

**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. 8383

Дейнега В.Е.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2019

## Цель работы.

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

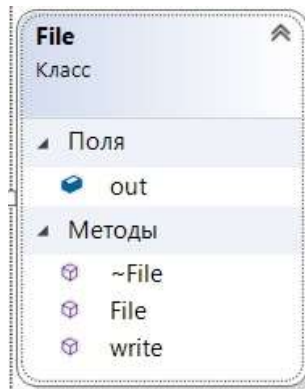
## Постановка задачи.

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

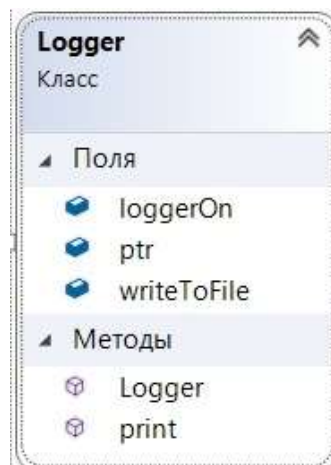
- Логирование действий пользователя
- Логирование действий юнитов и базы

## Ход работы.

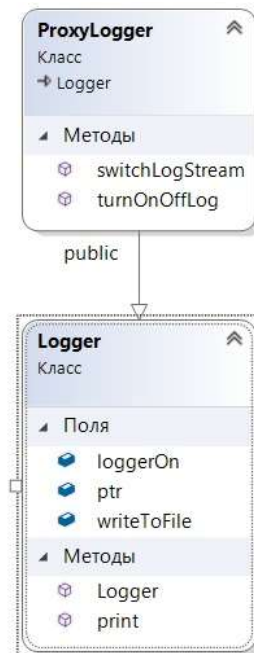
0) Для взаимодействий с файлами был написан класс File(File.h/cpp)



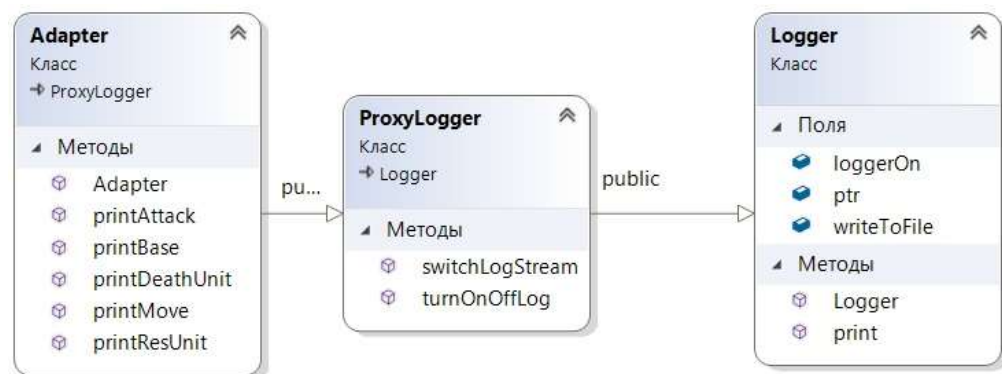
1) Переключение между разным логированием реализовано с помощью паттерна прокси. Класс прокси является наследником класса Logger(Logger.h/cpp), класс логгер содержит поля-флаги вида печати логов, указатель на файл и метод печати логов.



Класс ProxyLogger(ProxyLogger.h/cpp) наследуется от Logger'а и имеет методы, в которых в зависимости от входных параметров меняет поля-флаги в родительском классе.



2) Реализован разный формат записи при помощи паттерна адаптер. Класс Adapter(Adapter.h/cpp) наследуется от класса ProxyLogger, имеет несколько методов, формирующих строку лога и скармливающих эту строку методу класса Logger.



2) Были перегружены операторы вывода в поток для юнитов и базы.

```

std::ostream& operator<<(std::ostream& out, const Base& base) {
    out << "Base with x = " + std::to_string(base.x) + " y = " + std::to_string(base.y) + " have " + std::to_string(base.health) + " health.\n";
    return out;
}

std::ostream& operator<<(std::ostream& out, Unit& unit) {
    out << unit.getName() + " respowned at x = " + std::to_string(unit.x) + " y = " + std::to_string(unit.y) + "\n";
    return out;
}
  
```

## Примеры работы.

Респавн базы в клетке 3 0, респавн лучника в клетке 0 0

```
type N:
7
Type coords of base:
3 0
-----
|  0  ||  0  ||  0  ||  В  ||  0  ||  0  ||  0  ||
|-----|
|  0  ||  0  ||  0  ||  0  ||  0  ||  0  ||  0  ||
|-----|
|  ^  ||  0  ||  0  ||  0  ||  0  ||  0  ||  0  ||
|-----|
|  0  ||  0  ||  0  ||  0  ||  ^  ||  0  ||  0  ||
|-----|
|  0  ||  ~  ||  0  ||  0  ||  0  ||  0  ||  0  ||
|-----|
|  0  ||  0  ||  0  ||  0  ||  0  ||  0  ||  0  ||
|-----|
|  0  ||  0  ||  0  ||  0  ||  0  ||  0  ||  0  ||
|-----|
NEW TURN!

Type x of chosen cell
Type y of chosen cell
0 0
Cell is clear you can spawn:
Press 1 to spawn archer
Press 2 to spawn ballista
Press 3 to spawn farm tower
Press 5 to spawn bomber
Press 6 to spawn general-bomber
Press 0 to end turn
1
```

