**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет ИУ

Кафедра ИУ5

Курс «Основы информатики»

Отчет по рубежному контролю №1

Вариант 10

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-33Б: |  | преподаватель каф. \_ |
| Номоконов В.А. |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2024 г.

**Постановка задачи**

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.

**Вариант В.**

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех сотрудников, у которых фамилия начинается с буквы «А», и названия их отделов.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов с минимальной зарплатой сотрудников в каждом отделе, отсортированный по минимальной зарплате.
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по сотрудникам, сортировка по отделам произвольная.

|  |  |  |
| --- | --- | --- |
| 10 | Браузер | Компьютер |

**Текст программы**

class Browser:

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

self.id = id

self.name = name

self.version = version

self.year = year

self.computer\_id = None

class Computer:

def \_\_init\_\_(self, id, model, ram):

self.id = id

self.model = model

self.ram = ram

self.browsers = []

def main():

# Создание списков объектов классов с тестовыми данными

browsers = [

Browser(1, "Google Chrome", "90.0.4430.212", 2024),

Browser(2, "Mozilla Firefox", "88.0.1", 2021),

Browser(3, "Microsoft Edge", "90.0.818.42", 2024),

Browser(4, "Safari", "14.0.3", 2300),

Browser(5, "Opera", "76.0.4017.107", 2021),

Browser(6, "Atom", "26.0.0.21", 1812),

Browser(7, "Vivaldi", "5.5.0.0", 2023),

]

computers = [

Computer(1, "Dell Inspiron", 16),

Computer(2, "HP Envy", 8),

Computer(3, "Apple MacBook", 16),

Computer(4, "Apple MacBook Pro", 32),

Computer(5, "Apple MacBook Air", 16)

]

# Связывание браузеров с компьютерами

browsers[0].computer\_id = computers[0].id

browsers[1].computer\_id = computers[0].id

browsers[2].computer\_id = computers[1].id

browsers[3].computer\_id = computers[2].id

browsers[4].computer\_id = computers[2].id

browsers[5].computer\_id = computers[3].id

browsers[6].computer\_id = computers[3].id

computers[0].browsers = [browsers[0], browsers[1]]

computers[1].browsers = [browsers[2]]

computers[2].browsers = [browsers[3], browsers[4]]

computers[3].browsers = [browsers[5], browsers[6]]

# Вывод результатов

print("Результаты запросов:")

print("Запрос 1: Браузеры, у которых название начинается с буквы 'A', и названия их компьютеров.")

result = [

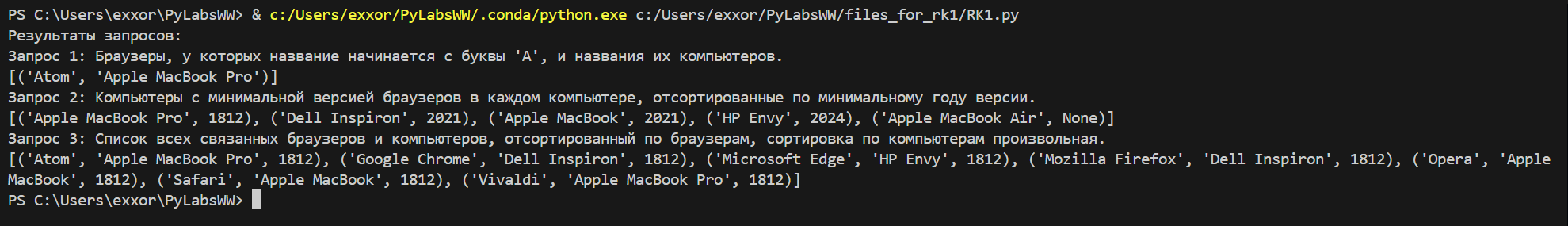
(browser.name, next((computer.model for computer in computers if computer.id == browser.computer\_id), None))

for browser in browsers

if browser.name.startswith("A") and browser.computer\_id is not None

]

**Анализ результатов:**

****