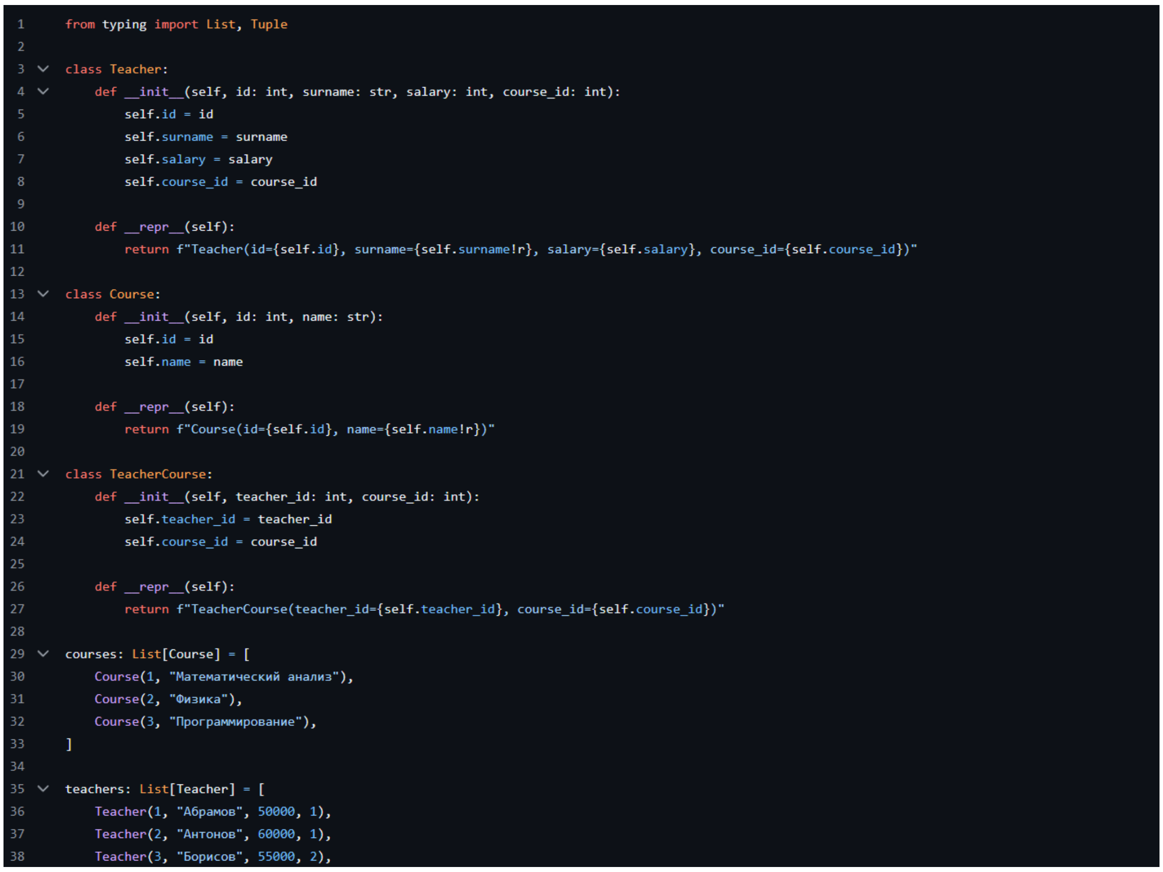
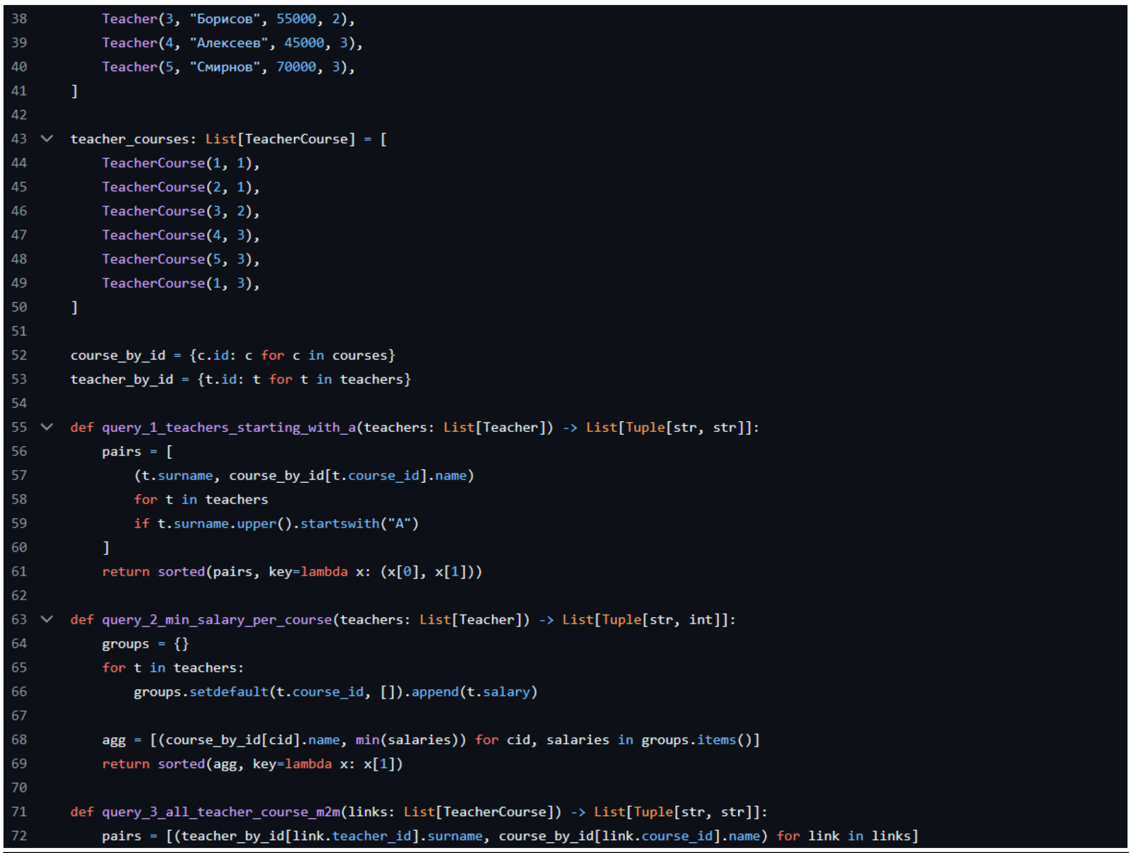
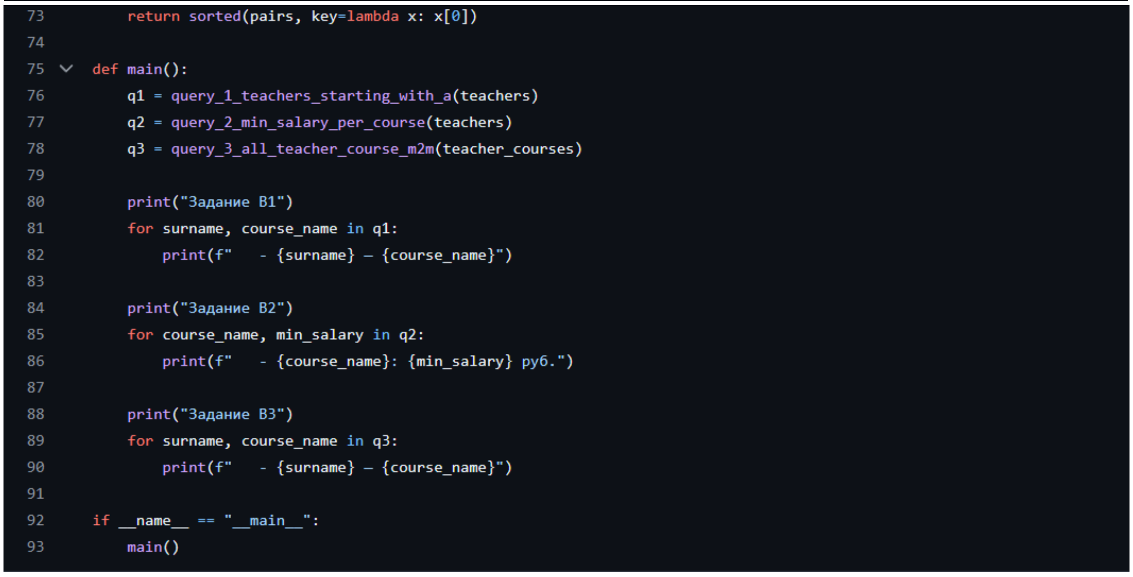
Код программы



```python
from typing import List, Tuple

class Teacher:
    def __init__(self, id: int, surname: str, salary: int, course_id: int):
        self.id = id
        self.surname = surname
        self.salary = salary
        self.course_id = course_id

    def __repr__(self):
        return f"Teacher(id={self.id}, surname={self.surname!r}, salary={self.salary}, course_id={self.course_id})"

class Course:
    def __init__(self, id: int, name: str):
        self.id = id
        self.name = name

    def __repr__(self):
        return f"Course(id={self.id}, name={self.name!r})"

class TeacherCourse:
    def __init__(self, teacher_id: int, course_id: int):
        self.teacher_id = teacher_id
        self.course_id = course_id

    def __repr__(self):
        return f"TeacherCourse(teacher_id={self.teacher_id}, course_id={self.course_id})"

courses: List[Course] = [
    Course(1, "Математический анализ"),
    Course(2, "Физика"),
    Course(3, "Программирование"),
]

teachers: List[Teacher] = [
    Teacher(1, "Абрамов", 50000, 1),
