

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

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Объектно-ориентированное программирование»
Тема: «Логирование, перегрузка операций»

Студент гр. 1384

Белокобыльский И. В.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2022

Цель работы.

Научиться реализовывать логирование имеющихся классов, а также перегружать операции приведения к типу и вывода в поток

Задание.

Реализовать класс/набор классов отслеживающих изменения состояний в программе. Отслеживание должно быть 3-х уровней:

Изменения состояния игрока и поля, а также срабатывание событий

Состояние игры (игра начата, завершена, сохранена, и.т.д.)

Отслеживание критических состояний и ошибок (поле инициализировано с отрицательными размерами, игрок попытался перейти на непроходимую клетку, и.т.д.)

Реализованы классы для вывода информации разных уровней для в консоль и в файл с перегруженным оператором вывода в поток.

Требования:

Разработан класс/набор классов, отслеживающий изменения разных уровней

Разработаны классы для вывода в консоль и файл с соблюдением идиомы RAII и перегруженным оператором вывода в поток.

Разработанные классы спроектированы таким образом, чтобы можно было добавить новый формат вывода без изменения старого кода (например, добавить возможность отправки логов по сети)

Выбор отслеживаемых уровней логирования должен происходить в runtime

В runtime должен выбираться способ вывода логов (нет логирования, в консоль, в файл, в консоль и файл)

Примечания:

Отслеживаемые сущности не должны ничего знать о сущностях, которые их логируют

Уровни логирования должны быть заданными отдельными классами или перечислением

Разные уровни в логах должны помечаться своим префиксом

Рекомендуется сделать класс сообщения

Для отслеживания изменений можно использовать наблюдателя

Для вывода сообщений можно использовать адаптер, прокси и декоратор

Выполнение работы.

Для реализации логирования был выбран шаблон проектирования «Заместитель». Были созданы прокси для игрока, поля, контроллера состояния игры и событий. Они выступают в роли наблюдателя, передавая сообщение – объект класса `LoggerMessage` – во время изменений в классы, реализующие интерфейс `ISubscriber`.

Сами заместители добавляются на объекты в `GameMediator`, который вызывает метод `getProху` класса `LoggerController` для каждого логируемого объекта в методах `init` для игрока и контроллера состояния, `setField` для поля и `addCellEvent` для события. Таким образом от классов, не участвующих в логировании, скрывается реализация создания объектов и добавления прокси к ним.

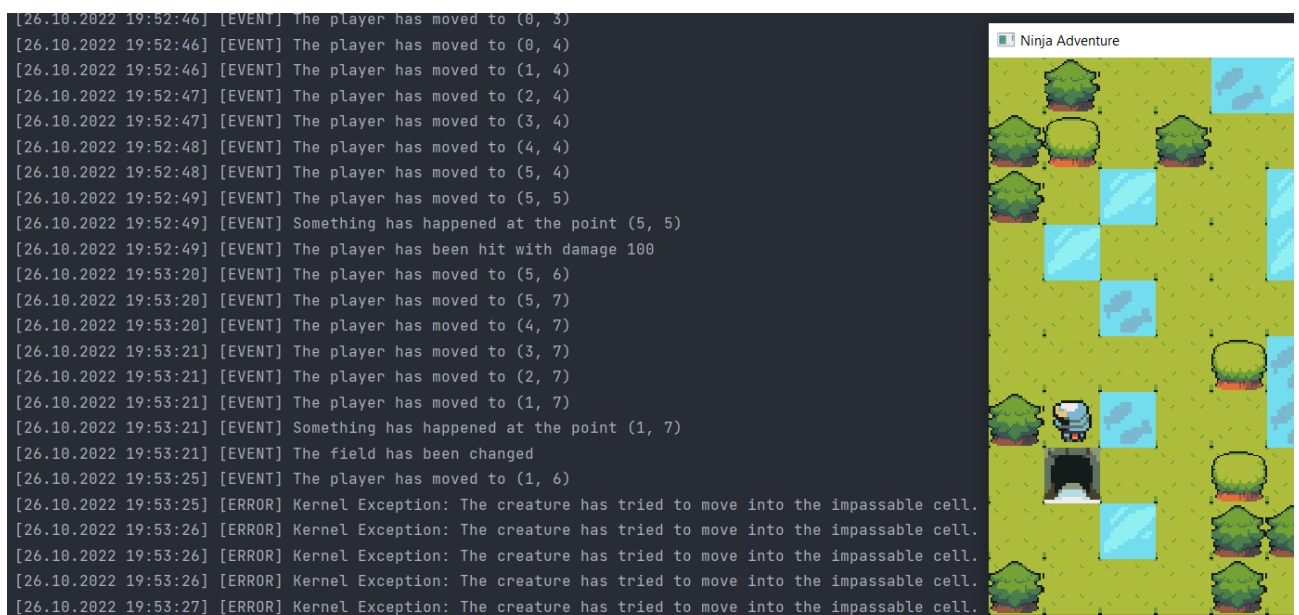
В качестве классов, реализующих интерфейс `ISubscriber` выступают `ConsoleLogger` и `FileLogger`, которые логируют в консоль и файл соответственно. Они принимают на вход сообщение, которое выводят в необходимый поток при условии, что просматриваемый уровень соответствует уровню сообщения. Префикс даты с временем и конкретного события добавляется в конструкторе класса `LoggerMessage`.

