

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: Полиморфизм

Студентка гр. 8382

Рочева А.К.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2020

Цель работы.

Разработать и реализовать набор классов для ведения логирования действий и состояний программы.

Ход выполнения.

Были созданы классы для логирования действий пользователя, юнитов и базы (с интерфейсом `Logger` (`loggers/Logger.hpp`)). При создании игры пользователь выбирает тип логирования (`Manager::setLogger()`). Затем при выборе любого пункта меню логируются его действия (какую команду он выбрал). Уже в самих классах команд логируются действия юнитов и базы (записывается лог, возвращенный из метода `GameField::getLogString`, в котором используется лог, возвращенный из метода `Base::getLogString`).

Пользователь может выбрать три типа логирования – в файл (`loggers/FileLogger.hpp`), в терминал (`loggers/TerminalLogger.hpp`) либо вообще отказаться от логирования (`loggers/NoLogger.hpp`). Класс `Manager` (`Manager.hpp`) работает с заместителем этих логгеров (`LogProху.hpp`), в любой момент пользователь может поменять тип логирования.

Вывод логов происходит в методе `Logger::print()`. Для того, чтобы записывать номер лога и тип логирования, в классах логгеров перегружен оператор вывода в поток.

В классе `FileLogger` открытие файла происходит в конструкторе, а закрытие – в деструкторе, что соответствует идиоме RAII.

Так же при действиях базы и юнитов пользователь может выбрать формат вывода логов с текущим состоянием поля (пока что выводится свободное кол-во клеток, больше ничего интересного не придумала). Для этого создается класс `Adapter` (`loggers/Adapter.hpp`) с одним параметром – указателем на игровое поле. Затем в этом адаптере рассчитывается кол-во свободных клеток и запись в логгер новых строк.

В проекте находится файл file.txt, в котором показан пример логирования. Был выбран режим записи в файл состояния поля после действий базы или юнита.

Выводы.

В ходе выполнения работы были разработаны классы для логирования действий пользователя, базы и юнита.