

**Московский государственный технический университет  
им. Н.Э. Баумана**

Факультет «Информатика и системы управления»  
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по рубежному контролю № 1  
Вариант Г-10

Выполнил:  
студент группы ИУ5-34Б  
Нечаева Мария

Проверил:  
преподаватель каф. ИУ5  
Нардид А. Н.

Москва, 2024 г.

## Текст программы (main.py):

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

class Computer:
    """Компьютер"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class BrowserComputer:
    """ для реализации связи многие-ко-многим """
    def __init__(self, computer_id, browser_id, users_count):
        self.computer_id = computer_id
        self.browser_id = browser_id
        self.users_count = users_count

# Список компьютеров
Computers = [
    Computer(1, "AComputer"),
    Computer(2, "AComputer1"),
    Computer(3, "BComputer"),
]

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

# Связь браузеров и компьютеров
Browsers_Computers = [
    BrowserComputer(1, 1, 1000),
    BrowserComputer(2, 4, 2000),
    BrowserComputer(3, 2, 1500),
]

def main():
    # Соединение данных один-ко-многим
    one_to_many = [(brows.name, comp.id, comp.name)
                    for comp in Computers
                    for brows in Browsers
                    if brows.comp_id == comp.id]

    # Соединение данных многие-ко-многим
    many_to_many = [(brows.name, comp.name, cob.users_count)
                    for comp in Computers
                    for cob in Browsers_Computers
                    for brows in Browsers if brows.id == cob.browser_id and
                    comp.id == cob.computer_id]

    # Г1: Выводим список всех компьютеров, которые начинаются с буквы «А»
    print('Задание Г1')
    result = filter(lambda x: x[2].startswith('A'), one_to_many)
    for i in result:
        print(f"Компьютер: {i[2]}, Браузер: {i[0]}")

    # Г2: Список компьютеров с максимальным количеством пользователей для
    # каждого браузера
    print('\nЗадание Г2')
    maximum_users = {}
    for brows_name, comp_name, users_count in many_to_many:
```

```

        if comp_name not in maximum_users or maximum_users[comp_name] <
users_count:
            maximum_users[comp_name] = users_count
        sorted_computers = sorted(maximum_users.items(), key=lambda x: x[1])
        for comp_name, users_count in sorted_computers:
            print(f"Компьютер: {comp_name}, Максимальное количество
пользователей: {users_count}")

# Г3: Список всех связанных браузеров и компьютеров
print('\nЗадание Г3')
sorted_browsers = sorted(many_to_many, key=lambda x: (x[1], x[0]))
for brows in sorted_browsers:
    print(f"Средство разработки: {brows[1]}, Язык программирования:
{brows[0]}")

if __name__ == '__main__':
    main()

```

## Результат выполнения программы:

Задание Г1

Компьютер: AComputer, Браузер: Chrome

Компьютер: AComputer1, Браузер: Safari

Задание Г2

Компьютер: AComputer, Максимальное количество пользователей: 1000

Компьютер: BComputer, Максимальное количество пользователей: 1500

Задание Г3

Средство разработки: AComputer, Язык программирования: Chrome

Средство разработки: BComputer, Язык программирования: Firefox