

**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. 8304

Щука А.А.

Преподаватель

Размочаева Н.В.

Санкт-Петербург
2020

Цель работы.

Научиться сохранять/загружать состояние программы.

Постановка задачи.

Реализация сохранения и загрузки состояния программы. Основные требования:

- Возможность записать состояние программы в файл
- Возможность считать состояние программы из файла
- Загрузка и сохранение должно выполняться в любой момент программы
- Взаимодействие с файлами должны быть по идиоме RAII

Ход выполнения.

1) Для того, чтобы реализовать сохранение и загрузку состояния программы был реализован класс Snapshot. Для компактной передачи параметров была реализована структура UnitSnapshot. Класс имеет два статических метода, которые принимают объект класса «игры». UML-диаграмма приведена на рис. 1.

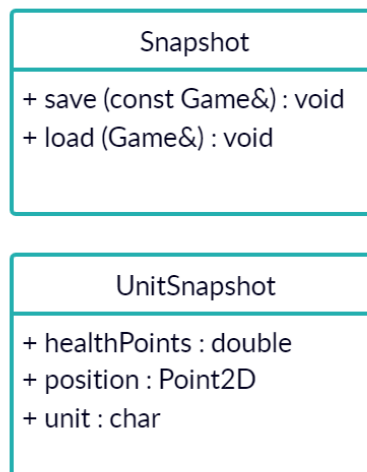


Рисунок 1- UML-диаграмма

2) Для сохранения все характеристики базы, количество золота, а также всех характеристики юнитов записываются в файл в определенном порядке. Действия повторяются для игрока 2.

3) Для загрузки состояния игры из файла данные считываются, создаются объекты с этими данными, и происходит связывание компонентов, например, добавление юнитов на поле.

4) Загрузка и сохранение происходят в любой момент работы программы, при нажатии на клавиши «0» и «9» соответственно.

5) Конструкторы некоторых классов, связанных с созданием юнитов, были изменены/перегружены.

6) Взаимодействие с файлами реализовано по идиоме RAII.

Выводы.

В ходе выполнения работы был создан класс для сохранения/загрузки состояния программы. Были выполнены все основные требования.