Middle East Technical University | Department of Computer Engineering

CENG 242

Programming Language Concepts

Spring '2016-2017

Programming Assignment 4

Due date: 21 May 2017, Monday, 23:55

1 Objectives

In this assignment you will practice the Abstraction, Inheritance and Polymorphism concepts of Object Oriented Programming.

Keywords: *OOP, inheritance, polymorphism, (pure) abstract class*

2 Problem Definition

Welcome to the World's Greatest PokéTournament! Prepare your freshly caught Pokémons, and fasten your seatbelts ladies and gentlemen, soon the Pokémon Champion will be determined and we will witness this glorious event.

Oh, wait... We're not in Japan... Change of plans! We will create our own PokéTournament with Black-Jack and *...Wait, that's not what I had in my mind...* Nevermind, I have an idea! We will **SIMULATE** our own PokéTournament. But we need your help.

3 Specifications

3.1 TrainerCenter

This class is provided for storing the Trainers, Pokémons and modularizing the input parsing. For Tournament(s), you will parse every Trainer and Pokemon from given file.

3.2 Trainer

Brave young guys and gals, determined to win the PokéTournament. They're identified with **trainerID** and **name**.

Its attributes are **favoriteArena** and **pokemons**. Every trainer will have the same number of Pokémon.

3.3 Pokémon

Our lovely and loyal pet(s). It's identified by **pokemonID** and **name**.

Its attributes are **HP**, **ATK**, **MAG_DEF** and **PHY_DEF**.

Every Pokémon has a special skill that will be applied to its target. It differs with the Pokémon's type.

There will be 5 different types of Pokémons:

`(PokéFire, PokéWater, PokéElectric, PokéGrass and PokéFlying).`

3.3.1 PokéFire

`HP = 600, ATK = 60, MAG_DEF = 20, PHY_DEF = 10.`

Will activate `BURNING`, which will decrease opponent's health by `BURN_DAMAGE` at each turn.

It's immune to `BURNING`, but will take twice damage from `DROWNING`.

3.3.2 PokéWater

`HP = 700, ATK = 50, MAG_DEF = 10, PHY_DEF = 20.`

Will activate `DROWNING`, which will decrease opponent's health by `DROWN_DAMAGE` at each turn.

It's immune to `DROWNING`, but will take twice damage from `ELECTRIFIED`.

3.3.3 PokéElectric

`HP = 500, ATK = 70, MAG_DEF = 30, PHY_DEF = 0.`

Will activate `ELECTRIFIED`, which will decrease opponent's health by `ELECTRIFY_DAMAGE` at each turn.

It's immune to `ELECTRIFIED`, but will take twice damage from `ROOTED`.

3.3.4 PokéGrass

`HP = 800, ATK = 40, MAG_DEF = 0, PHY_DEF = 30.`

Will activate `ROOTED`, which will decrease opponent's health by `ROOT_DAMAGE` at each turn.

It's immune to `ROOTED`, but will take twice damage from `BURNING`.

3.3.5 PokéFlying

`HP = 650, ATK = 55, MAG_DEF = 0, PHY_DEF = 15.`

Will **NOT** activate any effect, but immune to **ALL** effects.

3.4 Tournament

The Tournament class is responsible for holding duels. Basic rules of tournaments apply, two trainer will duel, winners will continue to next phase, until only 1 trainer -*the ultimate winner*- is left.

The tournament will be commenced as such:

1. Initialize the list with trainers.
2. While there are more than 1 trainers in the list:
   (a) Create Trainer pairs for duel(T1-T2). (*)
   (b) While there are Trainer pairs that didn't duel:
   - There will be 3 different match:
     - First Match: T1 vs T2, at T1's favorite arena.
     - Second Match: T2 vs T1, at T2's favorite arena.
     - Third Match: T1 vs T2, at Stadium (Neutral) arena.
   - Determine the winner, remove loser from the list.
3. Determine the Champion.

- (*) While there are non-paired Trainers in the list:
  1. Pick the first and the last Trainer from the list.
  2. Pair them, and remove from the list.

