Московский государственный технический

университет им. Н.Э. Баумана

Факультет «Радиотехнический»

Кафедра ИУ5 «Информатика и вычислительная техника»

**Курс «Разработка интернет-приложений»**

**Отчет по рубежному контролю №1**

**«Разработка программ на языке Python.»**

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы РТ5-51 |  | преподаватель |
| Бушуев В.М. |  | Гапанюк Ю. Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2021 г.

Цель работы

Работа с классами в Python, организация и реализация запросов.

Задание

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

1) Необходимо создать два класса данных, которые связаны отношениями один-ко-многим и многие-ко-многим.

Классы:

1. Класс «Компьютер», содержащий поля:
   * ID компьютера;
   * Марка компьютера;
   * Цена (количественный признак);
   * ID записи о кабинете. (для реализации связи один-ко-многим)
2. Класс «Кабинет», содержащий поля:
   * ID кабинета;
   * Наименование кабинета.
3. (Для реализации связи многие-ко-многим) Класс «Сотрудники отдела», содержащий поля:
   * ID записи о компьютере;
   * ID записи о кабинете.

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 следующие запросы:

1. «Кабинеты» и «Компьютеры» связаны соотношением один-ко-многим. Выведите список всех кабинетов, у которых в названии присутствует слово «кабинет», и список находящихся в них компьютеров.
2. «Кабинеты» и «Компьютеры» связаны соотношением один-ко-многим. Выведите список кабинетов со средней ценой компьютеров в каждом кабинете, отсортированный по средней цене.
3. «Кабинеты» и «Компьютеры» связаны соотношением многие-ко-многим. Выведите список всех компьютеров, у которых марка начинается с буквы «А», и названия их кабинетов.

При разработке запросов необходимо по возможности использовать функциональные возможности языка Python (list/dict comprehensions, функции высших порядков).

Текст программы:

import functools as ft

import fnmatch

class Computer:

def \_\_init\_\_(self, ID, MARK, PRICE, RM\_ID):

self.id = ID

self.mark = MARK

self.pr = PRICE

self.rm\_id = RM\_ID

class Room:

def \_\_init\_\_(self, ID, NAME):

self.id = ID

self.name = NAME

class CompRoom:

def \_\_init\_\_(self, CMP\_ID, RM\_ID):

self.cmp\_id = CMP\_ID

self.rm\_id = RM\_ID

def room\_print(one\_many):

if ('кабинет' in one\_many[2].lower()):

print("%-25s" % one\_many[2])

return one\_many

def main():

computers = [Computer(1, "Apple", 70000, 1),

Computer(2, "Acer", 40000, 3),

Computer(3, "HP", 50000, 2),

Computer(4, "MSI", 60000, 1),

Computer(5, "Xiaomi", 55000, 2)]

rooms = [Room(1, "Лабораторный кабинет"),

Room(2, "Учительская"),

Room(3, "Кабинет робототехники")]

comp\_room = [CompRoom(1, 1),

CompRoom(1, 3),

CompRoom(2, 2),

CompRoom(3, 2),

CompRoom(3, 3),

CompRoom(4, 1),

CompRoom(5, 1),

CompRoom(5, 3)

]

one\_many = [(comp.mark, comp.pr, room.name)

for comp in computers

for room in rooms

if comp.rm\_id == room.id]

comps\_table = {comp.mark: [table.rm\_id for table in comp\_room

if (comp.id == table.cmp\_id)]

for comp in computers

}

a3 = {mark: [rm.name for rm in rooms if (rm.id in comps\_table[mark])]

for mark in comps\_table

}

class A12:

def \_\_init\_\_(self, room\_name, comps, sum\_pr) -> None:

self.rm = room\_name

self.computers = comps

self.avg\_pr = sum\_pr / len(self.computers)

a12 = [A12(rm.name, [comp for comp, pr, room in one\_many if room == rm.name], ft.reduce(

lambda acc, elem: acc + elem, [pr for comp, pr, room in one\_many if room == rm.name], 0))for rm in rooms]

print("\n\n%-25s|%-15s\n%25s|" % ("Кабинет", "Компьютер", ""), end="")

for room in a12:

print("\n%-25s|" % (room.rm), end='')

for comp in room.computers:

print("%-15s\n%25s|" % (comp, ""), end="")

print("\n\n\n\n%-25s|%-15s\n%25s|\n%25s|" %

("Кабинет", "Средняя цена", "", ""))

for room in sorted(a12, key=lambda x: x.avg\_pr):

print("%-25s|{0}\n%-25s|".format(room.avg\_pr) % (room.rm, ""))

print("\n\n\nКомпьютер / Его кабинеты:\n")

for key in a3:

if fnmatch.fnmatch(key, "A\*"):

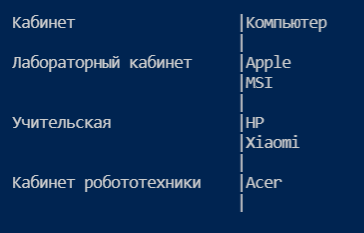
print(key + ": " + ", ".join(a3[key]))

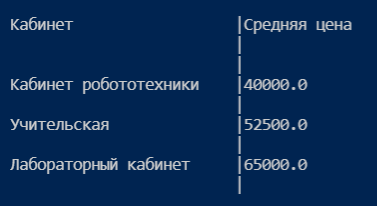
print("\n\n")

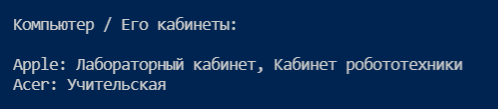
if \_\_name\_\_ == "\_\_main\_\_":

main()

Экранные формы с примерами выполнения программы:







Вывод

Продемонстрированы возможности работы с классами в Python, организации и реализации запросов.