

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 «Радиотехнический»

КАФЕДРА «Системы обработки информации и управления»

**Отчет по рубежному контролю №1 по курсу**

**«Базовые компоненты интернет-технологий»**

Выполнил: Проверил:

Кузнецов Артемий Викторович Гапанюк Юрий Евгеньевич

2021 год

**Описание задания**

Рубежный контроль представляет собой разработку программы на языке Python, которая выполняет следующие действия:

1) Необходимо создать два класса данных в соответствии с Вашим вариантом предметной области, которые связаны отношениями один-ко-многим и многие-ко-многим. Пример классов данных для предметной области Сотрудник-Отдел:

1. Класс «Сотрудник», содержащий поля:

● ID записи о сотруднике;

● Фамилия сотрудника;

● Зарплата (количественный признак);

● ID записи об отделе. (для реализации связи один-ко-многим)

2. Класс «Отдел», содержащий поля:

● ID записи об отделе;

● Наименование отдела.

3. (Для реализации связи многие-ко-многим) Класс «Сотрудники отдела», содержащий поля:

● ID записи о сотруднике;

● ID записи об отделе.

2) Необходимо создать списки объектов классов, содержащих тестовые данные (3-5 записей), таким образом, чтобы первичные и вторичные ключи соответствующих записей были связаны по идентификаторам.

3) Необходимо разработать запросы в соответствии с Вашим вариантом. Запросы сформулированы в терминах классов «Сотрудник» и «Отдел», которые используются в примере. Вам нужно перенести эти требования в Ваш вариант предметной области. При разработке запросов необходимо по возможности использовать функциональные возможности языка Python (list/dict comprehensions, функции высших порядков).

Для реализации запроса №2 введите в класс, находящийся на стороне связи «много», произвольный количественный признак, например, «зарплата сотрудника».

Результатом рубежного контроля является документ в формате PDF, который содержит текст программы и результаты ее выполнения.

**Текст программы**

from operator import itemgetter

class PL:

"""Язык программирования"""

def \_\_init\_\_(self, id, name\_PL, year, SR\_id):

self.id = id

self.name\_PL = name\_PL # имя

self.year = year # год создания

self.SR\_id = SR\_id # среда разработки

class SR:

"""Средство разработки"""

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

self.id = id

self.name = name

class PLSR:

"""

связь многие ко многим

"""

def \_\_init\_\_(self, SR\_id, PL\_id):

self.SR\_id = SR\_id

self.PL\_id = PL\_id

# Компьютеры

SRs = [

SR(1, 'Консоль'),

SR(2, 'Среда Visual Studio'),

SR(3, 'Среда Visual Code'),

SR(4, 'Блокнот'),

SR(5, 'Среда Xcode'),

]

# Жесткие диски

PLs = [

PL(1, 'Python', 1991, 3),

PL(2, 'C#', 2000, 2),

PL(3, 'C++', 1983, 2),

PL(4, 'Assembler', 1949, 1),

PL(5, 'Html', 1993, 4),

PL(6, 'Swift', 2014, 5),

]

PLs\_SRs = [

PLSR(1,1),

PLSR(1,2),

PLSR(2,4),

PLSR(2,6),

PLSR(3,3),

PLSR(3,5),

PLSR(4,6),

PLSR(4,1),

PLSR(5,2),

PLSR(5,4),

PLSR(6,1),

PLSR(6,3),

]

def main():

"""Основная функция"""

# Соединение данных один-ко-многим

one\_to\_many = [(p.name\_PL, p.year, s.name)

for s in SRs

for p in PLs

if p.SR\_id==s.id]

# Соединение данных многие-ко-многим

many\_to\_many\_temp = [(s.name, sp.PL\_id, sp.SR\_id)

for s in SRs

for sp in PLs\_SRs

if s.id==sp.SR\_id]

many\_to\_many = [(p.name\_PL, p.year, SR\_name)

for SR\_name, SR\_id, PL\_id in many\_to\_many\_temp

for p in PLs if p.id==PL\_id]

print('E1')

E1 = []

for name\_PL, year, name in one\_to\_many:

if 'Среда' in name: # Ищем средство разработки с ключевым словом "Среда"

E1.append((name, name\_PL))

print(E1)

print('\nE2')

# находим средний год выпуска языков

E2\_unsorted = []

# Перебираем все компьютеры

for s in SRs:

# Список языков компьютера

PLSS = list(filter(lambda i: i[2]==s.name, one\_to\_many))

if len(PLSS) > 0:

Y = [year for \_,year,\_ in PLSS]

avg\_sum = sum(Y)/len(Y)

E2\_unsorted.append((s.name, avg\_sum))

E2 = sorted(E2\_unsorted, key=itemgetter(1))

print(E2)

print('\nE3')

# находим языки программирования, начинающиеся с "С" и выводим их среды

E3 = []

for name\_PL, year, name in many\_to\_many:

if name\_PL.find("C") == 0:

E3.append((name\_PL, name))

print(E3)

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

main()