ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(национальный исследовательский университет)» (МАИ)

Направление подготовки: 27.03.05 «Инноватика»

**Лабораторная работа №1**

по дисциплине «Информатика»

Выполнили:

Студенты гр. М3О-137БВ-25

Лопандин Т. К.

Пантелеев З. Д.

Преподаватель:

Александрова С.С.

Москва 2025

Оглавление

[1. Цель работы 3](#_Toc211295712)

[2. Разработка архитектуры игры 3](#_Toc211295713)

[2.1. Структура проекта 3](#_Toc211295714)

[2.2. Основные компоненты системы 3](#_Toc211295715)

[3. Реализация классов и интерфейсов 3](#_Toc211295716)

[3.1. Базовый класс Character 3](#_Toc211295717)

[3.2. Система интерфейсов 4](#_Toc211295718)

[3.3. Класс Boss 5](#_Toc211295719)

[3.5. Система эффектов 7](#_Toc211295720)

[4.1. Механика пошагового боя 7](#_Toc211295721)

[4.2. Управление командой 8](#_Toc211295722)

[4.3. Система стратегий босса 9](#_Toc211295723)

[5. Реализация пользовательского интерфейса 10](#_Toc211295724)

[5.1. InputManager для управления вводом 10](#_Toc211295725)

[5.2. Визуальное представление 11](#_Toc211295726)

[6.1. Основной сценарий игры 13](#_Toc211295727)

[6.2. Результаты тестирования 13](#_Toc211295728)

[7. Вывод 14](#_Toc211295729)

# 1. Цель работы

Разработать консольную игру на языке Python с использованием объектно-ориентированного программирования, реализовать систему классов, интерфейсов и механику пошагового боя с боссом.

# 2. Разработка архитектуры игры

## 2.1. Структура проекта

Проект организован в виде модульной структуры с четким разделением ответственности между компонентами:

LR2/  
├── Scripts/  
│ ├── Entity/  
│ │ ├── basic\_entity.py  
│ │ └── hero\_classes.py  
│ ├── Battle/  
│ │ └── battle.py  
│ ├── command/  
│ │ └── command.py  
│ ├── Types/  
│ │ └── types.py  
│ └── Effects/  
│ └── effects.py  
├── Interfaces/  
│ └── Interfaces.py  
└── main.py

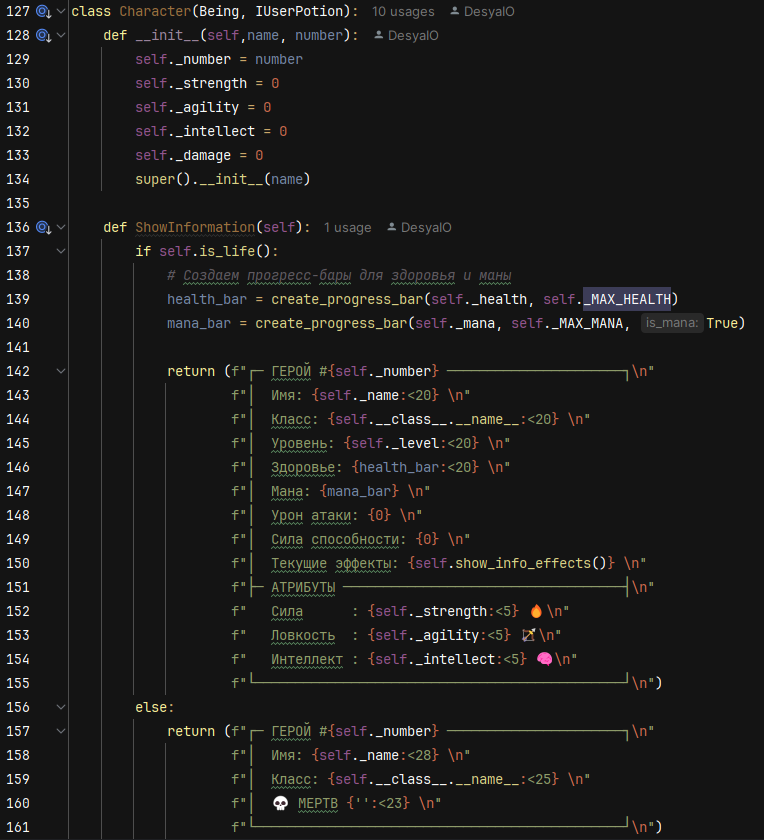
## 2.2. Основные компоненты системы

* Entity - содержит базовые классы персонажей и их специализации
* Interfaces - определяет интерфейсы для различных типов действий
* Battle - реализует механику пошагового боя
* Command - управляет командой героев
* Effects - система эффектов и их применения
* Types - перечисления для типизации

# 3. Реализация классов и интерфейсов

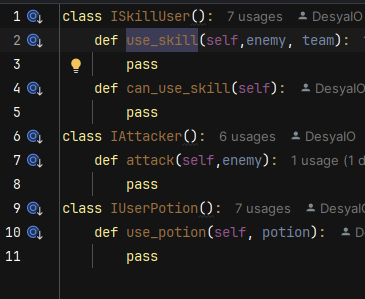
## 3.1. Базовый класс Character

Базовый класс определяет общие свойства и методы для всех персонажей:

