

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

ОТЧЕТ
по практической работе № 2
по дисциплине «ООП»
Тема: Интерфейсы классов; взаимодействие классов; перегрузка
операций

Студент гр. 8383

Степанов В.Д.

Преподаватель

Жангиров Т. Р.

Санкт-Петербург

2020

Цель работы.

Научиться работать с интерфейсами классов, взаимодействием классов, и перегрузкой арифметических операций в языке C++.

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

Выполнение работы.

1. Был реализован класс базы. База первого игрока размещается в левом нижнем угле карты, а второго в верхнем правом. База имеет метод создания юнитов `Unit* createUnit(int whatUnit, bool isProtected)`, который создает определенного юнита. Так же имеется `bool setFirstUnits (Field *field)` и `bool setNeutralObjects (Field *field)`, которые расставляют юнитов и нейтральные объекты на поле соответственно. Учет количества юнитов ведется в переменной `int countUnit`, все юниты созданные базой, хранятся с массиве `Unit * units[9]`, максимальное хранимы юнитов не должно превышать 9.

1.1. Для хранения характеристик юнитов используется паттерн “Легковес”. UML-диаграмма реализации представлена на рисунке 1.

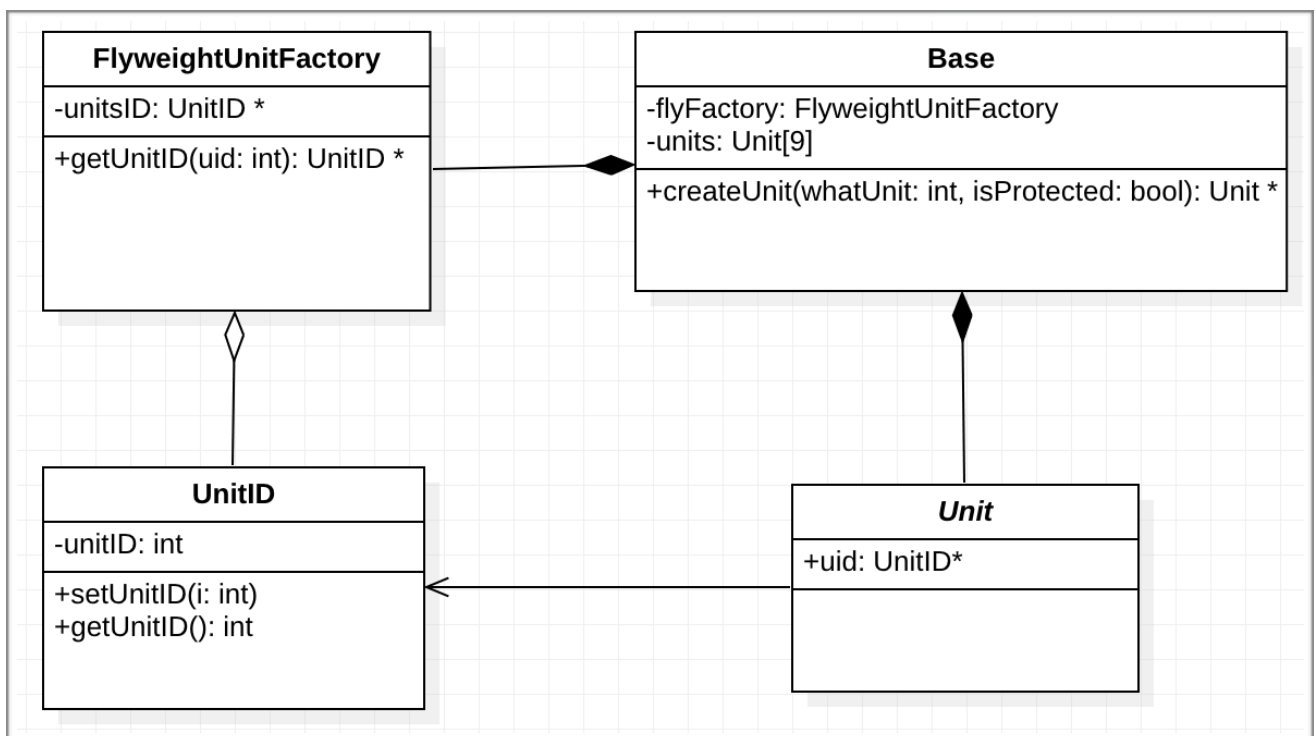


Рисунок 1 – UML-диаграмма реализации паттерна “Легковес”

1.2. Для наблюдения над юнитами в классе база используется паттерн “Наблюдатель”. UML-диаграмма реализации представлена на рисунке 2.

