

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: Интерфейсы классов; взаимодействие классов; перегрузка
операций.

Студент гр. 8383

Дейнега В.Е.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

Цель работы.

Научиться создавать интерфейсы классов, реализовать их взаимодействие с помощью паттернов, перегрузки операций.

Постановка задачи.

Разработать и реализовать набор классов:

- Класс базы
- Набор классов ландшафта карты
- Набор классов нейтральных объектов поля

Класс базы должен отвечать за создание юнитов, а также учитывать юнитов, относящихся к текущей базе. Основные требования к классу база:

- База должна размещаться на поле
- Методы для создания юнитов
- Учет юнитов, и реакция на их уничтожение и создание
- База должна обладать характеристиками такими, как здоровье, максимальное количество юнитов, которые могут быть одновременно созданы на базе, и.т.д.

Набор классов ландшафта определяют вид поля. Основные требования к классам ландшафта:

Должно быть создано минимум 3 типа ландшафта

- Все классы ландшафта должны иметь как минимум один интерфейс
- Ландшафт должен влиять на юнитов (например, возможно пройти по клетке с определенным ландшафтом или запрет для атаки определенного типа юнитов)
- На каждой клетке поля должен быть определенный тип ландшафта

Набор классов нейтральных объектов представляют объекты, располагаемые на поле и с которыми могут взаимодействовать юниты. Основные требования к классам нейтральных объектов поля:

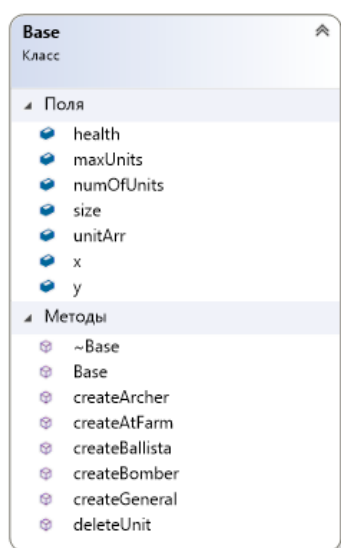
- Создано не менее 4 типов нейтральных объектов

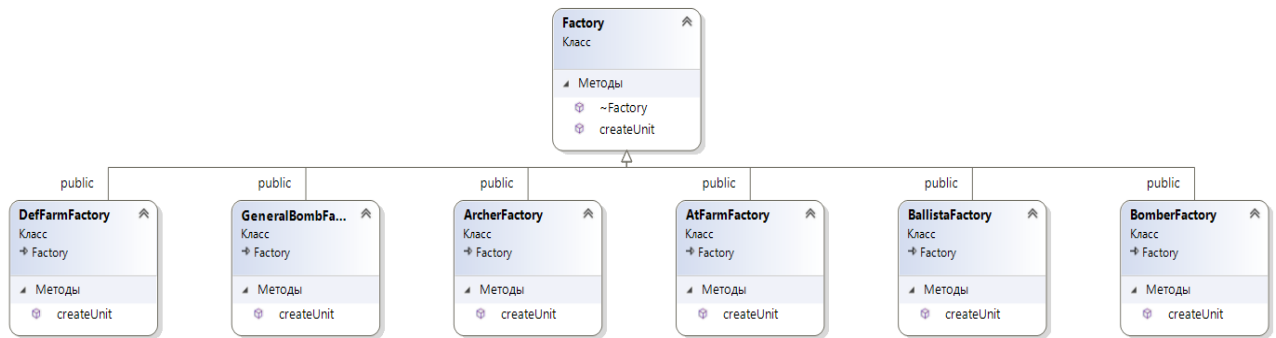
- Взаимодействие юнитов с нейтральными объектами, должно быть реализовано в виде перегрузки операций
- Классы нейтральных объектов должны иметь как минимум один общий интерфейс
- **Для взаимодействия одного типа нейтрального объекта с разными типами юнитов используется паттерн “Стратегия”*
- **Для взаимодействия ландшафта с юнитами используется паттерн “Прокси”*

Ход работы.

