

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

Мирсков А.А.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2020

Цель работы.

Изучить работу с интерфейсами классов, способы взаимодействия классов и перегрузку операций в C++.

Задание

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 взаимодействие юнитов. 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

Выполнены основные требования к классу база	2 балла
Выполнены основные требования к набору классов ландшафта	2 балла
Выполнены основные требования к набору классов нейтр. объектов	2 балла
Добавлено взаимодействие юнитов	1 балла
Имеется 3+ демонстрационных примера	1 балл
Взаимодействие через перегрузку операторов	2 балла

<i>*Для хранения информации о юнитах в классе базы используется паттерн "Компоновщик"/ Использование "Легковеса" для хранения общих характеристик юнитов</i>	2 балла
<i>*Для наблюдения над юнитами в классе база используется паттерн "Наблюдатель"</i>	2 балла
<i>*Для взаимодействия ландшафта с юнитам используется паттерн "Прокси"</i>	3 балла
<i>*Для взаимодействия одного типа нейтрального объекта с разными типами юнитов используется паттерн "Стратегия"</i>	3 балла
Кол-во баллов за основные требования	10 баллов
Максимальное кол-во баллов за лаб. работу	20 баллов

Выполнение работы.

1)

- Был создан класс базы.
- База размещается на поле с помощью полей с координатами x и y, которые инициализируются в конструкторе.
- Класс базы хранит в себе юнитов в полу `Vector<Unit*> units` и имеет методы для создания (`addUnit`), удаления (`removeUnit`) и перемещения(`removeUnit`) юнитов.
- База учитывает количество юнитов и не позволяет создать больше юнитов, чем число в поле `maxUnit`.
- У базы есть здоровье, значение которого хранится в поле `health` и метод `takeDamage`, который позволяет ей получать урон.

2)

- Были созданы 3 класса ландшафта.
- Классы имеют общий интерфейс `Landscape`.
- Он влияет на юнитов. `Wall` не позволяет юнитам встать на клетку, `Liquid` уменьшает урон юнитов, находящихся на этой клетке, `Ground` является

обычной поверхностью без эффектов. Взаимодействие с юнитами реализовано в функции moveUnit класса Field.

- Ландшафт карты хранится в классе поля в переменной cells.

3)

- Нейтральные объекты могут располагаться на поле с помощью полей с координатами x и y, которые инициализируются в конструкторе. Юниты могут взаимодействовать и нейтральными объектами.
- Были созданы 4 класса Нейтральных объектов.
- Взаимодействие юнитов с нейтральными объектами, реализовано в виде перегрузки оператора += в Классе Unit, который вызывается из функции takeItem класса Base. Оператор вызывает функцию takeItem класса Unit, которая перегружена для работы с разными типами нейтральных объектов.
- Классы имеют общий интерфейс Neutral.

4)

- Было добавлено взаимодействие юнитов между собой
- Когда функция moveUnit пытается переместить юнита на клетку, в которой стоит другой юнит, то вызывается функция attack, которая находит нужных юнитов в объектах баз и вызывает у юнитов функцию takeDamage, которая уменьшает здоровье и говорит выжил ли юнит. Если юнит умер, то он удаляется с поля с помощью функции removeUnit.

5)

- Добавлено 3 демонстрационных примера в классе Test.
- Юниты обозначаются большими буквами, нейтральные объекты маленькими, буква О обозначает пустую клетку.
- Цвета букв обозначают ландшафт.

6)

- Взаимодействие юнитов с нейтральными объектами, реализовано в виде перегрузки оператора += в Классе Unit.

Выводы.

В ходе лабораторной работы была изучена работа с интерфейсами классов, способы взаимодействия классов и перегрузка операций в C++.