Респавн в клетке 1 0 еще одного лучника

```
-----
|  1  ||  0  ||  0  ||  В  ||  0  ||  0  ||  0  ||
|-----|
|  0  ||  0  ||  0  ||  0  ||  0  ||  0  ||  0  ||
|-----|
|  ^  ||  0  ||  0  ||  0  ||  0  ||  0  ||  0  ||
|-----|
|  0  ||  0  ||  0  ||  0  ||  ^  ||  0  ||  0  ||
|-----|
|  0  ||  ~  ||  0  ||  0  ||  0  ||  0  ||  0  ||
|-----|
|  0  ||  0  ||  0  ||  0  ||  0  ||  0  ||  0  ||
|-----|
|  0  ||  0  ||  0  ||  0  ||  0  ||  0  ||  0  ||
|-----|
NEW TURN!

Type x of chosen cell
Type y of chosen cell
1 0
Cell is clear you can spawn:
Press 1 to spawn archer
Press 2 to spawn ballista
Press 3 to spawn farm tower
Press 5 to spawn bomber
Press 6 to spawn general-bomber
Press 0 to end turn
1
```

Выстрел первого лучника по второму, переход первого лучника в клетку 1 6

```

| 1 || 1 || 0 || в || 0 || 0 || 0 ||
| 0 || 0 || 0 || 0 || 0 || 0 || 0 ||
| ^ || 0 || 0 || 0 || 0 || 0 || 0 ||
| 0 || 0 || 0 || 0 || ^ || 0 || 0 ||
| 0 || ~ || 0 || 0 || 0 || 0 || 0 ||
| 0 || 0 || 0 || 0 || 0 || 0 || 0 ||
| 0 || 0 || 0 || 0 || 0 || 0 || 0 ||
NEW TURN!

Type x of chosen cell
Type y of chosen cell
0 0
You pick archer! You can relocate it or make shot.
1 - shot; 2 - relocate:
Type x and y:
1
Type coordinats of target:
1 0
Shot made successfully!
Health of uint is: 0
Для продолжения нажмите любую клавишу . . .
Press 2 to relocate, 0 to end turn
2
Type x and y:
1 6

```

Отрисовка перехода

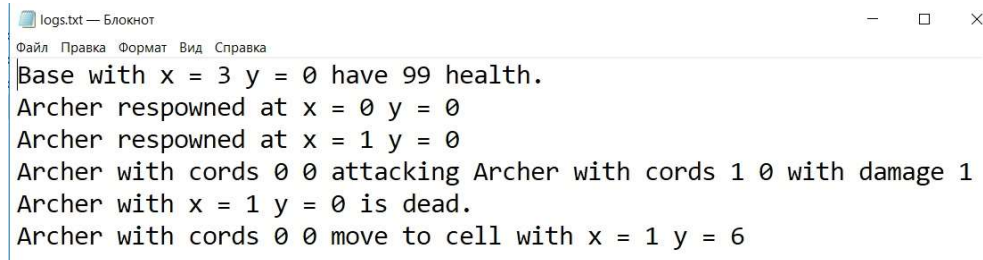
```

| 0 || 0 || 0 || в || 0 || 0 || 0 ||
| 0 || 0 || 0 || 0 || 0 || 0 || 0 ||
| ^ || 0 || 0 || 0 || 0 || 0 || 0 ||
| 0 || 0 || 0 || 0 || ^ || 0 || 0 ||
| 0 || ~ || 0 || 0 || 0 || 0 || 0 ||
| 0 || 0 || 0 || 0 || 0 || 0 || 0 ||
| 0 || 1 || 0 || 0 || 0 || 0 || 0 ||
NEW TURN!

Type x of chosen cell
Type y of chosen cell

```

## Вывод логов после сессии



The screenshot shows a Notepad window titled 'logs.txt — Блокнот'. The menu bar includes 'Файл', 'Правка', 'Формат', 'Вид', and 'Справка'. The text content of the log is as follows:

```
Base with x = 3 y = 0 have 99 health.  
Archer respowned at x = 0 y = 0  
Archer respowned at x = 1 y = 0  
Archer with cords 0 0 attacking Archer with cords 1 0 with damage 1  
Archer with x = 1 y = 0 is dead.  
Archer with cords 0 0 move to cell with x = 1 y = 6
```

### Выводы.

В ходе выполнения работы были создан набор классов для ведения логирования действий и состояний программы.