**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. 8303 |  | Логинов Е.А. |
| Преподаватель |  | Филатов А.Ю. |

Санкт-Петербург

2020

**Цель работы.**

Познакомиться с паттерном программирования “Снимо” и реализовать через него сохранение и загрузку игры.

**Ход работы.**

1. Была реализована возможность сохранения и загрузки состояния программы. Для этого был создан класс SnapshotBackup (файл SnapshotBackup.h/cpp).

Класс SnapshotBackup имеет метоты void SnapshotBackup::save() и void SnapshotBackup::load(), для загрузки и схранения соответственно. Метод save() записывает параметры поля, ландшафта, нейтральных объектов быза, юнитов и их характеристики в файл. Метод load() загружает все параметры из файла в той же последовательности, как они были сохранены и восстанавливает их на поле.

1. Сохранение и загрузку программы можно выполнить в любой момент.

Для этого пользователь вводит команду "load" или "save". Эта команда со обрабатывается в паттерне “Цепочка обязанностей” (класс ChainOfResposobility, файл ChainOfCommand.h), после чего вызывается соответсвующий метод в паттерне «Команда» (классы LoadCommand и SaveCommand файл Command.h) создается экземпляр класса SnapshotBackup и у него вызывается соответствующий метод.

В случае, если база отсутствовала на момент сохранения программы, то во время загрузки программа не будет пытаться ее считать и восстановить.

1. Взаимодействие с файлами происходит по идиоме RAII.

Конструктор SnapshotBuckup() захватывает экземпляр класса File(файл FileRAII.h) и высвобождает его в деструкторе ~SnapshotBuckup().

1. Сохранение и загрузка состояние игры реализованы при помощи паттерна "Снимок".

В методе поля SnapshotBackup\* Field::createSnap(), создается экземпляр снимка и в конструктор снимка передаются все поля игрового поля, которые необходимо сохранить. Существует возможность их передать, так как этот метод находится в классе Field и имеет доступ к приватным полям. Снимком является экземпляр класса SnapshotBackup. Экземпляр класса SnapshotBackup хранит в себе все необходимые для сохранения поля игрового поля и имеет возможность записывать их в файл. Также экземпляр снимка связан с полем, которое его создало и при загрузке данных меняет его параметры.

**Выводы.**

Были реализованы сохранение и загрузка состояния программы при помощи паттерна “Снимок”.