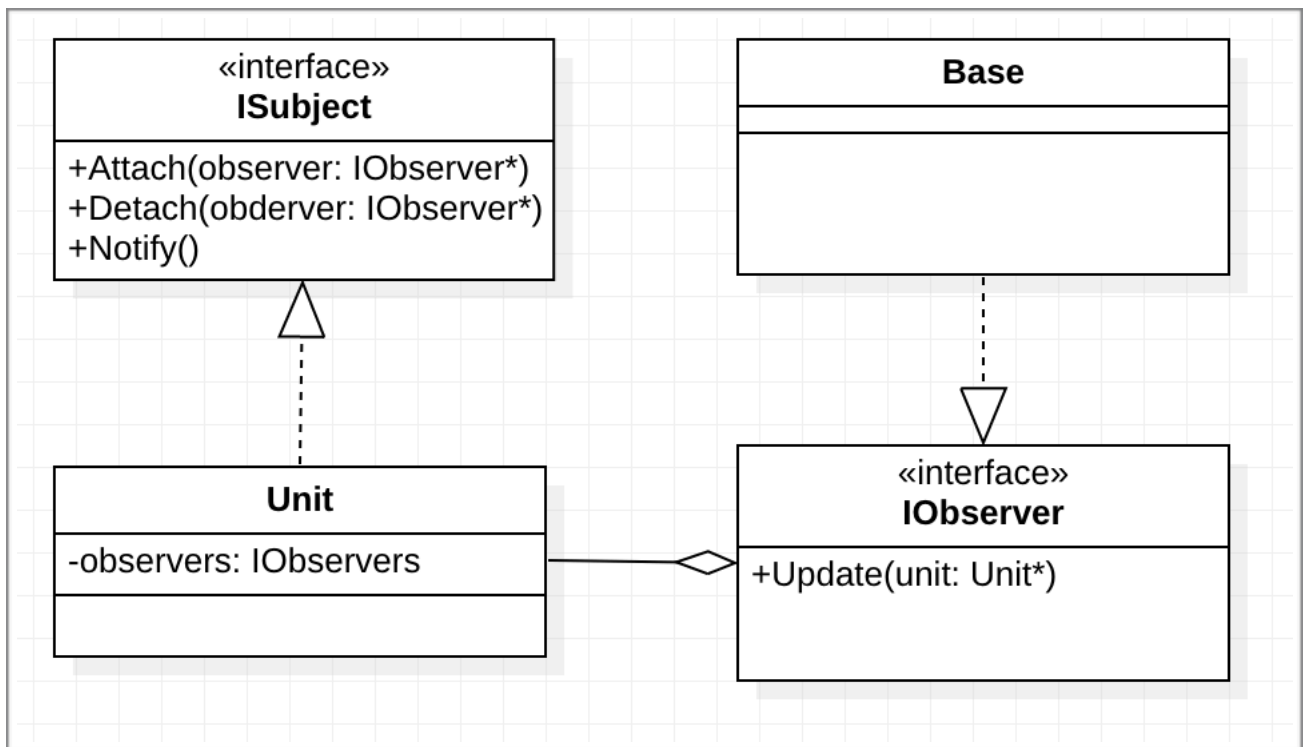


Рисунок 2 – UML-диаграмма реализации паттерна “Наблюдатель”

2. Было реализовано 3 типа ландшафта: Ground, Brush и DeepPuddel. Они имеют один интерфейс Landscap. Каждая клетка поля имеет ландшафт. В большинстве клеток, кроме зон спавна юнитов, ландшафт определяется случайно. Grount с вероятностью 80%, Brush с вероятностью 10% и DeepPuddel с вероятностью 10%. Разный ландшафт влияет на юнитов по-разному. UML-диаграмма реализации ландшафта представлена на рисунке 3.

3. Для взаимодействия ландшафта с юнитам используется паттерн “Прокси”. UML-диаграмма реализации паттерна представлена на рисунке 4.