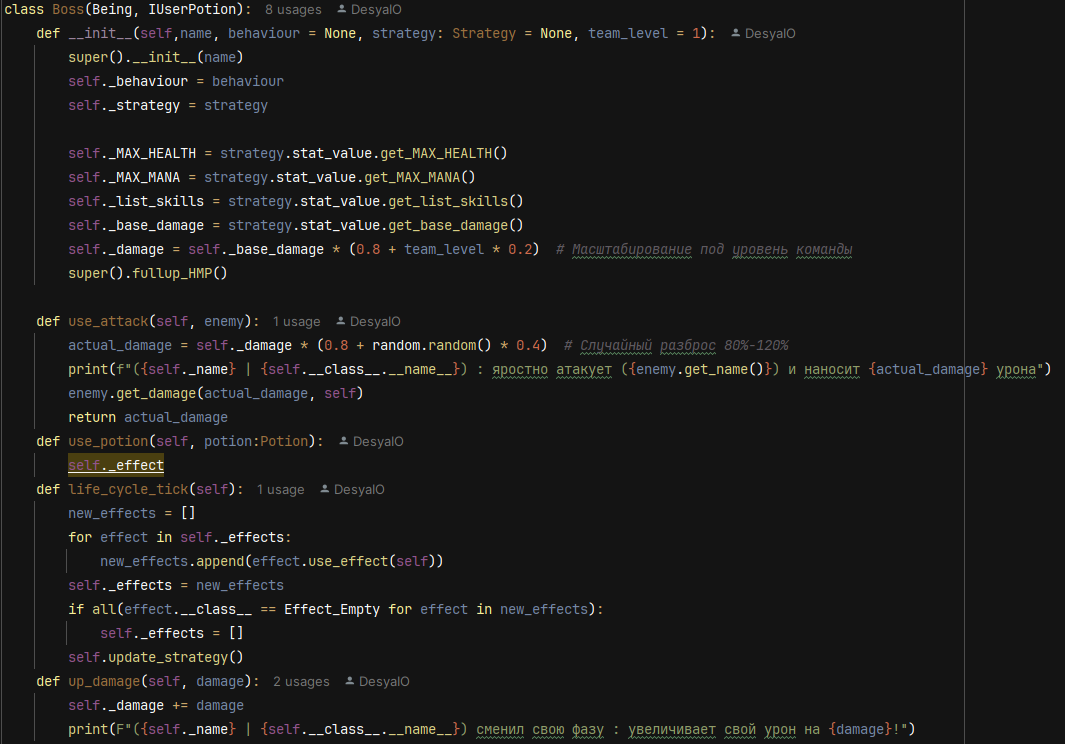
Основные методы включают получение урона, лечение, управление маной, проверку состояния персонажа и систему эффектов.

## 3.2. Система интерфейсов

Для обеспечения гибкости системы реализованы интерфейсы:



## 3.3. Класс Boss

**3.4. Специализированные классы героев**

**Воин (Warrior):**

* Высокое здоровье (150)
* Низкий запас маны (60)
* Сила: 8, Ловкость: 2, Интеллект: 1
* Базовая атака: 15 + сила × 2
* Умение "Щит": 25 + сила × 3 урона, лечение 15 + сила × 0.5

**Маг (Mage):**

* Среднее здоровье (120)
* Высокий запас маны (250)
* Сила: 1, Ловкость: 2, Интеллект: 10
* Магическая атака: 10 + интеллект × 1.5
* Огненный шар: 15 + интеллект × 2 урона, лечение 20

**Лекарь (Healer):**

* Среднее здоровье (130)
* Высокий запас маны (200)
* Сила: 6, Ловкость: 1, Интеллект: 7
* Исцеление команды: (сила + ловкость + интеллект) × 2

## 3.5. Система эффектов



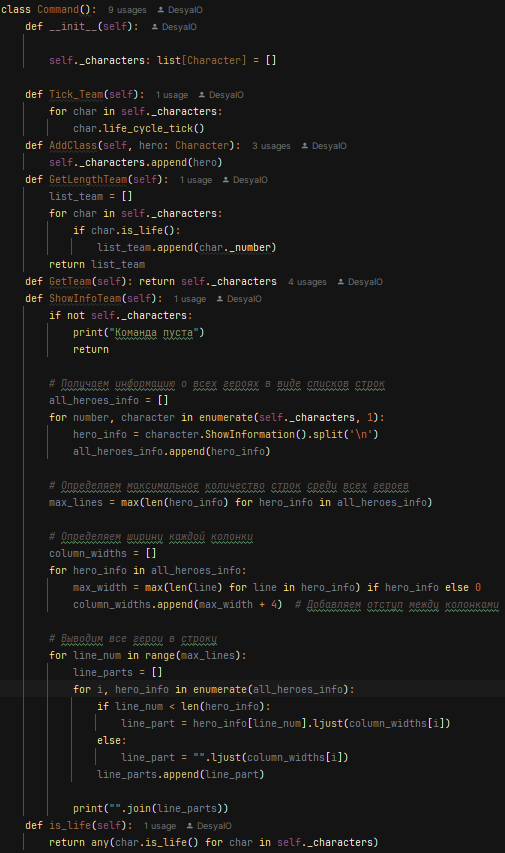
## 4.1. Механика пошагового боя

Система боя реализована в классе Battle и работает по следующему алгоритму:

