, общее количество студентов)

dept\_totals = [(department\_dict[dept\_id], total\_students) for dept\_id, total\_students in dept\_student\_counts.items()]

# Сортируем по общему количеству студентов (по убыванию)

dept\_totals.sort(key=lambda x: x[1], reverse=True)

# Выводим результат

print("\nСписок кафедр с общим количеством студентов в студенческих группах, отсортированный по общему количеству студентов:")

for dept, total\_students in dept\_totals:

  print(f"Кафедра: {dept.name}, Общее количество студентов: {total\_students}")

# Запрос №3:

# Выведите список всех кафедр, у которых в названии присутствует слово «кафедра», и список связанных с ними студенческих групп.

# Создаем словарь department\_id -> [group\_id, ...] для связей многие-ко-многим

dept\_to\_groups = {}

for dg in department\_groups:

  dept\_to\_groups.setdefault(dg.department\_id, []).append(dg.group\_id)

# Выводим список кафедр и связанных с ними студенческих групп

print("\nСписок кафедр, у которых в названии присутствует слово «кафедра», и связанных с ними студенческих групп:")

for dept\_id, group\_ids in dept\_to\_groups.items():

  dept = department\_dict[dept\_id]

  if "кафедра" in dept.name.lower():

    # Получаем список студенческих групп

    groups = [group\_dict[group\_id] for group\_id in group\_ids]

    print(f"Кафедра: {dept.name}")

    print("Студенческие группы:")

    for group in groups:

      print(f"- {group.name}")

    print()

Результат выполнения программы:  