3. Создано 4 типа нейтральных объектов: HelthPound, VoxWithArmor. PowerChemical, WorkbenchForWeapons. Все классы имеют общий интерфейс NeutralObject. С нейтральными объектами могу взаимодействовать юниты Shooter и Runner, юниту Smoker взаимодействие запрещено. UML-диаграмма реализации нейтральных объектов представлена на рисунке 5. Взаимодействие происходит через перегрузку оператора “+=”. Для того чтобы юнит взаимодействовал с нейтральным объектом юнит должен находится сверху, снизу, справа или слева от объекта. После игрок должен

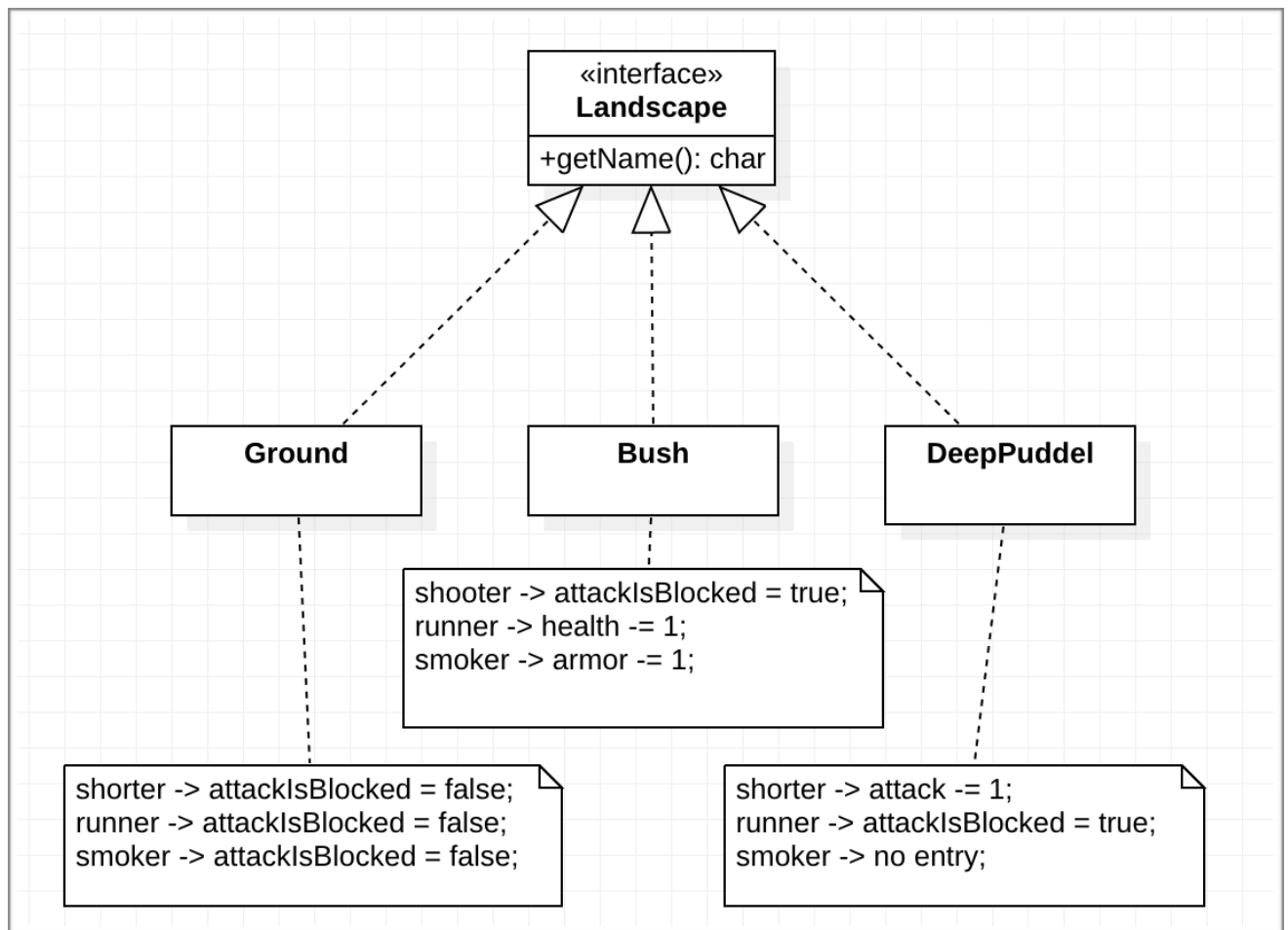


Рисунок 3 – UML-диаграмма реализации ландшафта

прописать команду вида – <имя юнита><space><направление (wasd)><‘h’> (например: !1 wh). Для перемещения юнитов используется та же схема, но в конце вместо ‘h’ печется ‘m’. В данной версии игры реализация паттерна “Стратегия” была удалена.

