

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ
по лабораторной работе №4
по дисциплине «Объектно-ориентированное программирование»
Тема: Полиморфизм

Студентка гр. 8381

Преподаватель

Звегинцева Е.Н.

Жангиров Т.Р.

Санкт-Петербург

2020

Цель работы.

Реализовать набор классов, для ведения логирования действий и состояний программы.

Задание.

Основные требования:

- Логирование действий пользователя.
- Логирование действий юнитов и базы.
- Реализована возможность записи логов в файл и терминал.
- Взаимодействие с файлами должно быть по идиоме RAII.
- Для логирования состояний перегружен оператор вывода в поток.
- Переключение между разным логированием (логирование в файл, в терминал, без логирования) реализуется при помощи паттерна «Прокси».
- Реализован разный формат записи при помощи паттерна «Адаптер».

Выполнение работы.

В данной работе реализована система логирования

Классы для логирования в файл и терминал:

- `Logger` – интерфейс для логирования в поток. Может вернуть/вывести сохраненную строку
- `TerminalLogger` – предназначен для вывода логов в терминал. В качестве потока используется `cout`. Использует перегрузку оператора вывода в поток.
- `FileLogger` – предназначен для вывода логов в файл. Использует перегрузку оператора вывода в поток. В качестве потока используется `ofstream`.

Переключение между разным логированием (логирование в файл, в терминал, без логирования) по принципу паттерна «Прокси» реализуется в классе `ProxyLogger`. Унаследован от интерфейса `Logger`. В конструктор передается желаемый тип логгера и создается соответствующий объект, указатель на

который хранится в приватном поле класс. Если же логирование отключено, указателю передается nullptk.

Класс Adapter реализован по принципу паттерна «адаптер». В классе реализована обработка информации для формирования строки и вывод её логгером. Логгер хранится в приватном поле класса.

Основные методы класса представлены в табл.1.

Таблица 1 - методы класс Adapter

Метод	Описание
setLogger	Метод для установки логгера
makeLog	Метод вывода лога. Принимаемые параметры:
baseAddLog	Формирование лога создания базы
neutralAddLog	Формирование лога создания нейтрального объекта
unitAddLog	Формирование лога создания юнита
attackLog	Формирование лога атаки
moveLog	Формирование лога передвижения
gameInfoLog	Формирование лога информация об игре
baseInfoLog	Формирование лога информация о базе
userInfoLog	Формирование лога информация о юните
neutralInfoLog	Формирование лога информация о нейтральном объекте
landCellInfo	Формирование лога информация о ландшафте

Был реализован класс фасада, который хранит объект адаптера в приватном поле.

Вывод.

В ходе выполнения лабораторной работы была написана программа, в которой реализованы классы для логирования состояний программы. Были использованы паттерны проектирования, а также принципы объектно-ориентированного программирования.