Московский государственный технический университет им. Н.Э. Баумана

Факультет Радиотехнический
Кафедра “Системы обработки информации и управления”

Курс «Парадигмы и конструкции языков программирования»

Отчет по рубежному контролю №2

Выполнил:
студент группы РТ5-31Б:
Каландаров Алим
Шамильевич
Подпись и дата:

Москва, 2025 г.

Постановка задачи

1) Проведите рефакторинг текста программы рубежного контроля №1 таким образом, чтобы он был пригоден для модульного тестирования.

2) Для текста программы рубежного контроля №1 создайте модульные тесты с применением TDD - фреймворка (3 теста).

Текст программы

```
from dataclasses import dataclass
from typing import Iterable, List, Dict, Tuple


@dataclass(frozen=True)
class OpSystem:
    name: str
    system_id: int
    size_gb: float
    computer_id: int


@dataclass(frozen=True)
class Computer:
    computer_id: int
    model: str


@dataclass(frozen=True)
class OsComputer:
    system_id: int
    computer_id: int


def build_default_data() -> tuple[list[OpSystem], list[Computer],
list[OsComputer]]:

    opsystems = [
        OpSystem("Windows 10 Home", 1, 20, 1),
        OpSystem("Windows 10 Pro", 2, 27, 2),
        OpSystem("Windows 11", 3, 25, 3),
        OpSystem("MacOS Sonoma", 4, 28, 3),
        OpSystem("Arch Linux", 5, 1.3, 2),
    ]

    computers = [
        Computer(1, "Рабочий компьютер IMac"),
        Computer(2, "Игровой компьютер ASUS"),
        Computer(3, "Офисный компьютер HP"),
    ]

    os_comp = [
```

```
        OsComputer(4, 1),
        OsComputer(2, 2),
        OsComputer(3, 2),
        OsComputer(5, 3),
        OsComputer(1, 3),
    ]
    return opsystems, computers, os_comp


# ----------------------------
# Task B1
# ----------------------------

def task_b1_filtered_computers(
    opsystems: Iterable[OpSystem],
    computers: Iterable[Computer],
    keyword: str = "компьютер",
) -> list[tuple[str, float, str]]:

    kw = keyword.lower()
    comp_by_id = {c.computer_id: c for c in computers}

    filtered: list[tuple[str, float, str]] = []
    for os in opsystems:
        comp = comp_by_id.get(os.computer_id)
        if comp and kw in comp.model.lower():
            filtered.append((os.name, os.size_gb, comp.model))

    return sorted(filtered)


# ----------------------------
# Task B2
# ----------------------------

def task_b2_average_os_size_by_computer(
    opsystems: Iterable[OpSystem],
    computers: Iterable[Computer],
    *,
    rounding: int = 2,
) -> list[tuple[str, float]]:

    sizes_by_comp: Dict[int, list[float]] = {}
    for os in opsystems:
        sizes_by_comp.setdefault(os.computer_id, []).append(float(os.size_gb))

    result: list[tuple[str, float]] = []
    for comp in computers:
        sizes = sizes_by_comp.get(comp.computer_id, [])
        if sizes:
            avg = round(sum(sizes) / len(sizes), rounding)
```

```
            result.append((comp.model, avg))

    return sorted(result, key=lambda x: x[1])


# ---------------------------
# Task B3
# ---------------------------

def _first_word_len(s: str) -> int:
    first = s.split()[0] if s.split() else ""
    return len(first)


def task_b3_windows_os_and_computers(
    opsystems: Iterable[OpSystem],
    computers: Iterable[Computer],
) -> list[tuple[str, str]]:

    comp_by_id = {c.computer_id: c for c in computers}

    pairs: list[tuple[str, str]] = []
    for os in opsystems:
        if os.name and os.name[0].upper() == "W":
            comp = comp_by_id.get(os.computer_id)
            if comp:
                pairs.append((os.name, comp.model))

    return sorted(pairs, key=lambda item: _first_word_len(item[0]))



def main() -> None:
    opsystems, computers, _ = build_default_data()

    print("\n" + "=" * 70)
    print("Задание Б1 (Компьютеры со словом 'компьютер' и их операционные
системы):")
    print("=" * 70)
    print("Сортировка по названиям операционных систем:\n")

    for os_name, size_gb, comp_model in task_b1_filtered_computers(opsystems,
computers):
        print(f"Операционная система: {os_name:<30} | Размер: {size_gb:>8} ГБ |
Компьютер: {comp_model}")

    print("\n" + "=" * 70)
    print("Задание Б2 (Компьютеры со средним размером операционных систем):")
    print("=" * 70)
    print("Отсортированы по среднему размеру ОС в ГБ:\n")
```

```
    for comp_model, avg in task_b2_average_os_size_by_computer(opsystems,
computers):
        print(f"Компьютер: {comp_model:<35} | Средний размер ОС: {avg:>8.2f} ГБ")

    print("\n" + "=" * 70)
    print("Задание Б3 (ОС, начинающиеся на 'W', и их компьютеры):")
    print("=" * 70)
    print("Сортировка по длине названия операционных систем:\n")

    for os_name, comp_model in task_b3_windows_os_and_computers(opsystems,
computers):
        print(f"ОС: {os_name:<35} | Компьютер: {comp_model}")

    print("\n" + "=" * 70)
    print("Конец выполнения программы")
    print("=" * 70)


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

Тесты