1. Инициализация команды героев и босса
2. Цикл битвы пока босс или команда живы
3. Поочередный ход каждого живого героя
4. Выбор действия через InputManager
5. Ход босса
6. Проверка условий победы/поражения

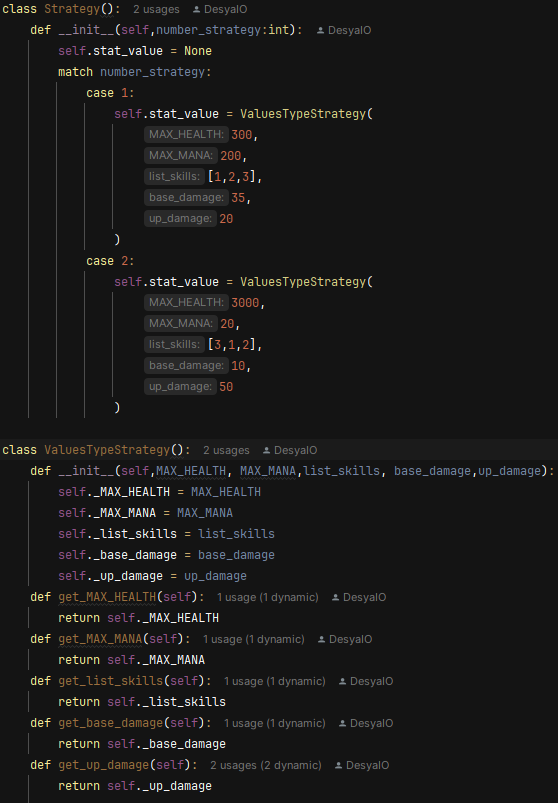
## 4.2. Управление командой

Класс Command обеспечивает управление группой героев:



Отображение информации о команде в табличном формате

## 4.3. Система стратегий босса



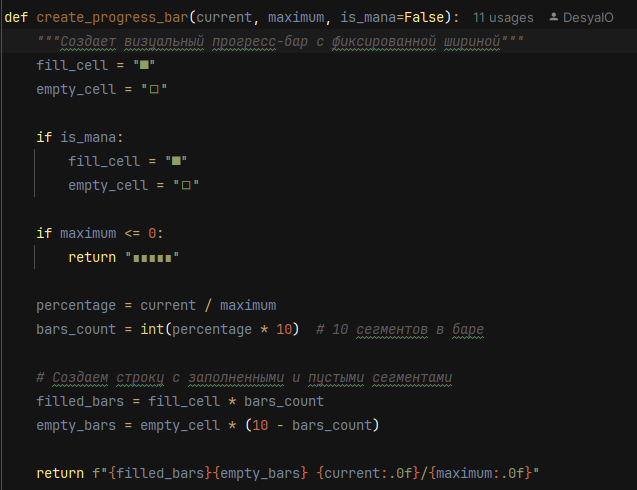
# 5. Реализация пользовательского интерфейса

## 5.1. InputManager для управления вводом



## 5.2. Визуальное представление

Реализована система прогресс-баров для отображения здоровья и маны:

**6. Тестирование приложения**

## 6.1. Основной сценарий игры



## 6.2. Результаты тестирования

В процессе тестирования проверены следующие аспекты:

1. Корректное создание персонажей с заданными характеристиками
2. Работоспособность системы пошагового боя
3. Корректное расходование маны при использовании умений
4. Правильная работа системы эффектов
5. Корректное определение победителя в битве
6. Баланс игрового процесса между классами
7. Работа системы ввода и подтверждения действий

Все основные функции работают корректно, игровой процесс сбалансирован.

# 7. Вывод

В ходе лабораторной работы была успешно разработана консольная RPG-игра на языке Python с применением принципов объектно-ориентированного программирования.

Основные достижения:

1. Спроектирована модульная архитектура приложения с четким разделением ответственности между компонентами
2. Реализована сложная система наследования с базовым классом Character и специализированными классами героев
3. Применены интерфейсы для разделения различных типов поведения персонажей
4. Обеспечена инкапсуляция через использование защищенных атрибутов и методов
5. Реализована расширяемая система эффектов с поддержкой длительности
6. Создана гибкая система стратегий для босса
7. Разработан удобный пользовательский интерфейс с прогресс-барами и табличным выводом

Разработанная игра демонстрирует практическое применение основных принципов ООП и может быть расширена добавлением новых классов персонажей, умений, эффектов и игровых механик. Баланс характеристик обеспечивает интересный и разнообразный игровой процесс, где каждый класс выполняет уникальную роль в команде.

Система обладает высокой степенью сопровождаемости и может быть легко модифицирована для добавления новой функциональности.