Рефакторинг кода

```
# module: models.py

class OperatingSystem:
    def __init__(self, os_id: int, name: str):
        self.os_id = os_id
        self.name = name


class Computer:
    def __init__(self, comp_id: int, os_id: int, model: str, salary: int):
        self.comp_id = comp_id
        self.os_id = os_id
        self.model = model
        self.salary = salary


class ComputerOperatingSystem:
    def __init__(self, comp_id: int, os_id: int):
        self.comp_id = comp_id
        self.os_id = os_id

# module: queries.py
from models import OperatingSystem, Computer, ComputerOperatingSystem


def query1(computers, operating_systems):
    return [
        (comp.model, os.name)
        for comp in computers
        if comp.model.startswith("A")
        for os in operating_systems
        if comp.os_id == os.os_id
    ]


def query2(computers, operating_systems):
    min_salary_per_os = {
        os.name: min(
            [comp.salary for comp in computers if comp.os_id == os.os_id], default=None
        )
        for os in operating_systems
    }
    return sorted(min_salary_per_os.items(), key=lambda x: x[1])


def query3(computers, operating_systems, computer_os):
    return sorted(
        [
```

```
        (comp.model, os.name)
        for co in computer_os
        for comp in computers
        for os in operating_systems
        if co.comp_id == comp.comp_id and co.os_id == os.os_id
    ],
    key=lambda x: x[0],
  )

# module: data.py
from models import OperatingSystem, Computer, ComputerOperatingSystem

operating_systems = [
  OperatingSystem(1, "Windows"),
  OperatingSystem(2, "Linux"),
  OperatingSystem(3, "macOS"),
]

computers = [
  Computer(1, 1, "Dell Inspiron", 70000),
  Computer(2, 2, "HP EliteBook", 80000),
  Computer(3, 1, "Lenovo ThinkPad", 75000),
  Computer(4, 3, "Apple MacBook", 150000),
  Computer(5, 2, "Acer Aspire", 60000),
]

computer_os = [
  ComputerOperatingSystem(1, 1),
  ComputerOperatingSystem(2, 2),
  ComputerOperatingSystem(3, 1),
  ComputerOperatingSystem(4, 3),
  ComputerOperatingSystem(5, 2),
]
```

Модульные тесты

```
import unittest
from queries import query1, query2, query3
from data import operating_systems, computers, computer_os


class TestQueries(unittest.TestCase):
  def test_query1(self):
    result = query1(computers, operating_systems)
    expected = [("Acer Aspire", "Linux")]
    self.assertEqual(result, expected)

  def test_query2(self):
    result = query2(computers, operating_systems)
    expected = [
```

```
            ("Linux", 60000),
            ("Windows", 70000),
            ("macOS", 150000),
        ]
        self.assertEqual(result, expected)

    def test_query3(self):
        result = query3(computers, operating_systems, computer_os)
        expected = [
            ("Acer Aspire", "Linux"),
            ("Apple MacBook", "macOS"),
            ("Dell Inspiron", "Windows"),
            ("HP EliteBook", "Linux"),
            ("Lenovo ThinkPad", "Windows"),
        ]
        self.assertEqual(result, expected)


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