

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В. И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №4
по дисциплине «Объектно-ориентированное программирование»
Тема: Шаблонные классы

Студент гр. 3342

Львов А.В.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2024

Цель работы

Создать шаблонный класс управления игрой, шаблонный класс отображения игры, Реализовать класс считывающий ввод пользователя из терминала и преобразующий ввод в команду

Задание

- A. Создать шаблонный класс управления игрой. Данный класс должен содержать ссылку на игру. В качестве параметра шаблона должен указываться класс, который определяет способ ввода команда, и переводящий введенную информацию в команду. Класс управления игрой, должен получать команду для выполнения, и вызывать соответствующий метод класса игры.
- B. Создать шаблонный класс отображения игры. Данный класс реагирует на изменения в игре, и производит отрисовку игры. То, как происходит отрисовка игры определяется классом переданном в качестве параметра шаблона.
- C. Реализовать класс считывающий ввод пользователя из терминала и преобразующий ввод в команду. Соответствие команды введенному символу должно задаваться из файла. Если невозможно считать из файла, то управление задается по умолчанию.
- D. Реализовать класс, отвечающий за отрисовку поля.

Примечание:

- Класс отслеживания и класс отрисовки рекомендуется делать отдельными сущностями. Таким образом, класс отслеживания инициализирует отрисовку, и при необходимости можно заменить отрисовку (например, на GUI) без изменения самого отслеживания
- После считывания клавиши, считанный символ должен сразу обрабатываться, и далее работа должна проводить с сущностью, которая представляет команду.
- Для представления команды можно разработать системы классов или использовать перечисление enum.
- Хорошей практикой является создание “прослойки” между считыванием/обработкой команды и классом игры, которая сопоставляет команду и вызываемым методом игры. Существуют альтернативные решения без явной “прослойки”

- При считывания управления необходимо делать проверку, что на все команды назначена клавиша, что на одну клавишу не назначено две команды, что на одну команду не назначено две клавиши.

Выполнение работы

Класс GameController представляет собой шаблонный класс управления игрой. Параметрами шаблона являются класс отображения игры и класс, реализующий считывание пользовательского ввода. Этот контроллер имеет методы начала игры, раунда и исполнения команды. Он передает команды в класс игры для их выполнения.

Класс Painter является шаблонным. Параметром шаблона является класс, который реализует отображение игры. Он имеет методы отрисовки поля и печати сообщения.

Класс TerminalPainter представляет собой отрисовщик поля в терминале и является параметром для шаблона Painter.

Класс InputHandler представляет собой класс, который считывает и валидирует данные пользователя. После считывания он преобразует полученные данные в команды. Он содержит методы для получения координат, длин кораблей, получения символов для исполнения соответствующей команды.

Тестирование

Для проверки работоспособности программы был запущен код, и был сыгран один раунд. В качестве параметров шаблонного класса GameController были выбраны InputHandler для считывания пользовательского ввода и TerminalPainter, отображающий изменения в игровом процессе в терминал. Для наглядности поле компьютера было видимым.

```
#include ...

int main() {
    GameController<InputHandler, Painter<TerminalPainter>> gameController;
    gameController.init();
    gameController.startGame();
    return 0;
}
```

```

./Battleship
===== Battleship =====
0 - New game
1 - Load game
0
Enter the count of ships: 1
Enter sizes of 1 ships:
Size of ship 1: 1
Next ability: Double damage
Enter coordinates for ship with length 1: 1 8
Orientation:
0 - Vertical
1 - Horizontal
1
  0 1 2 3 4 5 6 7 8 9
0 . . . . . . . . . .
1 . . . . . . . . . .
2 . . . . . . . . . .
3 . . . . . . . . . .
4 . . . . . . . . . .
5 . . . . . . . . . .
6 . . . . . . . . . .
7 . . . . . . . . . .
8 . S . . . . . . . .
9 . . . . . . . . . .

```

```

===== Your field =====
  0 1 2 3 4 5 6 7 8 9
0 . . . . . . . . . .
1 . . . . . . . . . .
2 . . . . . . . . . .
3 . . . . . . . . . .
4 . . . . . . . . . .
5 . . . . . . . . . .
6 . . . . . . . . . .
7 . . . . . . . . . .
8 . S . . . . . . . .
9 . . . . . . . . . .

===== Computer's field =====
  0 1 2 3 4 5 6 7 8 9
0 . . . . . . . . . .
1 . . . . . . . . . .
2 . . . . . . . . . .
3 . . . . . . . . . .
4 . . . . . . . . . .
5 . . . . . . . S . .
6 . . . . . . . . . .
7 . . . . . . . . . .
8 . . . . . . . . . .
9 . . . . . . . . . .