Тестирование программы.

1. Проверка логирования событий



2. Проверка логирования ошибок



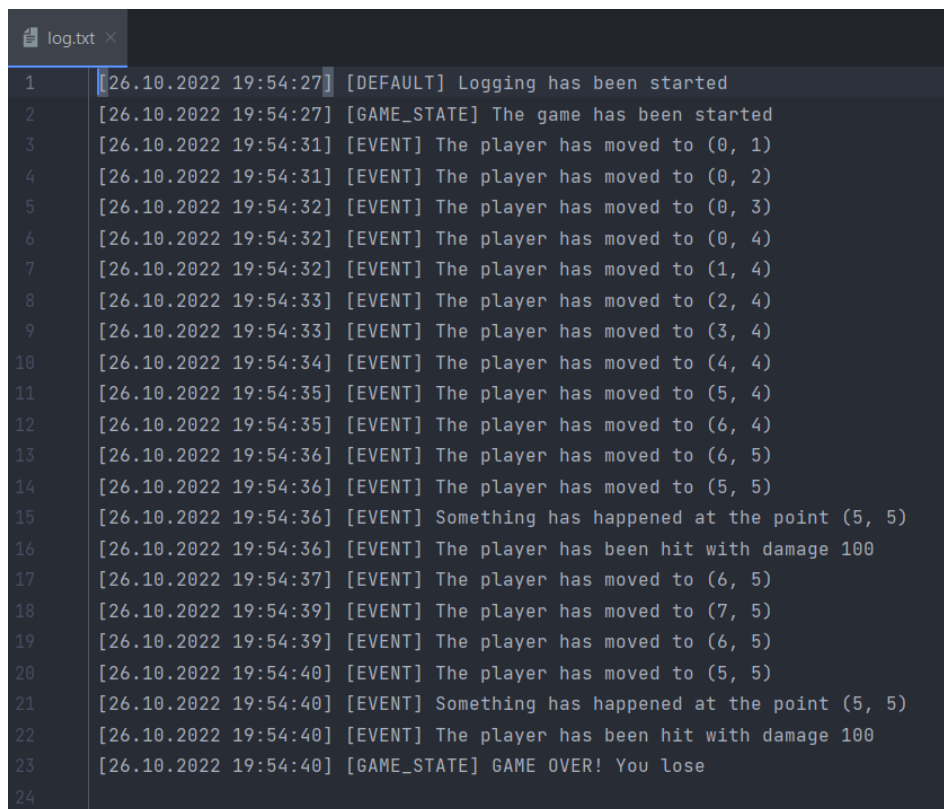
3. Проверка логирования состояния игры

```

[26.10.2022 19:54:27] [GAME_STATE] The game has been started
[26.10.2022 19:54:31] [EVENT] The player has moved to (0, 1)
[26.10.2022 19:54:31] [EVENT] The player has moved to (0, 2)
[26.10.2022 19:54:32] [EVENT] The player has moved to (0, 3)
[26.10.2022 19:54:32] [EVENT] The player has moved to (0, 4)
[26.10.2022 19:54:32] [EVENT] The player has moved to (1, 4)
[26.10.2022 19:54:33] [EVENT] The player has moved to (2, 4)
[26.10.2022 19:54:33] [EVENT] The player has moved to (3, 4)
[26.10.2022 19:54:34] [EVENT] The player has moved to (4, 4)
[26.10.2022 19:54:35] [EVENT] The player has moved to (5, 4)
[26.10.2022 19:54:35] [EVENT] The player has moved to (6, 4)
[26.10.2022 19:54:36] [EVENT] The player has moved to (6, 5)
[26.10.2022 19:54:36] [EVENT] The player has moved to (5, 5)
[26.10.2022 19:54:36] [EVENT] Something has happened at the point (5, 5)
[26.10.2022 19:54:36] [EVENT] The player has been hit with damage 100
[26.10.2022 19:54:37] [EVENT] The player has moved to (6, 5)
[26.10.2022 19:54:39] [EVENT] The player has moved to (7, 5)
[26.10.2022 19:54:39] [EVENT] The player has moved to (6, 5)
[26.10.2022 19:54:40] [EVENT] The player has moved to (5, 5)
[26.10.2022 19:54:40] [EVENT] Something has happened at the point (5, 5)
[26.10.2022 19:54:40] [EVENT] The player has been hit with damage 100
[26.10.2022 19:54:40] [GAME_STATE] GAME OVER! You lose
Ops.. You lose...