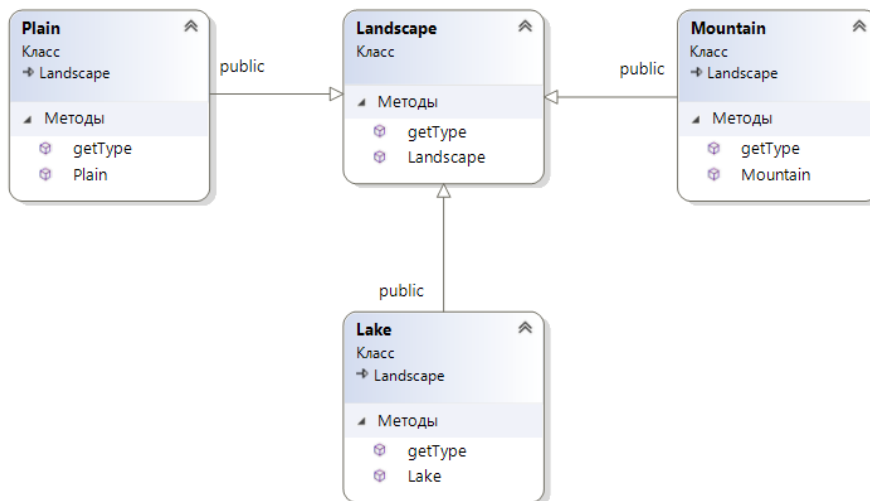
0) Файлы, добавленные в этой лабораторной работе Neutral.h/cpp, NeutralFactory.h/cpp, Factory.h/cpp, Base.h/cpp, Landscape.h/cpp.

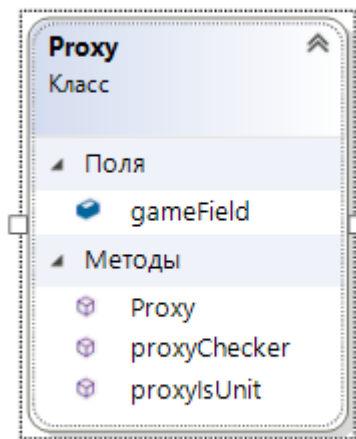
1) Был создан класс Base, содержащий методы для создания юнитов, эти методы проверяют количество юнитов на поле и с помощью фабрики юнитов(добавлено в этой лабораторной работе) создают юнит нужного типа. Также Base содержит метод для удаления юнитов. Поля int x, int y, указывают местоположение Base на игровом поле. У базы есть здоровье; поле, показывающее сколько юнитов максимально может создать база int maxUnits и вектор указателей на юниты.



