МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ

по лабораторной работе №1
по дисциплине «Объектно-ориентированное программирование»
Тема: Создание классов, конструкторов классов, методов классов; Наследование

Студентка гр. 8381 _____ Звегинцева Е.Н.
Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург
2020

Цель работы.

Ознакомиться с основными принципами объектно-ориентированного программирования. Реализовать модель игрового поля с объектами на нем с помощью классов, их методов, а также наследования.

Задание.

Разработать и реализовать набор классов:

- Класс игрового поля
- Набор классов юнитов

Игровое поле является контейнером для объектов, представляющим собой прямоугольную сетку. Основные требования к классу игрового поля:

- Создание поля произвольного размера
- Контроль максимального количества объектов на поле
- Возможность добавления и удаления объектов на поле
- Возможность копирования поля (включая объекты на нем)
- Для хранения запрещается использовать контейнеры из stl

Юнит является объектов, размещаемым на поля боя. Один юнит представляет собой отряд. Основные требования к классам юнитов:

- Все юниты должны иметь как минимум один общий интерфейс
- Реализованы 3 типа юнитов (например, пехота, лучники, конница)
- Реализованы 2 вида юнитов для каждого типа (например, для пехоты могут быть созданы мечники и копейщики)
- Юниты имеют характеристики, отражающие их основные атрибуты, такие как здоровье, броня, атака.
- Юнит имеет возможность перемещаться по карте

Выполнение работы.

Написание работы производилось на базе операционной системы Windows 10 в среде разработки QtCreator.

Для игрового поля был создан класс Field. Основные методы класса:

- Field(unsigned width, unsigned height, unsigned itemLimit) – устанавливает высоту, ширину и число объектов на поле

- Field(Field &field) - копирование поля
- Field(Field &&field) - перемещение поля
- bool addItem(unsigned x, unsigned y, FieldItem *item) – добавление объекта по его координатам на поле
- bool deleteItem(FieldItem *item) - поиск и удаление объекта на поле
- bool moveItem(FieldItem *item, int x, int y) – нахождение и перемещение объектов на поле
- bool deleteItem(unsigned x, unsigned y) - удаление объекта по координатам
- FieldItem *getItem(unsigned x, unsigned y) – вызов объекта по координатам

Для класса Field был создан итератор в виде отдельного класса FieldIterator, который позволяет последовательно получать доступ к размещенным на поле объектам.

Для всех юнитов был создан класс Unit, наследованный от интерфейса FieldItem. Для атрибутов юнитов создан класс Attributes, три объекта которого (Health, Armor, Attack) хранятся в классе Unit. В классе реализованы методы для установки и получения характеристик, получения информации в виде строки, имени, а также для перемещения.

Перемещение юнита осуществляется с помощью паттерна «посредник», реализованного классом MoveMediator. Класс передает классу Field запрос от юнита на передвижение, а в случае ошибки передает исключение, брошенное методом moveItem.

Различные типы и виды юнитов реализованы с применением паттерна абстрактной фабрики. В нашем случае fantasyArmy и humanityArmy

Все методы классов сохраняют инварианты этих классов. Так, для координат используется беззнаковый тип unsigned, а при некорректных входных параметрах (например, координата вне поля) методы бросают исключения.

Выводы.

В ходе выполнения лабораторной работы была написана программа, в которой реализованы классы поля и различных юнитов, взаимодействие между ними. Был использован объектно-ориентированный стиль программирования,

были изучены и применены его основные положения, такие как наследование, инкапсуляция, инвариантность.