

Национальный исследовательский институт «МЭИ»

Лабораторная работа по технологиям программирования

Объектно-ориентированное программирование
в C++ и C#

Работу выполнил:

студент группы Аэ-21-24

Лыкум Фёдор

Оценка _____

Подпись _____

Дата _____

Москва, 2025

Оглавление

Введение.....	2
1 Техническое задание.....	2
2 Разработка и реализация программы на языке C++.....	3
2.1 Диаграммы классов.....	4
2.2 Разработка пользовательского интерфейса.....	6
2.3 Описание программы.....	6
2.4 Тестирование программы.....	6
3 Разработка и реализация программы на языке C#.....	8
3.1 Диаграммы классов и интерфейсов.....	8
3.2 Разработка пользовательского интерфейса.....	9
3.3 Описание программы.....	9
3.4 Тестирование программы.....	9
4 Сравнительный анализ реализаций.....	9
5 Заключение.....	10
6 Список используемых источников.....	10
7 Приложения. Исходный код программ.....	10

Введение

В современном мире всё больше и больше людей[1] хочет поиграть в D&D (Настольную игру, в русскоязычном сообществе известную как «Подземелья и драконы»), однако многие из них сталкиваются с невозможностью найти партнёров, с которыми можно поиграть. Уже существуют игры, позволяющие играть в D&D с компьютером, например, Baldur's Gate 3. Данная игра, однако, крайне дорогая, а также имеет высокие системные требования.

Необходимо простое и компактное приложение, заключающее в себе основную суть настольной игры D&D, чтобы все желающие могли попробовать и узнать, что вообще такое D&D в первом приближении.

1 Техническое задание

Необходимо разработать программу на C++/C# с использованием функций ООП[2] для операционных систем Windows и/или Linux. Программа должна симулировать фэнтезийное подземелье с комнатами, в которых должны быть предметы и существа. Все основные механики должны быть заимствованы из настольной игры D&D.

2 Разработка и реализация программы на языке C++

2.1 Диаграммы классов

Ниже приведены диаграммы классов. Важно заметить: классы Dungeon, Room, Creature и Item также связаны друг с другом через агрегацию, поэтому *абсолютно* все классы, так или иначе, имеют связь, даже если на диаграмме это не отображается.