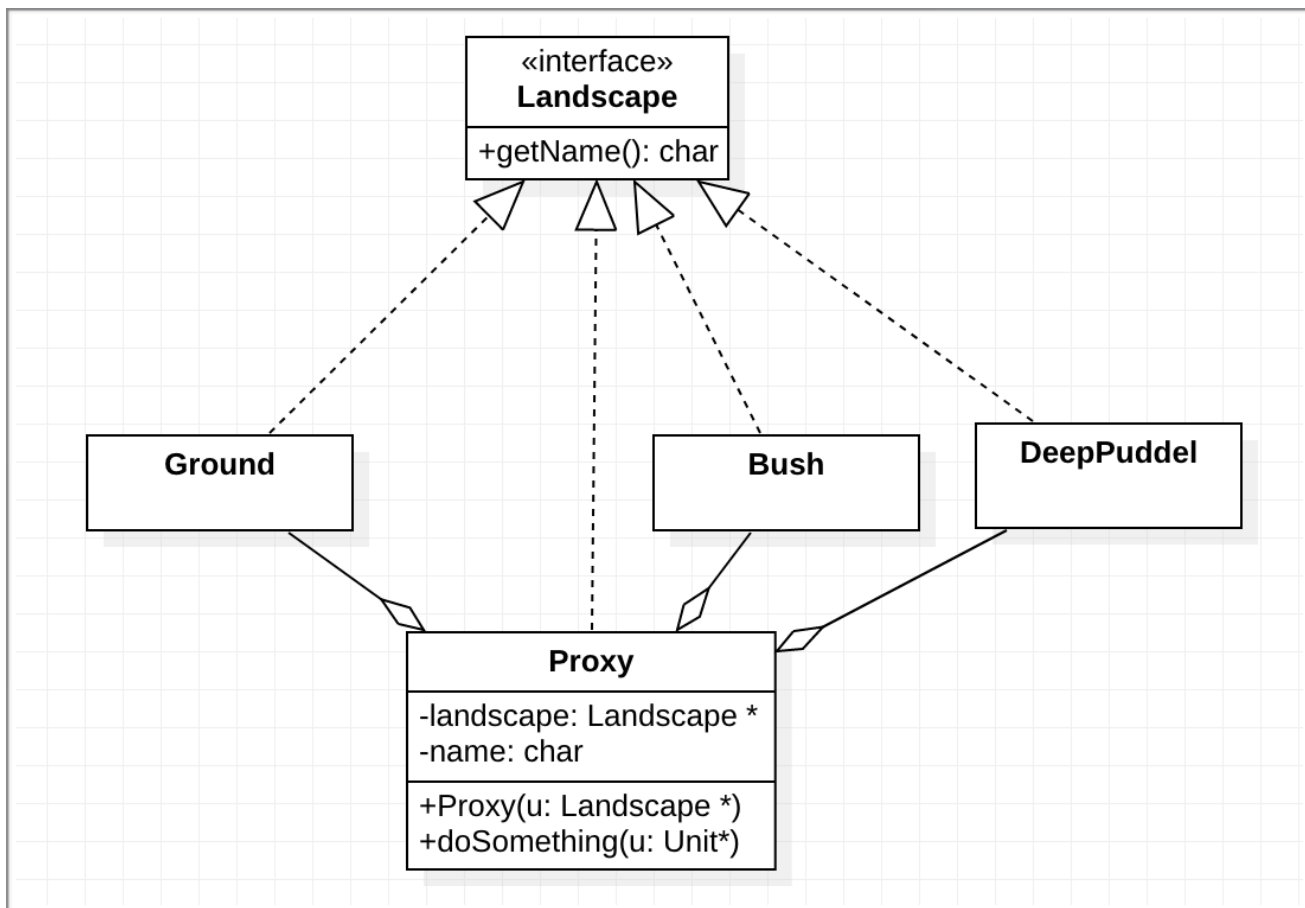


Рисунок 4 – UML-диаграмма реализации паттерна “Прокси”

