

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

**ОТЧЕТ**  
**по лабораторной работе №5**  
**по дисциплине «Объектно-ориентированное программирование»**  
**Тема: Сериализация состояния программы**

Студент гр. 8383

\_\_\_\_\_

Колмыков В.Д.

Преподаватель

\_\_\_\_\_

Жангиров Т..

Санкт-Петербург

2020

## **Цель работы.**

Изучение сериализации в языке программирования C++, изучение паттерна, работающего с ней.

## **Ход работы.**

Для сохранения состояния программы используется паттерн снимок.

- 1) Для реализации паттерна были написаны классы UnitSnapshot, TowerSnapShot, ObjectSnapshot для записи и передачи информации о юнитах, базах и нейтральных объектах соответственно. В классах Unit, Tower и NeutralObject был добавлен метод getSnapshot(), возвращающий объект класса снимка, а также новые конструкторы, принимающие все сохраняемые параметры.

Новые классы находятся в файлах UnitSnapshot.h, UnitSnapshot.cpp, TowerSnapShot.h, TowerSnapShot.cpp, ObjectSnapshot.h, ObjectSnapshot.cpp.

- 2) Для сохранения состояния игры был написан класс Saver. Он последовательно сохраняет клетки поля, сохраняя информацию о башнях, юнитах, объектах используя снимки объектов.

Класс находится в файлах Saver.h и Saver.cpp.

- 3) Для загрузки состояния игры был написан класс Loader. Он последовательно загружает клетки поля.

Класс находится в файлах Loader.h и Loader.cpp.

- 4) Для сохранения игры по требованию игрока будет создаваться объект класса Saver. Он будет сохранять поле (в будущем так же состояние игры и т.д.) следующим образом: для каждой клетки будет получать информацию о юнитах, башнях, нейтральных объектах на ней при помощи «снимков» (метода getSnapshot()). Вся информация о каждой клетке последовательно записывается в файл.

- 5) Для загрузки игры по требованию игрока будет создаваться объект класса Loader. Он считывает информацию из файла, создает поле,

юнитов, башни и объекты заново. Также производится проверка последовательности ввода, предназначенная для контроля корректности файла загрузки.

6) UML диаграмма проекта представлена в файле uml5.jpeg.

### **Выводы.**

В ходе выполнения работы были изучены сериализация и паттерн, реализующие ее для использования в языке C++.