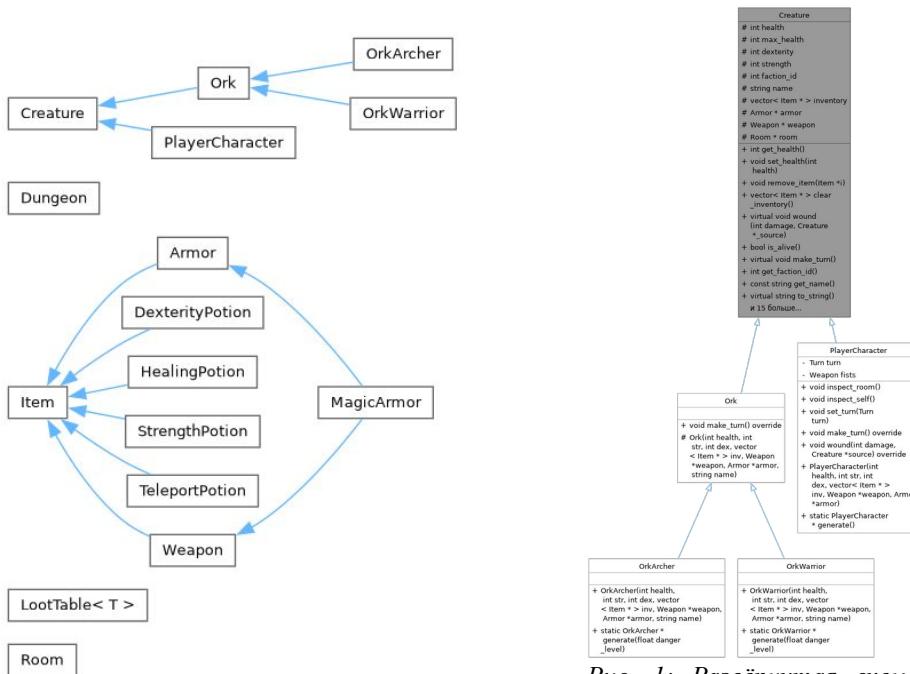


Рис. 2: Общая схема наследования

Рис. 1: Развёрнутая схема наследования класса Creature

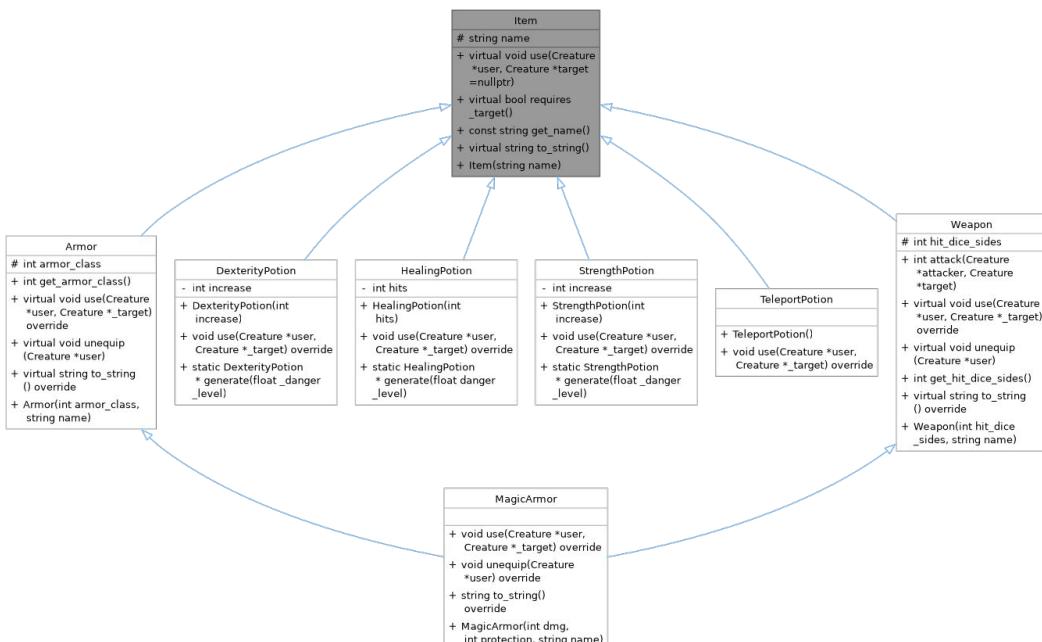


Рис. 3: Развёрнутая схема наследования класса Item

2.2 Разработка пользовательского интерфейса

Было принято решение разработать консольный пользовательский интерфейс, т.к. он простой и эффективный, и т. к. программа, в первую очередь, рассчитана на пользователей, хорошо управляющихся с компьютером.

Получился пользовательский интерфейс, состоящий из следующих команд:

1. me - осмотреть самого себя
2. room - осмотреть текущую комнату
3. use [предмет] - использовать предмет из инвентаря
4. take [предмет] - взять предмет из инвентаря комнаты
5. attack [цель] - атаковать существа основным оружием
6. go [r/t/l/b] - переместиться в соседнюю комнату
7. help - получить помощь по командам
8. quit - выйти из приложения

2.3 Описание программы

В созданной программе можно управлять персонажем, его инвентарём и снаряжением, можно атаковать врагов, можно использовать зелья, можно забирать предметы из комнат, а также из инвентарей врагов, находящихся без сознания. У персонажа имеется шкала здоровья, а также показатель ловкости и силы.

В целом, программа соответствует поставленному техническому заданию.

2.4 Тестирование программы

Объект испытаний — программа «a.out».

Цель испытаний — проверить корректность пользовательского интерфейса программы.

Средства испытаний — Компьютер, клавиатура, монитор, операционная система EndeavourOS Linux, терминальный эмулятор Konsole.

