

Pokemon Trading Card Game

Dans le cadre de cet exercice de travaux pratiques, vous allez mettre en œuvre le jeu de cartes Pokémon en C++. Utilisez des classes, des fonctions et des boucles pour créer une simulation précise des mécanismes du jeu. Assurez-vous de suivre attentivement les consignes pour produire une solution conforme.



<https://tcg.pokemon.com/en-us/>

Question 1

Commencez par implémenter une classe de base `Card` qui aura un élément appelé `cardName` de type chaîne de caractères (string). Cette classe comportera une fonction virtuelle pure appelée `displayInfo` de type void.

Question 2

Implémentez maintenant la classe `PokemonCard` qui hérite de la classe `Card`. Cette classe devrait avoir les éléments suivants:

- Le nom de Pokemon sera affecté à `cardName` dans la classe de Base (`Card`)
- `pokemonType` (chaîne de caractères)
- `familyName` (chaîne de caractères)
- `evolutionLevel` (entier)
- `maxHP` (entier)
- `hp` (entier)
- `attacks` (vector<tuple<int, int, string, int>>) contenant uniquement deux attaques. Chaque attaque est un tuple comprenant :
 - Coût en énergie (entier)
 - Coût en énergie actuel (entier)

- Description de l'attaque (chaîne de caractères)
- Dégâts de l'attaque (entier)

Question 3

Implémentez maintenant la classe EnergyCard qui hérite de la classe Card. Cette classe devrait avoir les éléments suivants:

- Le nom de la carte sera toujours par défaut "Energy" pour cardName
- energyType (chaîne de caractères)

Question 4

Implémentez maintenant la classe TrainerCard qui hérite de la classe Card. Cette classe devrait avoir les éléments suivants:

- Le nom du Trainer sera passé au constructeur pour cardName
- trainerEffect (chaîne de caractères)

Pour l'instant, le seul effet que c'est demandé à implementer est "heal all your action pokemon" qui remet les HP de tous les Pokemon en action au maximum.

Question 5

Implémentez maintenant la classe Player avec les attributs suivants :

- playerName (string)
- benchCards (vector<Card*>) un vecteur de pointeurs vers des éléments Card; c'est le deck des cartes en réserve.
- actionCards (vector<PokemonCard*>) un vecteur de pointeurs vers des éléments PokemonCard; c'est le deck des cartes d'action Pokémons.

T.S.V.P.

Question 6

Maintenant, l'objectif est d'adapter toutes les classes précédentes (ajouter des attributs, des fonctions, etc.) de manière à ce que ce code produise la sortie requise.

```
#include "headers/card.h"
#include "headers/pokemon_card.h"
#include "headers/energy_card.h"
#include "headers/trainer_card.h"
#include "headers/player.h"
#include <iostream>
#include <string>

using namespace std;

int main()
{
    Player player1("Hadi");

    player1.addCardToBench(new EnergyCard("Electric"));
    player1.addCardToBench(new EnergyCard("Electric"));
    player1.addCardToBench(new TrainerCard("Ash", "heal all your action pokemon"));
    player1.addCardToBench(new PokemonCard("Pikachu", "Electric", "Pikachu", 2, 100, 2, "thunder bolt", 20, 3, "thunder storm", 30));

    player1.activatePokemonCard(3);
    player1.attachEnergyCard(0, 0);
    player1.attachEnergyCard(0, 0);

    cout << endl;
    player1.displayBench();
    cout << endl;
    player1.displayAction();

    Player player2("Germain");

    player2.addCardToBench(new EnergyCard("Grass"));
    player2.addCardToBench(new TrainerCard("Brock", "heal all your action pokemon"));
    player2.addCardToBench(new PokemonCard("Bulbasaur", "Grass", "Bulbasaur", 1, 100, 2, "Leech Seed", 15, 3, "Vine Whip", 25));

    player2.activatePokemonCard(2);
    player2.attachEnergyCard(0, 0);

    cout << endl;
    player2.displayBench();
    cout << endl;
    player2.displayAction();

    player1.attack(0, 0, player2, 0);
    cout << endl;
    player2.displayAction();
    player2.useTrainer(0);
    cout << endl;
    player2.displayAction();

    return 0;
}
```

Output:

```
Hadi is activating a Pokemon Card: Pikachu
Hadi is attaching Energy Card of type Electric to the Pokemon Pikachu
Hadi is attaching Energy Card of type Electric to the Pokemon Pikachu
```

Bench cards for Player Hadi:

Trainer Card - Name: Ash, Effect: heal all your action pokemon

Action cards for Player Hadi:

Pokemon Card - Name: Pikachu, Type: Electric, Evolution Level: 2 of the family "Pikachu, HP: 100"

Attacks:

Attack #0:

Attack cost: 2

Attack current energy storage: 2

Attack description: thunder bolt

Attack damage: 20

Attack #1:

Attack cost: 3

Attack current energy storage: 2

Attack description: thunder storm

Attack damage: 30

Germain is activating a Pokemon Card: Bulbasaur

Germain is attaching Energy Card of type Grass to the Pokemon Bulbasaur

Bench cards for Player Germain:

Trainer Card - Name: Brock, Effect: heal all your action pokemon

Action cards for Player Germain:

Pokemon Card - Name: Bulbasaur, Type: Grass, Evolution Level: 1 of the family "Bulbasaur, HP: 100"

Attacks:

Attack #0:

Attack cost: 2

Attack current energy storage: 1

Attack description: Leech Seed

Attack damage: 15

Attack #1:

Attack cost: 3

Attack current energy storage: 1

Attack description: Vine Whip

Attack damage: 25

Only 2 attacks exist.

Hadi attacking Germain's Pokemon Bulbasaur with the Pokemon Pikachu with its attack: thunder bolt

Reducing 20 from Germain's Pokemon's HP

Pokemon Bulbasaur is still alive

Action cards for Player Germain:

Pokemon Card - Name: Bulbasaur, Type: Grass, Evolution Level: 1 of the family "Bulbasaur, HP: 80"

Attacks:

Attack #0:

Attack cost: 2

Attack current energy storage: 1

Attack description: Leech Seed

Attack damage: 15

Attack #1:

Attack cost: 3

Attack current energy storage: 1

Attack description: Vine Whip

Attack damage: 25

Germain is using the Trainer Card to "heal all your action pokemon"

Action cards for Player Germain:

Pokemon Card - Name: Bulbasaur, Type: Grass, Evolution Level: 1 of the family "Bulbasaur, HP: 100"

Attacks:

Attack #0:

Attack cost: 2

```
Attack current energy storage: 1
Attack description: Leech Seed
Attack damage: 15
Attack #1:
Attack cost: 3
Attack current energy storage: 1
Attack description: Vine Whip
Attack damage: 25
```