Duel between two Trainers (First Trainer: T1, Second Trainer: T2):

- While both trainers have more than 0 Pokémon:
  1. If there is no Pokémon, summon first Pokémons. T1 attacks first at the beginning.
  2. Duel T1's current Pokémon with T2's current Pokémon.
  3. If T1's Pokémon wins, T2 will replace his Pokémon and T2 will attack first.
  4. If T2's Pokémon wins, T1 will replace his Pokémon and T1 will attack first.

  Note: When a Pokémon is summoned, current arena will increase/decrease its powers.

Duel between two Pokémons (First Pokémon: P1, Second Pokémon: P2):

- While both Pokémons live:
  1. If P1 is alive; P1 is damaged from special effects (if any). Else, P2 has won.
  2. If P1 is alive; P1 attacks to P2 and applies its special effect. Else, P2 has won.
  3. If P2 is alive; P2 is damaged from special effects (if any). Else, P1 has won.
  4. If P2 is alive; P2 attacks to P1 and applies its special effect. Else, P1 has won.
- Declare the winner, and the winner will be leveled up.
- There will **NOT** be a case of tie in Pokémon duel.

Details of the attacks/effects:

- P1 will take damage from effects (if it has them):
  1. `MAX(0, BURNING - P1.MAG_DEF)`
  2. `MAX(0, DROWNING - P1.MAG_DEF)`
  3. `MAX(0, ELECTRIFIED - P1.MAG_DEF)`
  4. `MAX(0, ROOTED - P1.MAG_DEF)`
- P2 will take damage:
  `MAX(1, P1.ATK - P2.PHY_DEF)`
- P2 will take damage from effects (if it has them):
  1. `MAX(0, BURNING - P2.MAG_DEF)`
  2. `MAX(0, DROWNING - P2.MAG_DEF)`
  3. `MAX(0, ELECTRIFIED - P2.MAG_DEF)`
  4. `MAX(0, ROOTED - P2.MAG_DEF)`
- P1 will take damage:
  `MAX(1, P2.ATK - P1.PHY_DEF)`

  Note: Pokémon may be immune or take twice damage from various effects.

Details of the leveling up:

- **PokéFire**: Increase `HP` by **60**, `ATK` by **6**, `MAG_DEF` by **2**, `PHY_DEF` by **1**.
- **PokéWater**: Increase `HP` by **70**, `ATK` by **5**, `MAG_DEF` by **1**, `PHY_DEF` by **2**.
- **PokéElectric**: Increase `HP` by **50**, `ATK` by **7**, `MAG_DEF` by **3**, `PHY_DEF` by **0**.
- **PokéGrass**: Increase `HP` by **80**, `ATK` by **4**, `MAG_DEF` by **0**, `PHY_DEF` by **3**.
- **PokéFlying**: Increase `HP` by **65**, `ATK` by **5**, `MAG_DEF` by **0**, `PHY_DEF` by **3**.