Схема испытаний — по схеме «чёрный ящик».

```
[fedor@endeavour:~/D/u/s/p/a/cpp]-[20:08:27]-[G:main =]
$ ./a.out 0
  
> help  
Список доступных команд:  
me - осмотреть самого себя  
room - осмотреть текущую комнату  
use [предмет] - использовать предмет из инвентаря  
take [предмет] - взять предмет из инвентаря комнаты  
attack [цель] - атаковать существо основным оружием  
go [r/t/l/b] - переместиться в соседнюю комнату  
help - получить помощь по командам  
quit - выйти из приложения
```

```
Здоровье: 16/16  
Ловкость: 2  
Сила: 3  
Оружие: Кинжал (1к4)  
Доспехи: Проклённая кожа (12)  
Инвентарь:  
1. Кинжал (1к4) (экипировано)  
2. Проклённая кожа (12) (экипировано)  
> room  
Предметы:  
1. Зелье лечения  
2. Зелье лечения  
Существа:  
Ты тут один... Может оно и к лучшему  
Отсюда можно пойти: налево, вверх
```

```
> me  
Здоровье: 16/16  
Ловкость: 2  
Сила: 3  
Оружие: Кинжал (1к4)  
Доспехи: Проклённая кожа (12)  
Инвентарь:  
1. Кинжал (1к4) (экипировано)  
2. Проклённая кожа (12) (экипировано)
```

```
> take 1  
Взят предмет: Зелье лечения  
> me  
Здоровье: 16/16  
Ловкость: 2  
Сила: 3  
Оружие: Кинжал (1к4)  
Доспехи: Проклённая кожа (12)  
Инвентарь:  
1. Кинжал (1к4) (экипировано)  
2. Проклённая кожа (12) (экипировано)  
3. Зелье лечения
```

```
> asdklfj aslkjdf  
Неизвестная команда: asdklfj aslkjdf. Используйте команду "help", чтобы получить помощь  
> asdf a  
Неизвестная команда: asdf a. Используйте команду "help", чтобы получить помощь
```

```
> take 0  
Неверный аргумент команды. Требуется натуральное число  
> take 3  
В комнате нет предмета с таким номером  
> take abc  
Неверный аргумент команды. Требуется натуральное число  
> take  
Укажите номер предмета в комнате
```

3 Разработка и реализация программы на языке C#

3.1 Диаграммы классов и интерфейсов

Диаграмма наследования приведена на Рис. 4.

Такие классы, как Armor, MagicArmor, StrengthPotion, Weapon, DexterityPotion и HealingPotion реализуют интерфейс IItem. Классы Armor и MagicArmor реализуют интерфейс IAmmor. Классы Weapon и MagicArmor реализуют интерфейс IWeapon.

Классы Form1 и Program относятся к графической части и не содержат никакой «бизнес-логики».

Классы Config, Generators и Rand содержат вспомогательные функции или константы. Это статические классы, экземпляры которых никогда не создаются.

Класс Turn и его наследники задают возможные действия игрока во время хода.

Класс LootTable является вспомогательным, используется только для того, чтобы заполнить комнаты подземелья врагами и предметами, т.е. только на стадии генерации подземелья.