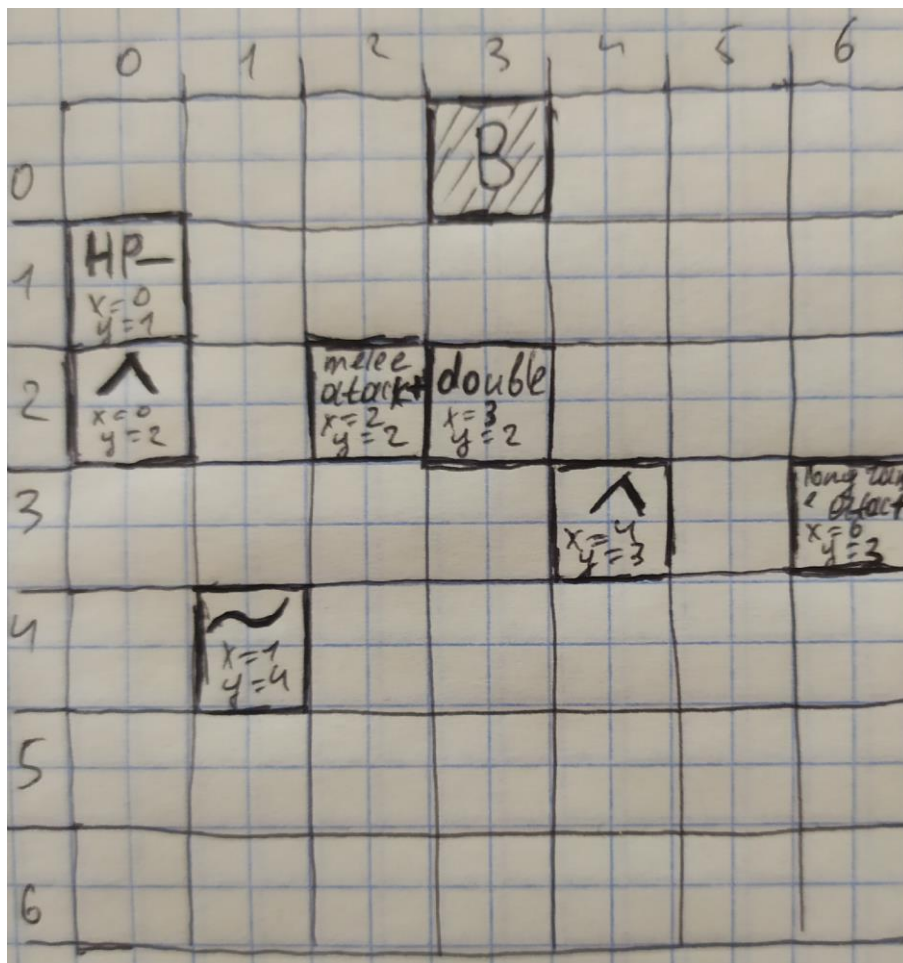


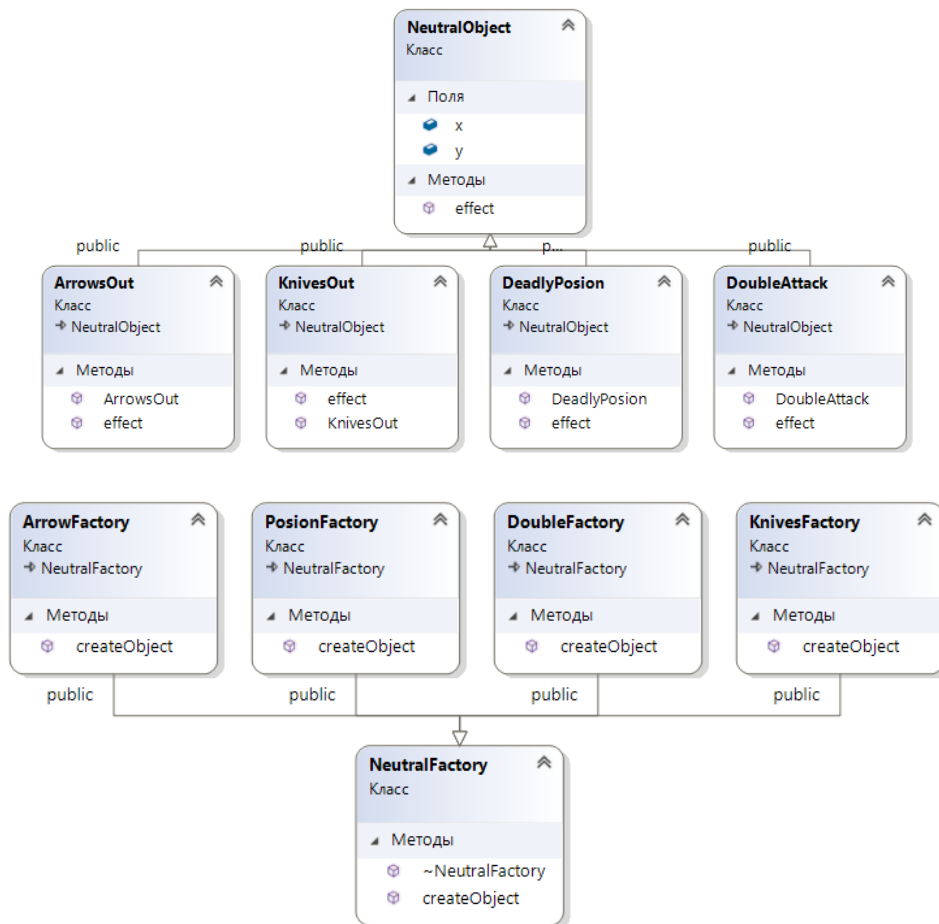
2) Был создан набор классов ландшафта. Абстрактный класс Landscape содержит чистую виртуальную функцию `int getType()`, от Landscape наследуются классы-ландшафты Lake, Plain и Mountain. В каждой клетке поля содержится указатель на 1 из 3 элементов ландшафта. Для взаимодействия с ландшафтом был написан класс Proxy (находится в файле GameField.h), содержащий указатель на игровое поле и имеющий методы `bool ProxyChecker(int x, int y)`, проверяющий можно ли поставить на клетку с данными координатами юнит (`true` - можно, `false` - нет) и метод `bool proxyIsUnit(int x, int y)`, проверяющий стоит ли на клетке юнит. На клетки с ландшафтами Lake и Mountain вставлять нельзя. ~ на игровом поле означает озеро, ^ - гора, 0 означает, что ландшафт клетки равнина и на клетке нет юнита.