```
import pytest

from rk1_refactored import (
    build_default_data,
    task_b1_filtered_computers,
    task_b2_average_os_size_by_computer,
    task_b3_windows_os_and_computers,
)


def test_b1_filters_by_keyword_and_sorts():
    opsystems, computers, _ = build_default_data()
    result = task_b1_filtered_computers(opsystems, computers,
keyword="компьютер")

# где содержится слово компьютер
    assert len(result) == 5

    # Sorted lexicographically by OS name (primary), then size, then mode
    # отсортированы по алфовиту, потом вторично по размеру
```

```
    assert [r[0] for r in result] == [
        "Arch Linux",
        "MacOS Sonoma",
        "Windows 10 Home",
        "Windows 10 Pro",
        "Windows 11",
    ]


def test_b2_computes_average_sizes_and_sorts_by_average():
    opsystems, computers, _ = build_default_data()
    result = task_b2_average_os_size_by_computer(opsystems, computers)


    # comp1: 20
    # comp2: (27 + 1.3)/2 = 14.15
    # comp3: (25 + 28)/2 = 26.5
    assert result == [
        ("Игровой компьютер ASUS", 14.15),
        ("Рабочий компьютер IMac", 20.0),
        ("Офисный компьютер HP", 26.5),
    ]


def test_b3_returns_only_windows_and_sorts_by_first_word_length():
    opsystems, computers, _ = build_default_data()
    result = task_b3_windows_os_and_computers(opsystems, computers)

    # Начинаются с W
    assert [r[0] for r in result] == ["Windows 10 Home", "Windows 10 Pro",
"Windows 11"]

    # отсортированы по слову "Windows"
    assert result[0][1] == "Рабочий компьютер IMac"
    assert result[1][1] == "Игровой компьютер ASUS"
    assert result[2][1] == "Офисный компьютер HP"
```

Анализ результатов

1. TDD тест

```
C:\Users\Mashiro\Projects\Labs\3_sem\PIKYAP\RK2_Klandarov>pytest test_rk1.py
============================================= test session starts =============================================
platform win32 -- Python 3.12.6, pytest-9.0.2, pluggy-1.6.0
rootdir: C:\Users\Mashiro\Projects\Labs\3_sem\PIKYAP\RK2_Klandarov
plugins: bdd-8.1.0
collected 3 items

test_rk1.py ...                                                                                      [100%]

============================================== 3 passed in 0.05s ==============================================

C:\Users\Mashiro\Projects\Labs\3_sem\PIKYAP\RK2_Klandarov>
```

2.Программа

```
Задание Б1 (Компьютеры со словом 'компьютер' и их операционные системы):
======================================================================
Сортировка по названиям операционных систем:

Операционная система: Arch Linux                    | Размер:      1.3 ГБ | Компьютер: Игровой компьютер ASUS
Операционная система: MacOS Sonoma                  | Размер:       28 ГБ | Компьютер: Офисный компьютер HP
Операционная система: Windows 10 Home               | Размер:       20 ГБ | Компьютер: Рабочий компьютер IMac
Операционная система: Windows 10 Pro                | Размер:       27 ГБ | Компьютер: Игровой компьютер ASUS
Операционная система: Windows 11                    | Размер:       25 ГБ | Компьютер: Офисный компьютер HP

======================================================================
Задание Б2 (Компьютеры со средним размером операционных систем):
======================================================================
Отсортированы по среднему размеру ОС в ГБ:

Компьютер: Игровой компьютер ASUS             | Средний размер ОС:    14.15 ГБ
Компьютер: Рабочий компьютер IMac             | Средний размер ОС:    20.00 ГБ
Компьютер: Офисный компьютер HP               | Средний размер ОС:    26.50 ГБ

======================================================================
Задание Б3 (ОС, начинающиеся на 'W', и их компьютеры):
======================================================================
Сортировка по длине названия операционных систем:

ОС: Windows 10 Home                    | Компьютер: Рабочий компьютер IMac
ОС: Windows 10 Pro                     | Компьютер: Игровой компьютер ASUS
ОС: Windows 11                         | Компьютер: Офисный компьютер HP

======================================================================
Конец выполнения программы
======================================================================
PS C:\Users\Mashiro>
```