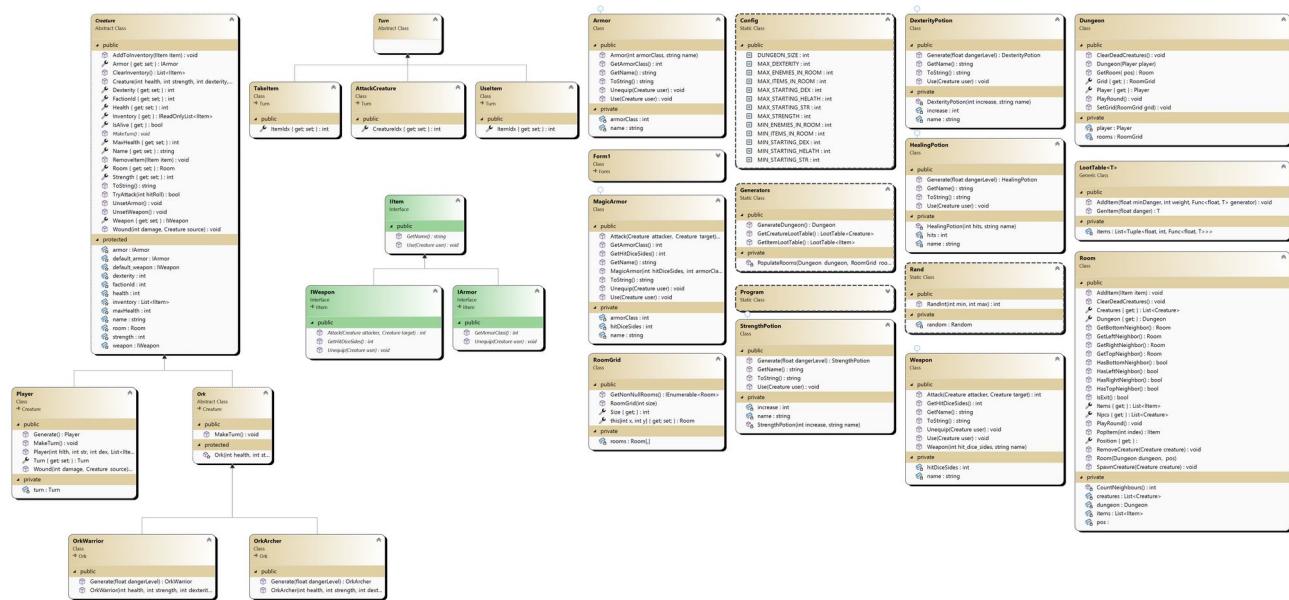


Рис. 4: Развёрнутая диаграмма наследования

3.2 Разработка пользовательского интерфейса

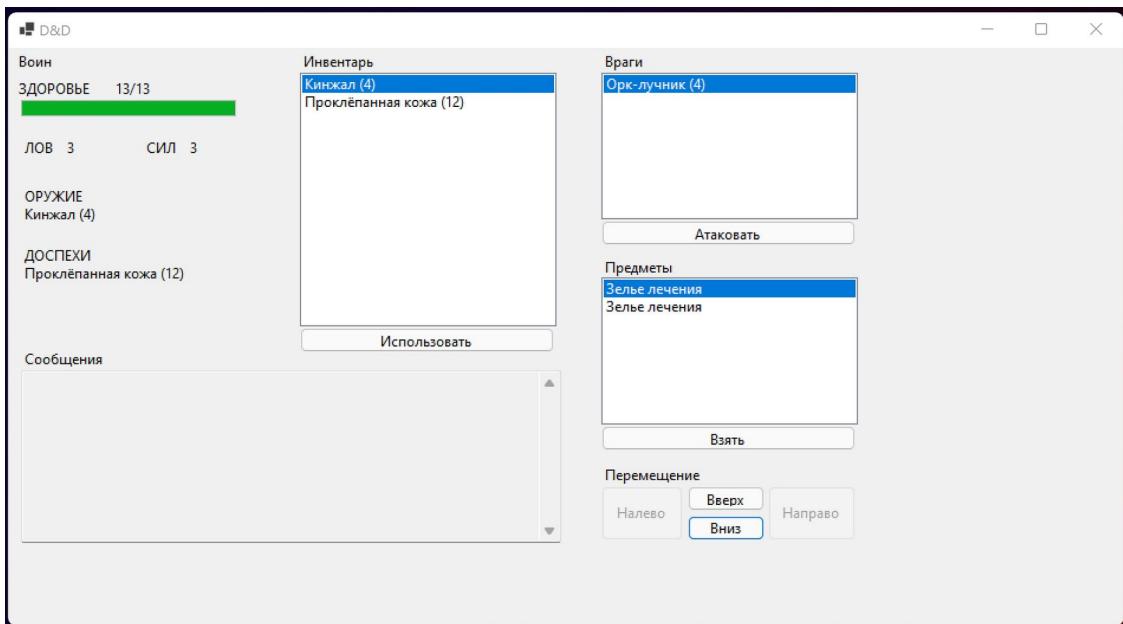


Рис. 5: Пользовательский интерфейс программы

3.3 Описание программы

В программе точно такой же функционал и логика, только вместо консольного интерфейса — графический.

3.4 Тестирование программы

Были проведены все те же самые тесты, что и в 2.4. Результаты получились те же самые.

4 Сравнительный анализ реализаций

Обе программы реализовывают одни и те же игровые механики, однако версия на C++ работает через консольный интерфейс, а версия на C# - через графический интерфейс.

Что касается кодовой части, реализация на C#, однозначно, получилась более простой. Отчасти из-за того, что в C# не нужно беспокоиться об управлении памятью, а также из-за того, что ООП было изначально заложено в дизайн языка C#.

5 Заключение

В ходе лабораторной работы была реализована игра, берущая за основу механики и идеи из настольной игры D&D. Игра имеет две реализации: консольную на C++ и графическую на C#.

К сожалению, в данной игре очень мало механик и контента, она быстро надоедает. В дальнейшем можно было бы, например, добавить новых врагов, предметы, увеличить подземелье и сделать его более разнообразным и т.п.

6 Список используемых источников

[1] - <https://webtribunal.net/blog/how-many-people-play-dnd/>

[2] - *Khurana, Rohit (1 November 2009). Object Oriented Programming with C++, 1E. Vikas Publishing House Pvt Limited. ISBN* .

7 Приложения. Исходный код программ

Т.к. исходного кода очень много (В версии на C++ более тысячи строчек кода, в версии для C# поменьше, но исходники разбросаны по множеству файлов), было решено предоставить доступ к репозиторию на Github, где находится весь исходный код.

В папке spp находится исходный код для программы на C++, а также дополнительные материалы и файлы. Сам код расположен в файлах spp/main.cpp и spp/config.cpp.

В папке cs находится проект Visual Studio с программой на C#. Сам код расположен в папке cs/dnd в файлах с расширением .cs.

Репозиторий находится по ссылке:

<https://github.com/Vinegret43/another-thing-for-uni>