

**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 РК №1
Вариант запросов В
Вариант предметной области 7

Выполнил:
студент группы ИУ5-33Б
Саичкин Иван

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю. Е.

Москва, 2025 г.

Запросы

- 1) “Микропроцессор” и “Компьютер” связаны отношением один-ко-многим. Выведите список всех микропроцессоров, у которых название начинается с «А», и их компьютеров.
- 2) “ Микропроцессор” и “ Компьютер” связаны отношением один-ко-многим. Выведите список компьютеров с минимальной тактовой частотой процессоров, отсортированный по частоте.
- 3) “ Микропроцессор” и “ Компьютер” связаны отношением многие-ко-многим. Выведите список всех связанных микропроцессоров и компьютеров, отсортированный по микропроцессорам.

Листинг

```
class Microprocessor:  
    __slots__ = ('id', 'brand', 'clock_speed', 'computer_id')  
  
    def __init__(self, id: int, brand: str, clock_speed: float, computer_id: int) -> None:  
        self.id: int = id  
        self.brand: str = brand  
        self.clock_speed: float = clock_speed  
        self.computer_id: int = computer_id  
  
class Computer:  
    __slots__ = ('id', 'model', 'purpose')  
  
    def __init__(self, id: int, model: str, purpose: str) -> None:  
        self.id: int = id  
        self.model: str = model  
        self.purpose: str = purpose  
  
class ComputerMicroprocessor:  
    __slots__ = ('computer_id', 'microprocessor_id')  
  
    def __init__(self, computer_id: int, microprocessor_id: int) -> None:  
        self.computer_id: int = computer_id  
        self.microprocessor_id: int = microprocessor_id  
  
computers: list[Computer] = [  
    Computer(1, 'Gaming Pro', 'Игровой'),  
    Computer(2, 'Office Basic', 'Офисный'),  
    Computer(3, 'Workstation Elite', 'Рабочая станция'),  
    Computer(4, 'Server X9000', 'Серверный'),  
    Computer(5, 'Home Media', 'Домашний медиацентр')  
]  
  
microprocessors: list[Microprocessor] = [  
    Microprocessor(1, 'AMD Ryzen 9', 5.7, 1),  
    Microprocessor(2, 'AMD Ryzen 7', 4.7, 2),  
    Microprocessor(3, 'Intel Xeon', 4.2, 4),  
    Microprocessor(4, 'AMD Threadripper', 4.8, 3),  
    Microprocessor(5, 'Intel Core i5', 4.4, 2),  
    Microprocessor(6, 'ARM Cortex A78', 3.0, 5)  
]
```

```

# СВЯЗЬ МНОГИЕ-КО-МНОГИМ
computer_microprocessors: list[ComputerMicroprocessor] = [
    ComputerMicroprocessor(1, 1),
    ComputerMicroprocessor(1, 2),
    ComputerMicroprocessor(2, 2),
    ComputerMicroprocessor(2, 5),
    ComputerMicroprocessor(3, 4),
    ComputerMicroprocessor(4, 3),
    ComputerMicroprocessor(5, 6),
    ComputerMicroprocessor(5, 2)
]

def task1_micros_starting_with_A(one_to_many: list[tuple]) -> list[tuple[str, str]]:
    res = [(micro_brand, computer_model)
            for micro_brand, _, computer_model in one_to_many
            if micro_brand.startswith('A')]
    return res

def task2_computers_min_clock_speed(one_to_many: list[tuple]) -> list[tuple[str, float]]:
    computer_min_speed = {}

    for micro_brand, clock_speed, computer_model in one_to_many:
        if computer_model not in computer_min_speed or clock_speed <
computer_min_speed[computer_model]:
            computer_min_speed[computer_model] = clock_speed

    res = sorted([(computer_model, min_speed)
                  for computer_model, min_speed in
computer_min_speed.items(),
                  key=lambda x: x[1]])
    return res

def task3_sorted_micros_and_computers(many_to_many: list[tuple]) -> list[tuple[str, str]]:
    res = sorted([(micro_brand, computer_model)
                  for micro_brand, _, computer_model in many_to_many],
                  key=lambda x: x[0])
    return res

def main() -> None:
    # СВЯЗЬ ОДИН КО МНОГИМ
    one_to_many: list[tuple[str, float, str]] = [
        (micro.brand, micro.clock_speed, comp.model)
        for comp in computers
        for micro in microprocessors
        if micro.computer_id == comp.id
    ]

    temp: list[tuple[str, int, int]] = [
        (comp.model, cm.computer_id, cm.microprocessor_id)
        for comp in computers
        for cm in computer_microprocessors
        if comp.id == cm.computer_id
    ]

    # СВЯЗЬ МНОГИЕ КО МНОГИМ
    many_to_many: list[tuple[str, float, str]] = [
        (micro.brand, micro.clock_speed, comp_model)
    ]

```

```
        for comp_model, _, micro_id in temp
        for micro in microprocessors if micro.id == micro_id
    ]

print('Задание 1')
first_res = task1_mics_starting_with_A(one_to_many)
print('Список всех микропроцессоров, у которых название начинается с «А»,
и их компьютеров:')
for micro_brand, computer_model in first_res:
    print(f'{micro_brand} - {computer_model}')
print()

print('Задание 2')
second_res = task2_computers_min_clock_speed(one_to_many)
print('Список компьютеров с минимальной тактовой частотой процессоров,
отсортированный по частоте:')
for computer_model, min_speed in second_res:
    print(f'{computer_model}: {min_speed} ГГц')
print()

print('Задание 3')
third_res = task3_sorted_mics_and_computers(many_to_many)
print('Список всех связанных микропроцессоров и компьютеров,
отсортированный по микропроцессорам:')
for micro_brand, computer_model in third_res:
    print(f'{micro_brand} - {computer_model}')

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

Скриншот работы приложения

```
C:\Users\1\AppData\Local\Programs\Python\Python313\python.exe C:\Users\1\PycharmProjects\PICAP-3sem\RK1\main.py
```

Задание 1

Список всех микропроцессоров, у которых название начинается с «А», и их компьютеров:

```
AMD Ryzen 9 - Gaming Pro
AMD Ryzen 7 - Office Basic
AMD Threadripper - Workstation Elite
ARM Cortex A78 - Home Media
```

Задание 2

Список компьютеров с минимальной тактовой частотой процессоров, отсортированный по частоте:

```
Home Media: 3.0 ГГц
Server X9000: 4.2 ГГц
Office Basic: 4.4 ГГц
Workstation Elite: 4.8 ГГц
Gaming Pro: 5.7 ГГц
```

Задание 3

Список всех связанных микропроцессоров и компьютеров, отсортированный по микропроцессорам:

```
AMD Ryzen 7 - Gaming Pro
AMD Ryzen 7 - Office Basic
AMD Ryzen 7 - Home Media
AMD Ryzen 9 - Gaming Pro
AMD Threadripper - Workstation Elite
ARM Cortex A78 - Home Media
Intel Core i5 - Office Basic
Intel Xeon - Server X9000
```

```
Process finished with exit code 0
```