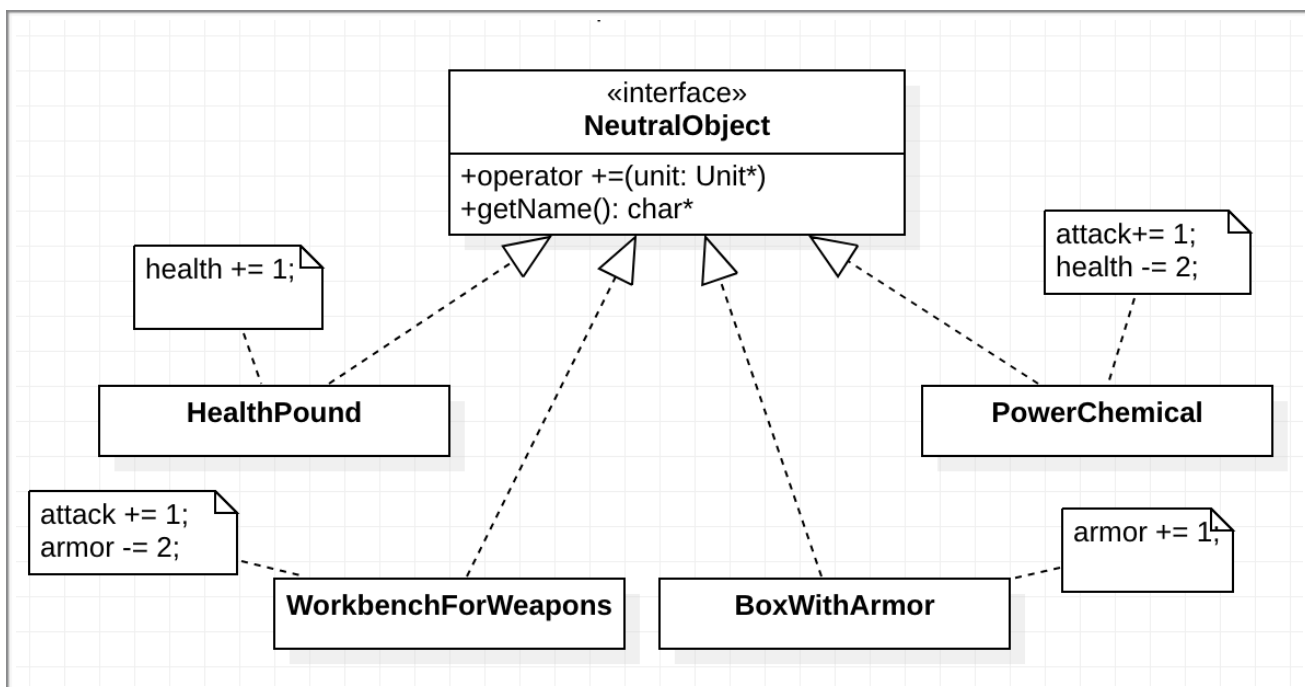


Рисунок 5 – UML-диаграмма реализации нейтральных объектов

Демонстрация работы программы.

Работа программы представлена на рисунках 6 - 8.

```

-----
|  BA  <1  B1 |
| HP    !5  $1 |
|  ##    ##  !3  <3 |
|                    !1 |
|          WWPC## |
|  ((    PCWW |
|  6! |
| 2>  4!  ((    (( |
|  2$  2!  ##    BA |
| B2  4>    ##  HP |
|-----
$1  h = 10  a = 10  p = 3  2$  h = 10  a = 10  p = 3
<1  h = 5   a = 5   p = 6  2>  h = 5   a = 5   p = 6
<3  h = 5   a = 5   p = 6  4>  h = 5   a = 5   p = 6
!1  h = 8   a = 13  p = 5  2!  h = 8   a = 13  p = 5
!3  h = 8   a = 13  p = 5  4!  h = 8   a = 13  p = 5
!5  h = 8   a = 13  p = 5  6!  h = 8   a = 13  p = 5
Введите имя персонажа и в какую сторону переместить или взаимодействовать (!5 w1)
!5 am
1
с
-----
|  BA  <1  B1 |
| HP    !5  $1 |
|  ##    ##  !3  <3 |
|                    !1 |
|          WWPC## |
|  ((    PCWW |
|  6! |
| 2>  4!  ((    (( |
|  2$  2!  ##    BA |
| B2  4>    ##  HP |
|-----
$1  h = 10  a = 10  p = 3  2$  h = 10  a = 10  p = 3
<1  h = 5   a = 5   p = 6  2>  h = 5   a = 5   p = 6
<3  h = 5   a = 5   p = 6  4>  h = 5   a = 5   p = 6
!1  h = 8   a = 13  p = 5  2!  h = 8   a = 13  p = 5
!3  h = 8   a = 13  p = 5  4!  h = 8   a = 13  p = 5
!5  h = 8   a = 13  p = 5  6!  h = 8   a = 13  p = 5
Введите имя персонажа и в какую сторону переместить или взаимодействовать (!5 w1)