    Teacher(2, "Антонов", 60000, 1),
    Teacher(3, "Борисов", 55000, 2),
```

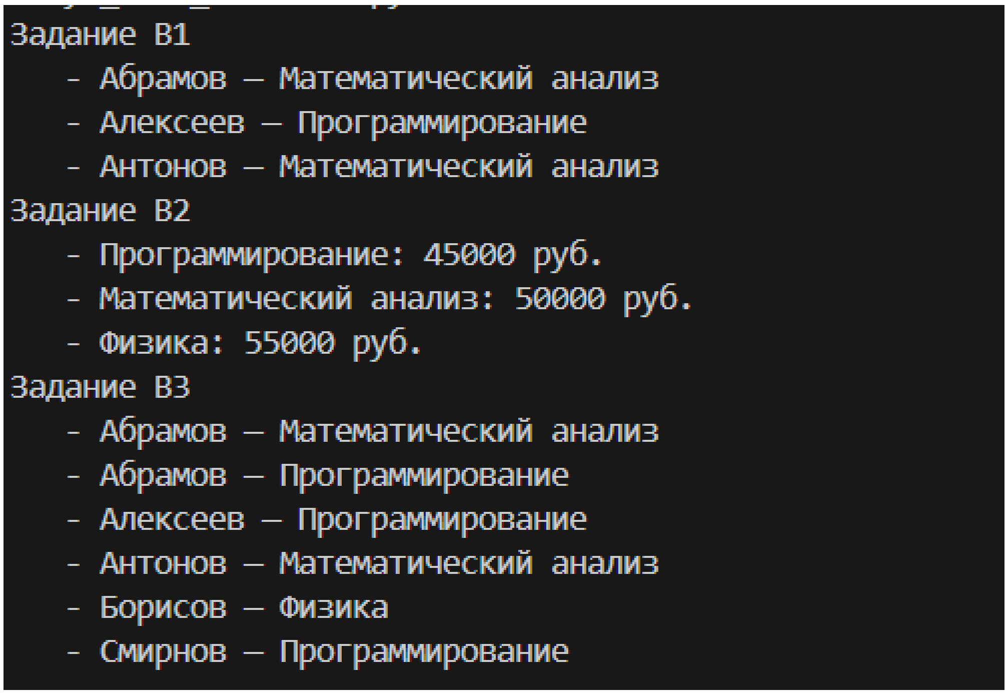


```python
    Teacher(3, "Борисов", 55000, 2),
    Teacher(4, "Алексеев", 45000, 3),
    Teacher(5, "Смирнов", 70000, 3),
]

teacher_courses: List[TeacherCourse] = [
    TeacherCourse(1, 1),
    TeacherCourse(2, 1),
    TeacherCourse(3, 2),
    TeacherCourse(4, 3),
    TeacherCourse(5, 3),
    TeacherCourse(1, 3),
]

course_by_id = {c.id: c for c in courses}
teacher_by_id = {t.id: t for t in teachers}

def query_1_teachers_starting_with_a(teachers: List[Teacher]) -> List[Tuple[str, str]]:
    pairs = [
        (t.surname, course_by_id[t.course_id].name)
        for t in teachers
        if t.surname.upper().startswith("А")
    ]
    return sorted(pairs, key=lambda x: (x[0], x[1]))

def query_2_min_salary_per_course(teachers: List[Teacher]) -> List[Tuple[str, int]]:
    groups = {}
    for t in teachers:
        groups.setdefault(t.course_id, []).append(t.salary)

    agg = [(course_by_id[cid].name, min(salaries)) for cid, salaries in groups.items()]
    return sorted(agg, key=lambda x: x[1])

def query_3_all_teacher_course_m2m(links: List[TeacherCourse]) -> List[Tuple[str, str]]:
    pairs = [(teacher_by_id[link.teacher_id].surname, course_by_id[link.course_id].name) for link in links]
```



```python
    return sorted(pairs, key=lambda x: x[0])

def main():
    q1 = query_1_teachers_starting_with_a(teachers)
    q2 = query_2_min_salary_per_course(teachers)
    q3 = query_3_all_teacher_course_m2m(teacher_courses)

    print("Задание Б1")
    for surname, course_name in q1:
        print(f"   - {surname} — {course_name}")

    print("Задание Б2")
    for course_name, min_salary in q2:
        print(f"   - {course_name}: {min_salary} руб.")

    print("Задание Б3")
    for surname, course_name in q3:
        print(f"   - {surname} — {course_name}")

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

Результат



```
Задание B1
   - Абрамов — Математический анализ
   - Алексеев — Программирование
   - Антонов — Математический анализ
Задание B2
   - Программирование: 45000 руб.
   - Математический анализ: 50000 руб.
   - Физика: 55000 руб.
Задание B3
   - Абрамов — Математический анализ
   - Абрамов — Программирование
   - Алексеев — Программирование
   - Антонов — Математический анализ
   - Борисов — Физика
   - Смирнов — Программирование
```