



*«Московский государственный технический университет
имени Н.Э. Баумана»
(национальный исследовательский университет)
(МГТУ им. Н.Э. Баумана)*

ФАКУЛЬТЕТ Инженерный бизнес и менеджмент

КАФЕДРА Бизнес-информатика

Отчет по рубежному контролю №1

“Парадигмы и конструкции языков программирования”

Вариант №10

Студент

В.Д. Зазовский

Группа ИБМЗ-34Б

Преподаватель

Гапанюк В.Ю.

Москва, 2024

Код программы:

```
from operator import itemgetter

class Browser:
    """Класс Браузер"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class Computer:
    """Класс Компьютер"""
    def __init__(self, id, model, browser_id):
        self.id = id
        self.model = model
        self.browser_id = browser_id

class BrowserComputer:
    """Класс для связи многие-ко-многим между браузерами и компьютерами"""
    def __init__(self, browser_id, computer_id):
        self.browser_id = browser_id
        self.computer_id = computer_id

# Список браузеров
browsers = [
    Browser(1, "Chrome"),
    Browser(2, "Firefox"),
    Browser(3, "Edge"),
]

# Список компьютеров
computers = [
    Computer(1, "Dell XPS", 1),
    Computer(2, "MacBook Pro", 2),
    Computer(3, "HP Spectre", 1),
    Computer(4, "ThinkPad", 2),
    Computer(5, "Mac Mini", 3),
]

# Связь многие-ко-многим между браузерами и компьютерами
browsers_computers = [
    BrowserComputer(1, 1),
    BrowserComputer(2, 2),
```

```

    BrowserComputer(1, 3),
    BrowserComputer(2, 4),
    BrowserComputer(3, 5),
]

def main():
    """Основная функция"""

    # Связь один-ко-многим: Браузер -> Компьютер
    one_to_many = [(c.model, b.name) for b in browsers for c in computers
if c.browser_id == b.id]

    # Связь многие-ко-многим: Браузер -> Компьютер
    many_to_many_temp = [(b.name, bc.browser_id, bc.computer_id) for b in
browsers for bc in browsers_computers if b.id == bc.browser_id]
    many_to_many = [(c.model, b_name) for b_name, browser_id, computer_id
in many_to_many_temp for c in computers if c.id == computer_id]

    # Запрос 1: Все браузеры и связанные с ними компьютеры,
отсортированные по названию браузеров
    print("Задание 1:")
    res_1 = sorted(one_to_many, key=itemgetter(1))
    print(res_1)

    # Запрос 2: Количество компьютеров для каждого браузера,
отсортированное по количеству компьютеров
    print("\nЗадание 2:")
    res_2 = []
    for b in browsers:
        # Список компьютеров для данного браузера
        computers_list = list(filter(lambda x: x[1] == b.name,
one_to_many))
        # Количество компьютеров
        res_2.append((b.name, len(computers_list)))
    res_2_sorted = sorted(res_2, key=itemgetter(1), reverse=True)
    print(res_2_sorted)

    # Запрос 3: Все браузеры, содержащие слово "Chrome", и их компьютеры
    print("\nЗадание 3:")
    res_3 = {}
    for b in browsers:
        if "Chrome" in b.name:
            computers_list = list(filter(lambda x: x[1] == b.name,
many_to_many))
            computers_names = [x[0] for x in computers_list]

```

```
        res_3[b.name] = computers_names
    print(res_3)

if __name__ == "__main__":
    main()
```

Вывод:

```
PS C:\Users\Spectre> & C:/Users/Spectre/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/Spectre/from operator import
temgetter.py"
Задание 1:
[('Dell XPS', 'Chrome'), ('HP Spectre', 'Chrome'), ('Mac Mini', 'Edge'), ('MacBook Pro', 'Firefox'), ('ThinkPad', 'Firefox')]

Задание 2:
[('Chrome', 2), ('Firefox', 2), ('Edge', 1)]

Задание 3:
{'Chrome': ['Dell XPS', 'HP Spectre']}
```