

BLG 252E Object Oriented Programming Homework 3

Due: Friday, May 12, 2023

Lecturer: Feza Buzluca CRN: 23136

Fatih Baskın

150210710

Contents

Introduction	3
Relations, Visibility, Parameter Hiding	3
UML Diagram	4

Introduction

In this homework, we were asked to implement a Pokemon game. I have created four `.h` files and four `.cpp` files. `pokemon.h` and `pokemon.cpp` contains the `Pokemon` class definitions and declarations respectively. `electricPokemon.h` and `electricPokemon.cpp` contains the `ElectricPokemon`, `psychicPokemon.h` and `psychicPokemon.cpp` contains the `PsychicPokemon`, `arena.h` and `arena.cpp` contains the `Arena` class definitions and declarations respectively. The code is explained in detail with comments in the source codes.

A makefile is also included, and homework can be compiled with `make assignment3` console command in a Linux environment.

Relations, Visibility, Parameter Hiding

`Pokemon` and `ElectricPokemon` classes have an inheritance relation, `ElectricPokemon` is a `Pokemon`. Inheritance is a public inheritance. Name hiding is used for the `dealDamage()` method.

`Pokemon` and `PsychicPokemon` classes have an inheritance relation, `PsychicPokemon` is a `Pokemon`. Inheritance is a public inheritance. Name hiding is used for the `dealDamage()` method.

`ElectricPokemon` and `Arena` classes have an aggregation relation, `Arena` has a `ElectricPokemon`. `Arena` has both parameter visibility (with the `void setElectricPokemon(ElectricPokemon *)` method) and attribute visibility to `ElectricPokemon`.

`PsychicPokemon` and `Arena` classes have an aggregation relation, `Arena` has a `PsychicPokemon`. `Arena` has both parameter visibility (with the `void setPsychicPokemon(PsychicPokemon *)` method) and attribute visibility to `PsychicPokemon`.

UML Diagram

