



**«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Компьютерные системы и сети

О т ч е т

по лабораторной работе № 3

Дисциплина: Современные средства разработки программного обеспечения

Название лабораторной работы: Описание модели изоляции предметной области

Студент гр. ИУ6-56

(Подпись, дата)

Т.А. Тищенко

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

М.В. Фетисов

(И.О. Фамилия)

Цель работы: Приобрести навыки описания модели предметной области, выполнения ее изоляции.

Задание: Постройте диаграмму последовательности выполнения команды remove. Диаграмма должна показывать прохождение выполнения команды между программными объектами, расположенными в соответствующих слоях многоуровневой архитектуры.

Задача: «Война в долине теней»

Карточка: «Воин Долины теней». Карточка должна содержать поля: фракция (перечисляемый тип), сила удара, защита, здоровье, ловкость, уклонение, тип боя (ближний, дальний).

Код программы:

```
class WarriorVS : public ICollectable {
    int _group;
    int _power;
    int _protection;
    int _health;
    int _skill;
    int _deviation;
    int _battle_type;

protected:
    bool invariant() const {
        return _group >= 1 && _group <= 5
            && _group >= 0 && _group <= 9
            && _group >= 0 && _group <= 9
            && _group >= 0 && _group <= 9
            && _group >= 0 && _group <= 9
            && _group >= 0 && _group <= 9
            && _group >= 1 && _group <= 2;
    }

public:
    WarriorVS() = delete;
    WarriorVS(const WarriorVS & p) = delete;

    WarriorVS & operator = (const WarriorVS & p) = delete;

    WarriorVS(int group, int power, int protection, int health, int skill,
        int deviation, int battle_type): _group(group), _power(power),
        _protection(protection), _health(health), _skill(skill),
        _deviation(deviation), _battle_type(battle_type) {
        assert(invariant());
    }
}
```

```

virtual string getOutputString() const override {
    return to_string(_group) + " " + to_string(_power) + " " +
        to_string(_protection) + " " + to_string(_health) + " " +
        to_string(_skill) + " " + to_string(_deviation) + " " +
        to_string(_battle_type);
}

virtual bool write(ostream& os) override {
    writeNumber(os, _group);
    writeNumber(os, _power);
    writeNumber(os, _protection);
    writeNumber(os, _health);
    writeNumber(os, _skill);
    writeNumber(os, _deviation);
    writeNumber(os, _battle_type);

    return os.good();
}
};

class ItemCollector: public ACollector {
public:
    virtual shared_ptr<ICollectable> read(istream& is) override {
        int _group = readNumber<int>(is);
        int _power = readNumber<int>(is);
        int _protection = readNumber<int>(is);
        int _health = readNumber<int>(is);
        int _skill = readNumber<int>(is);
        int _deviation = readNumber<int>(is);
        int _battle_type = readNumber<int>(is);

        return make_shared<WarriorVS>(_group, _power, _protection,
            _health, _skill, _deviation, _battle_type);
    }
};

```

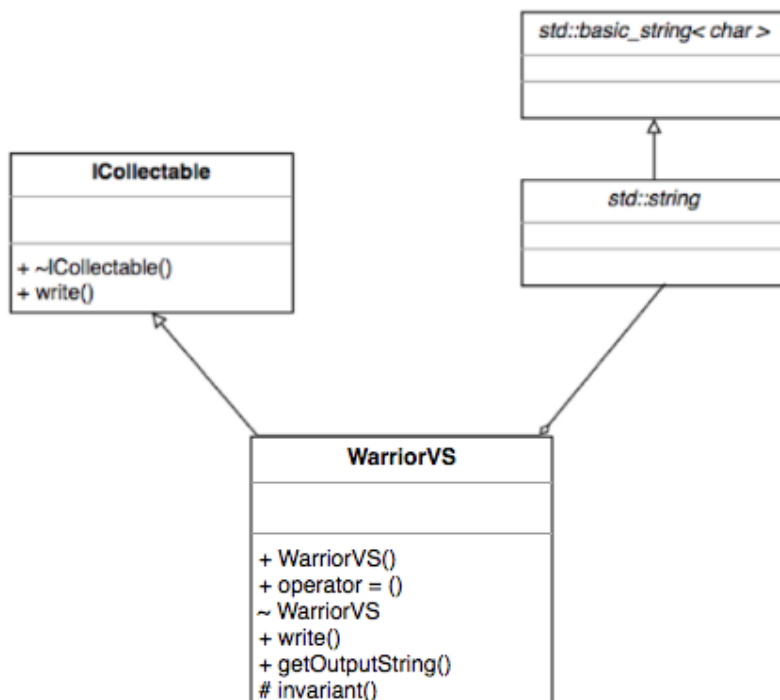


Рис.1 – диаграмма классов

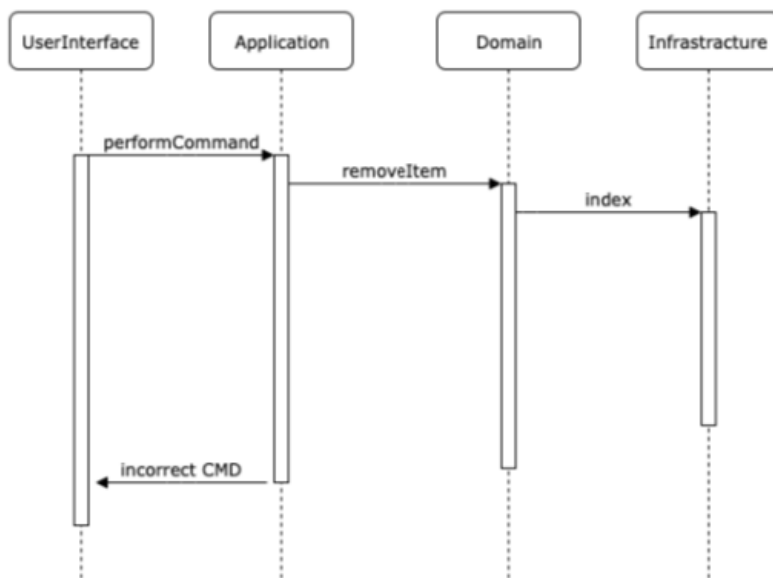


Рис.2 – диаграмма последовательности

Вывод программы:

```

a 2 2 2 5 3 2 1
update 0 1 10 9 10 10 10 2
v
[0] 1 10 9 10 10 10 2
Количество элементов в коллекции: 1
s

```

Вывод: Были приобретены навыки описания модели предметной области, выполнена ее изоляция.