

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. 3385

Тарасов М.М.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2024

Цель работы

Изучить шаблонные классы в объектно-ориентированном программировании на языке C++. Создать классы взаимодействия с пользователем с использованием шаблонных классов.

Задание

а) 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.

б) 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.

с) 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.

д) Реализовать класс, отвечающий за отрисовку поля.

Выполнение работы

Класс GameManager

Класс GameManager представляет собой шаблонный класс, который выступает в роли посредника между классами взаимодействия с пользователем и классом игры. Он получает команды от обработчика команд и вызывает соответствующие методы класса игры.

Шаблоны класса GameManager:

- **typename T:** Шаблон для обработчика команд.
- **typename G:** Шаблон для класса отрисовки.

Основные методы класса GameManager:

- **use_command:** Главная логика класса. Метод получает команду от класса, отвечающего за считывание команд, и применяет её к игре.
- **game: Game*:** Указатель на класс игры.

Класс Parser

Класс Parser отвечает за обработку команд пользователя, вводимых через консоль.

Основные поля класса:

- **std::map<std::string, char> commands:** Хранит команды, где ключ — строка, а значение — символ команды.

Основные методы класса:

- **set_control:** Устанавливает соответствие команд и символов из файла.
- **get_command:** Считывает команду из std::cin и возвращает её.

Класс Input

Класс Input отвечает за считывание данных для игры.

Основные методы класса:

- **get_coords:** Считывает координаты.
- **get_lenghts:** Считывает длины кораблей.
- **get_orientations:** Считывает ориентации кораблей.
- Другие методы для считывания параметров игры.

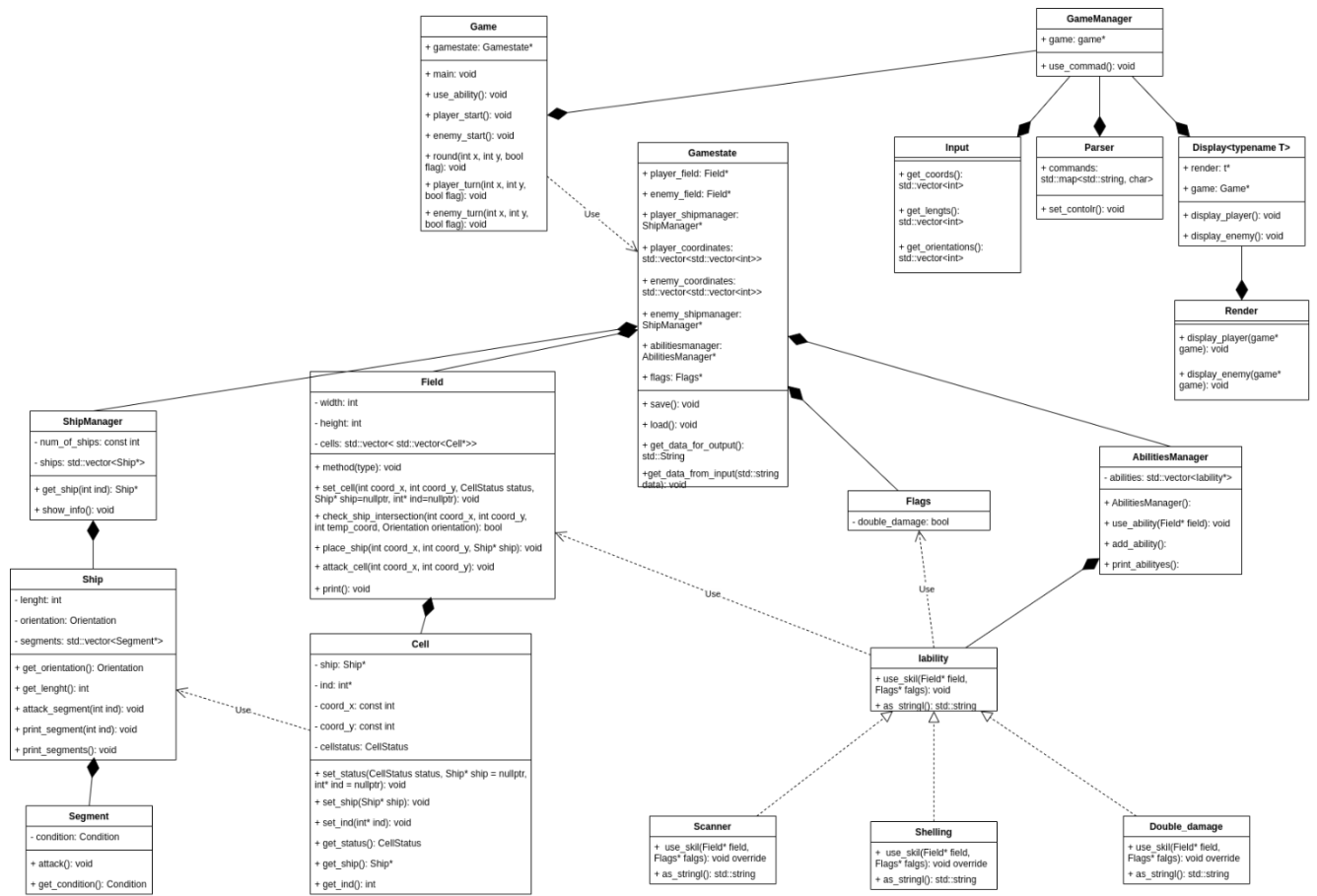
Класс Display

Класс Display — это шаблонный класс, отвечающий за отображение игры. Он реагирует на изменения в игре и выполняет отрисовку. Способ отрисовки определяется классом, переданным в качестве параметра шаблона.

Основные методы класса:

- **display_player_field:** Отображает поле игрока.
- **display_enemy_field:** Отображает поле врага.

UML-диаграмма классов



Выводы

В ходе работы был реализован проект игры "Морской бой" с использованием шаблонных классов.

Класс GameController отвечает за управление игрой, обрабатывает команды пользователя и вызывает соответствующие методы игры. Благодаря шаблонам, он поддерживает различные способы ввода и взаимодействия.

Класс GameDisplay отображает состояние игры и легко адаптируется к разным методам визуализации через параметр шаблона.

CommandParser позволяет гибко задавать привязки клавиш через файл настроек, с проверкой корректности.

Использование шаблонов повысило универсальность и масштабируемость программы, обеспечив простоту добавления новых функций и модификации.