```

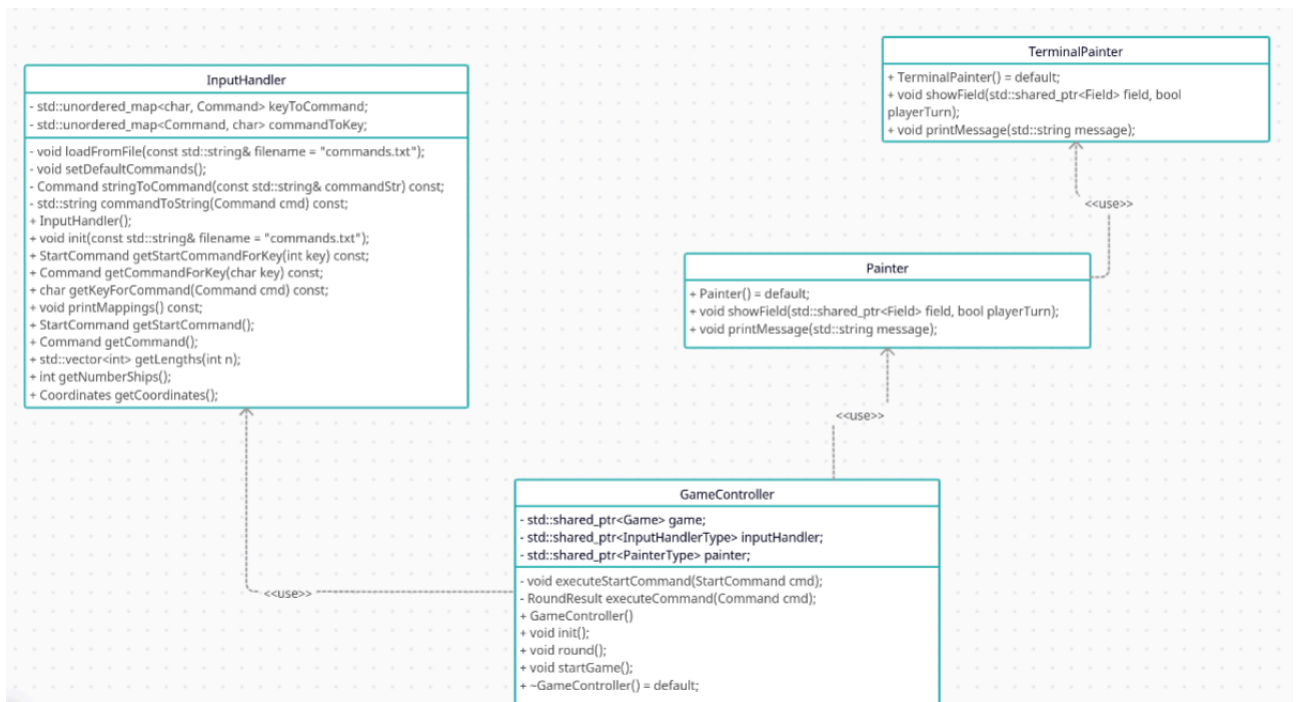
```

Enter a key for a command: z
Enter coordinates: 8 5
Computer attacks 0 2
Miss
===== Computer's field =====
  0 1 2 3 4 5 6 7 8 9
0 ~ ~ ~ ~ ~ ~ ~ ~ ~
1 ~ ~ ~ ~ ~ ~ ~ ~ ~
2 ~ ~ ~ ~ ~ ~ ~ ~ ~
3 ~ ~ ~ ~ ~ ~ ~ ~ ~
4 ~ ~ ~ ~ ~ ~ ~ ~ ~
5 ~ ~ ~ ~ ~ ~ ~ W ~
6 ~ ~ ~ ~ ~ ~ ~ ~ ~
7 ~ ~ ~ ~ ~ ~ ~ ~ ~
8 ~ ~ ~ ~ ~ ~ ~ ~ ~
9 ~ ~ ~ ~ ~ ~ ~ ~ ~

Key 'x' -> Exit
Key 's' -> Save
Key 'f' -> ShowField
Key 'a' -> Ability
Key 'z' -> Attack
Key 'v' -> Load

Enter a key for a command: z
Enter coordinates: 8 5
Computer attacks 9 8
Miss
Added new ability: Scanner
Player won!
===== Computer's field =====
  0 1 2 3 4 5 6 7 8 9
0 . . . . . . . . .
1 . . . . . . . . .
2 . . . . . . . . .
3 . . . . . . . . .
4 . . . . . . . . .
5 . . . . . . . . .
6 . . . . . . . . .
7 . . . . . . . . .
8 . . . . . . . S
9 . . . . . . . . .

```

Выводы

В ходе выполнения лабораторной работы, были реализованы шаблонный класс управления игрой, класс, обрабатывающий пользовательский ввод и класс, производящий отрисовку поля. Программа была успешно протестирована. Была реализована UML-диаграмма классов.