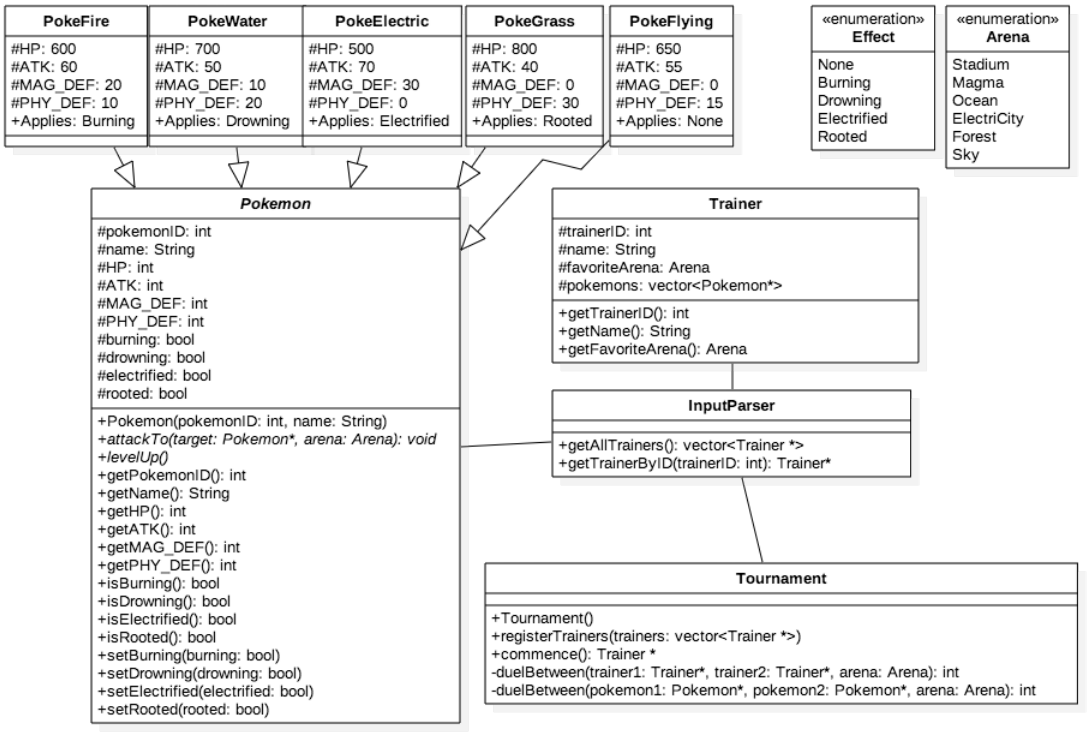
Details of the Arena Powers:

- **Stadium** $(Neutral)$
- **Magma** $(PokeFire++, PokeElectric--, PokeGrass--)$
- **Ocean** $(PokeWater++, PokeFire--, PokeFlying--)$
- **ElectriCity** $(PokeElectric++, PokeWater--, PokeFlying--)$
- **Forest** $(PokeGrass++, PokeWater--, PokeElectric--)$
- **Sky** $(PokeFlying++, PokeFire--, PokeGrass--)$

Buff ($++$): Increase `HP` by **100**, `ATK` by **10**.

Debuff ($--$): Decrease `HP` by **100**, `ATK` by **10**.

4 Class Diagram



5 Input Format

The input will be in this format:

```
Trainer<0>ID--Trainer<0>Name--Trainer<0>FavoriteArena
pokemon<0>ID::pokemon<0>Name::pokemon<0>Type
pokemon<1>ID::pokemon<1>Name::pokemon<1>Type
...
pokemon<N>ID::pokemon<N>Name::pokemon<N>Type
```

5.1 Input Example

2 Trainers, 3 Pokémon Each

```
1  Trainer Count:2
2  Pokemon Count:3
3
4  0--Clemont--Forest
5   0::Weepinbell::GRASS
6   1::Cloyster::WATER
7   2::Slowpoke::WATER
8
9  1--Gary--Sky
10  3::Tangela::GRASS
11  4::Electrode::ELECTRIC
12  5::Kingler::WATER
```

6 Output Format

The output should be in this format:

- **Pokémon Duel : P1 vs P2**
  `P1.name(P1.currentHealth) {is burning/is drowning/is electrified/is rooted}` (!!, !!!!)
  `P1.name(P1.currentHealth) hit P2.name(P2.currentHealth) <dmg>(+, /, -)`
  `P2.name(P2.currentHealth) {is burning/is drowning/is electrified/is rooted}` (!!, !!!!)
  `P2.name(P2.currentHealth) hit P1.name(P1.currentHealth) <dmg>(+, /, -)`
  `...`
  `WinnerP: WinnerPokemon.name`

  !! = Normal Effect Damage

  !!!! = Double Effect Damage

  + = Arena Buffed Attack

  / = Arena Neutral Attack

  - = Arena Debuffed Attack

  Example: (Arena: ElectriCity)

```
1 Pikachu(682) is rooted!!!!
2 Pikachu(641) hit Bulbasaur(690) 50(+)
3 Bulbasaur(640) is electrified!!
4 Bulbasaur(570) hit Pikachu(641) 68(/)
5 . . .
6 WinnerP:Pikachu
```

