

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

Студент гр. 8381

Преподаватель

Почаев Н.А.

Жангиров Т.Р.

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.

Задание.

- Выполнены основные требования к логированию
- Реализована возможность записи логов в файл
- Реализована возможность записи логов в терминал
- Взаимодействие с файлами должны 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.

Написание работы производилось на базе операционной системы Windows 10 в среде разработки Qt Creator, для компиляции и отладки использовалась UNIX-подобная среда Cygwin и набор адаптированных инструментов MiniGW. Были задействованы пакеты GCC, CMake, а также GDB. Для компиляции текущей версии программы под Windows необходим MinGW 8.10 (для более полноценной поддержки C++17) и Qt версии 14.10 и выше.

Реализованные классы

Классы, добавленные в программу в данной лабораторной работе и их функционал представлены в табл. 1. В ней приведено общее описание классов, отдельные моменты пояснены в комментариях к коду.

Таблица 1 – Основные добавленные классы

| Класс | Назначение |
|--------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Time (./Game/Logging) | Класс используется для работы со временем, вывода его в представлении “Y-m-d H:M:S” в приёмники логирования, а также хранения временной отметки начала работы. В данном классе выполнена <i>перегрузка оператора вывода в поток</i> , для читабельного вывода в аналогично перегруженном операторе вывода последующего класса. |
| ILogger (./Game/Logging) | Интерфейс, описывающий основные операции, для логгеров. |
| TermialLogger (./Game/Logging/Loggers) | Класс логгера для записи в терминал. Содержит <i>перегрузку оператора вывода в поток</i> . |
| FileLogger (./Game/Logging/Loggers) | Класс логгера для записи в файл. Содержит <i>перегрузку оператора вывода в поток</i> . Взаимодействие с файлом логирования LOG происходит по идиоме RAII – создаётся и инициализируется в конструкторе, а закрывается на запись в деструкторе. |
| ProxyLogger (./Game/Logging/Loggers) | Класс, реализующий паттерн “Прокси” для возможности смены логгера “на лету”. |
| ILoggerAdapter (./Game/Logging/Loggers) | Интерфейс, описывающий основные операции класса адаптера для логгеров. |
| ILoggerAdapterStringFormer (./Game/Logging/Loggers) | Интерфейс, описывающий основной операции формирования строк логирования для класса адаптера. |
| LogAdapter (./Game/Logging/Loggers) | Класс, реализующий паттерн “Адаптер” для выбора различных режимов вывода логов: STANDART – обычный и ADVANCED – расширенный, с полной информацией, требуемой для выполнения операции. |

Выводы.

В ходе выполнения лабораторной работы были написаны требуемые классы, а также реализовано логирование основных действий пользователя и событий игры.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ. MAIN.CPP

```
#include <iostream>

#include <QApplication>
#include <QGridLayout>
#include <QWidget>
#include <QLabel>
#include <QScreen>

#include "Tests/examples.h"
#include "Game/UIFacade.h"

int main(int argc, char *argv[])
{
    std::shared_ptr<UIFacade> game = std::make_shared<UIFacade>(argc, argv);
    game->start();

    return 0;
}
```