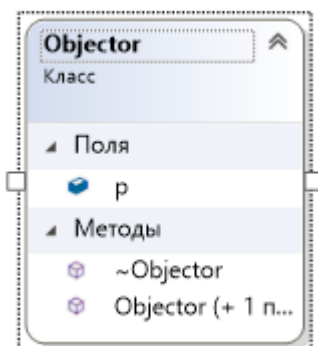


3) Были созданы 4 типа нейтральных объектов и фабрика нейтральных объектов. (ArrowsOut прибавляет +2 к атаке long range юнитам, KnivesOut +2 к атаке melee юнитам, DeadlyPoison -1 к здоровью, DoubleAttack удваивает атаку). Нейтральные объекты не видны игроку, пока что на карте их по 1 штуке и их местоположение постоянно. Карта поля:





Для взаимодействия нейтральных объектов с юнитами был перегружен оператор $+$ и произведена попытка реализовать паттерн стратегия. Был написан класс `Objector` (находится в файле `Units.h`), полем которого является `NeutralObject*` `p`. Для класса `Objector` перегружается оператор $+$, который в зависимости от возвращаемого значения метода `int effect()` производит с юнитами преобразования.



4) Лучник и баллиста теперь умеют стрелять и даже убивать юнитов! После попадания будет выведено сообщение о попадании, а также хп

раненого юнита, если юнит убит ($xhp \leq 0$), он пропадет с поля, но отрисовка пропадания случится после окончания хода.

Спавн юнита на клетке 0 0:

```

type N:
6
-----
| 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 || 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

```

Спавн второго юнита на клетке 1 0:

```

-----
| 1 || 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 || 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
3

```

Выстрел лучником по фарм-башне:


```

| 1 || 3 || 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 || 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 coordinates of target:
1 0
Shot made successfully!
Health of unit is: 0
Для продолжения нажмите любую клавишу . . .

```

Исчезновение убитого юнита с карты:

```

| 1 || 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 || 0 || 0 || 0 ||
-----
NEW TURN!

Type x of chosen cell
Type y of chosen cell

```

5)

Выводы.

В ходе выполнения работы были созданы наборы классов ландшафта и нейтральных объектов, а также база, произведена попытка реализовать паттерны стратегия и прокси. Была реализована фабрика юнитов из первой лабораторной работы.