- **Trainer Duel: T1 vs T2**
  `T1.name vs T2.name:  ArenaName`
  `<Pokémons Duels>`
  `WinnerT: WinnerTrainer.name`

  Example: May vs Max

```
1  May vs Max: ElectriCity
2    Bulbasaur(1280) hit Aerodactyl(875) 34(/)
3    Aerodactyl(841) hit Bulbasaur(1280) 22(-)
4    . . .
5    WinnerP:Bulbasaur
6
7    Pikachu(750) hit Bulbasaur(810) 50(+)
8    Bulbasaur(760) is electrified!!
9    . . .
10   WinnerP:Pikachu
11
12   .
13   .
14   .
15 WinnerT:May
```

- **Whole Tournament**

  ```
  Started Round:  IndexOfT1 vs IndexOfT2
  <Trainer Duels (3 match)>
  Round Winner:  WinnerTrainer.name!
  .
  .
  .
  ```

  Example: MegaTournament

```
Started Round: 0 vs 1
  May vs Max: ElectriCity
    Bulbasaur(1280) hit Aerodactyl(875) 34(/)
    Aerodactyl(841) hit Bulbasaur(1280) 22(-)
    . . .
    WinnerP:Bulbasaur

    Pikachu(750) hit Bulbasaur(810) 50(+)
    Bulbasaur(760) is electrified!!
    . . .
    WinnerP:Pikachu

    .
    .
    .
  WinnerT:May

  Max vs May: Sky
    .
    .
    .
  WinnerT:May

  May vs Max: Stadium
    .
    .
    .
  WinnerT:May

Round Winner: May!
```

Example Index Pairs for 2-Trainer Tournament:

1. **0 vs 1**

Example Index Pairs for 4-Trainer Tournament:

1. **0 vs 3, 1 vs 2** *(2 Trainers will go to next phase)*
2. **0 vs 1**

Example Index Pairs for 8-Trainer Tournament:

1. **0 vs 7, 1 vs 6, 2 vs 5, 3 vs 4** *(4 Trainers will go to next phase)*
2. **0 vs 3, 1 vs 2** *(2 Trainers will go to next phase)*
3. **0 vs 1**

7 Notes

7.1 Input Notes

1. There will be no erroneous input.
2. Trainer count will be $n > 1$.
3. Trainer count for registering the Tournament will always be $2^n, n > 0$
4. Pokémon count for a Trainer will be $n > 0$.

7.2 Output Notes

1. The grades will be determined with black-box testing, hence you need to be careful about spaces, typos, etc.

7.3 Memory Notes

1. When a Trainer has deallocated, his Pokémons should also be deallocated properly.
2. Memory-wise, every Trainer and Pokémon created in TrainerCenter is your responsibility. While testing, TrainerCenter will certainly be deallocated.
3. Please check your codes using valgrind –leak-check=full for memory-leaks.

8 Regulations

1. **Programming Language:** You must code your program in C++. You are expected make sure your code compiles successfully with `g++` on department lab computers using the flags `-ansi -pedantic`. A Makefile will be provided for sample usage.
2. **Late Submission:** You have a total of 10 days for late submission. You can spend this credit for any of the assignments or distribute it for all. For each assignment, you can use at most 3 days-late.
3. **Cheating:** Cheating will result in receiving 0 from all assignments and the university regulations will be applied.
4. **Newsgroup:** You must follow the CENG242 newsgroup for discussions and possible updates on a daily basis.
5. **Evaluation:** Your program will be evaluated automatically using "black-box" technique so make sure to obey the specifications.

9 Submission

Submission will be done via CengClass. Create a zip file named `hw4.zip` that contains all your source code files. Do not submit a file that contains a `main` function. Such a file will be provided and your code will be compiled with it.
Note: The submitted zip file should not contain any directories! The following command sequence is expected to run your program on a Linux system:

```
$ unzip hw4.zip
$ make clean
$ make all
$ make run
```