```

4. Проверка логирования в файл



```

log.txt x
1 [26.10.2022 19:54:27] [DEFAULT] Logging has been started
2 [26.10.2022 19:54:27] [GAME_STATE] The game has been started
3 [26.10.2022 19:54:31] [EVENT] The player has moved to (0, 1)
4 [26.10.2022 19:54:31] [EVENT] The player has moved to (0, 2)
5 [26.10.2022 19:54:32] [EVENT] The player has moved to (0, 3)
6 [26.10.2022 19:54:32] [EVENT] The player has moved to (0, 4)
7 [26.10.2022 19:54:32] [EVENT] The player has moved to (1, 4)
8 [26.10.2022 19:54:33] [EVENT] The player has moved to (2, 4)
9 [26.10.2022 19:54:33] [EVENT] The player has moved to (3, 4)
10 [26.10.2022 19:54:34] [EVENT] The player has moved to (4, 4)
11 [26.10.2022 19:54:35] [EVENT] The player has moved to (5, 4)
12 [26.10.2022 19:54:35] [EVENT] The player has moved to (6, 4)
13 [26.10.2022 19:54:36] [EVENT] The player has moved to (6, 5)
14 [26.10.2022 19:54:36] [EVENT] The player has moved to (5, 5)
15 [26.10.2022 19:54:36] [EVENT] Something has happened at the point (5, 5)
16 [26.10.2022 19:54:36] [EVENT] The player has been hit with damage 100
17 [26.10.2022 19:54:37] [EVENT] The player has moved to (6, 5)
18 [26.10.2022 19:54:39] [EVENT] The player has moved to (7, 5)
19 [26.10.2022 19:54:39] [EVENT] The player has moved to (6, 5)
20 [26.10.2022 19:54:40] [EVENT] The player has moved to (5, 5)
21 [26.10.2022 19:54:40] [EVENT] Something has happened at the point (5, 5)
22 [26.10.2022 19:54:40] [EVENT] The player has been hit with damage 100
23 [26.10.2022 19:54:40] [GAME_STATE] GAME OVER! You lose
24

```

5. Проверка отсутствия логирования



6. Проверка выбора уровня логирования



Выводы.

В рамках данной лабораторной работы была доработана программа на языке C++ из предыдущей работы. Реализован механизм логирования в консоль и в файл изменений при помощи шаблонов проектирования «Прокси» и «Наблюдатель».

ПРИЛОЖЕНИЕ 1

UML-диаграмма классов:

