

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

Студент гр. 8304

Мешков М.А.

Преподаватель

Размочаева Н.В.

Санкт-Петербург

2020

Цель работы.

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

- Логирование действий пользователя
- Логирование действий юнитов и базы
- Взаимодействие с файлами должны быть по идиоме RAII
- Реализована возможность записи логов в терминал
- Реализована возможность записи логов в файл

Ход работы.

Разработан и реализован набор классов логирования:

- Был создан интерфейс `Logger`. Этот интерфейс содержит 5 методов (которые должны реализовать наследники):
 - `logGameStarting` — при вызове логирует начало игры,
 - `logUnitCreation` — при вызове логирует создание юнита,
 - `logUnitMovement` — при вызове логирует перемещение юнита,
 - `logAttack` — при вызове логирует атаку,
 - `logThingApplying` — при вызове логирует применение нейтрального объекта к юниту.
- Интерфейс `Logger` реализуют два класса. Эти два класса реализуют паттерн «Адаптер» - они помогают обеспечить взаимодействие объекта (в данном случае это `GameProxyWithLogging` (см. ниже)), который может взаимодействовать только с наследниками `Logger`, с наследниками `std::ostream`. Эти два класса:
 - `PlainLoggerToOstreamAdapter` - обеспечивает вывод информации о игровом событии в `std::ostream`.
 - `WithTimeLoggerToOstreamAdapter` — делает тоже самое что `PlainLoggerToOstreamAdapter`, но добавляет в вывод время, когда событие произошло.

- Создано три класса реализующие паттерн «Прокси»: они перехватывают вызовы к классу Game, вызывают соответствующий оригинальный метод класса Game, затем вызывают метод объекта Logger. Эти три класса:
 - GameProxyWithLogging - работает с интерфейсом Logger, передавая ему перехваченные события для логирования.
 - GameProxyWithConsoleLogging - наследник GameProxyWithLogging, отличается тем, что создает один из двух наследников Logger, передавая в качестве std::ostream std::cout.
 - GameProxyWithFileLogging - наследник GameProxyWithLogging, отличается тем, что создает один из двух наследников Logger, передавая в качестве std::ostream открытый файл. Взаимодействие с файлом следует идиоме RAII.
- Была модифицирована функция main, чтобы программа реагировала на параметры переданные ей при запуске:
 - «-t» - приводит к добавлению времени в логи.
 - «-c» - приводит к выводу логов в консоль.
 - «-f» - приводит к выводу логов в файл «log.txt».
- Были построены UML диаграммы с помощью программы Doxygen, посмотреть их можно в браузере, открыв файл doxygen.zip/html/index.html и проследовав по ссылкам к нужным классам.

Выводы.

В ходе выполнения работы был разработан и реализован набор классов для ведения логирования действий и состояний программы. Программа успешно осуществляет вывод информации о игровых событиях как в консоль, так и в файл. Был реализован паттерн «Прокси», «Адаптер», выполнены требования идиомы RAII.