```

Рисунок 6 – Пример передвижения юнита

```

c -----
| BA      <1 B1 |
| HP      !5  $1 |
| ##      ##  !3 <3 |
|          !1 |
|          WWPC## |
|          PCWW |
|          6! |
| 2> 4!    ((    (( |
| 2$ 2!    ##    BA |
| B2 4>    ##    HP |
|-----|
$1 h = 10 a = 10 p = 3  2$ h = 10 a = 10 p = 3
<1 h = 5 a = 5 p = 6  2> h = 5 a = 5 p = 6
<3 h = 5 a = 5 p = 6  4> h = 5 a = 5 p = 6
!1 h = 8 a = 13 p = 5  2! h = 8 a = 13 p = 5
!3 h = 8 a = 13 p = 5  4! h = 8 a = 13 p = 5
!5 h = 8 a = 13 p = 5  6! h = 8 a = 13 p = 5
Введите имя персонажа и в какую сторону переместить или взаимодействовать (!5 w1)
!5 sm
1
c -----
| BA      <1 B1 |
| HP      !5  $1 |
| ##      ##  !3 <3 |
|          !1 |
|          WWPC## |
|          PCWW |
|          6! |
| 2> 4!    ((    (( |
| 2$ 2!    ##    BA |
| B2 4>    ##    HP |
|-----|
$1 h = 10 a = 10 p = 3  2$ h = 10 a = 10 p = 3
<1 h = 5 a = 5 p = 6  2> h = 5 a = 5 p = 6
<3 h = 5 a = 5 p = 6  4> h = 5 a = 5 p = 6
!1 h = 8 a = 13 p = 5  2! h = 8 a = 13 p = 5
!3 h = 8 a = 13 p = 5  4! h = 8 a = 13 p = 5
!5 h = 7 a = 13 p = 5  6! h = 8 a = 13 p = 5
Введите имя персонажа и в какую сторону переместить или взаимодействовать (!5 w1)

```

Рисунок 7 – Пример взаимодействия юнинита и ландшафта

с											
BA			<1	B1							
HP				\$1							
##			##	!3	<3						
			!5	!1							
			WWPC##								
((PCWW								
6!											
2>	4!	((((
2\$	2!	##	BA								
B2	4>	##	HP								

\$1	h = 10	a = 10	p = 3	2\$	h = 10	a = 10	p = 3				
<1	h = 5	a = 5	p = 6	2>	h = 5	a = 5	p = 6				
<3	h = 5	a = 5	p = 6	4>	h = 5	a = 5	p = 6				
!1	h = 8	a = 13	p = 5	2!	h = 8	a = 13	p = 5				
!3	h = 8	a = 13	p = 5	4!	h = 8	a = 13	p = 5				
!5	h = 7	a = 13	p = 5	6!	h = 8	a = 13	p = 5				
Ввести имя персонажа и в какую сторону переместить или взаимодействовать (!5 w1)											
!5 sh											
1											
с											
BA			<1	B1							
HP				\$1							
##			##	!3	<3						
			!5	!1							
			WWPC##								
((PCWW								
6!											
2>	4!	((((
2\$	2!	##	BA								
B2	4>	##	HP								

\$1	h = 10	a = 10	p = 3	2\$	h = 10	a = 10	p = 3				
<1	h = 5	a = 5	p = 6	2>	h = 5	a = 5	p = 6				
<3	h = 5	a = 5	p = 6	4>	h = 5	a = 5	p = 6				
!1	h = 8	a = 13	p = 5	2!	h = 8	a = 13	p = 5				
!3	h = 8	a = 13	p = 5	4!	h = 8	a = 13	p = 5				
!5	h = 5	a = 13	p = 6	6!	h = 8	a = 13	p = 5				
Ввести имя персонажа и в какую сторону переместить или взаимодействовать (!5 w1)											

Рисунок 8 – Пример взаимодействия юината и нейтрального объекта

Выводы.

В ходе лабораторной работы был получен навык работы с интерфейсами классов, взаимодействием классов, и перегрузкой арифметических операций в языке C++