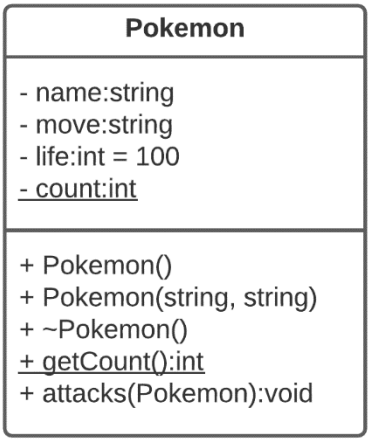
Présentation

On souhaite simuler un combat de Pokémon (pas de 's', c'est invariable 😊)



1. Créer le squelette de la classe Pokémon ci-contre :
    1. name est le nom du Pokémon,
    2. move est le nom de l'attaque du Pokémon,
    3. life est le nombre de points de vie (PV) du Pokémon, il est toujours de 100,
    4. count est le nombre de Pokémon en jeu.
2. Ecrire le constructeur vide. Il doit créer au hasard un des Pokémon sauvages suivants:
    1. Nom : "Roucool sauvage", Attaque : "Tornade",
    2. Nom : "Rattata sauvage", Attaque : "Vive-attaque",
    3. Nom : "Nosferapti sauvage", Attaque : "Morsure".

    Il doit afficher "Un *nomDuPokémonSauvage* apparaît !".
3. Ecrire le constructeur prenant deux paramètres string. Il doit créer un Pokémon avec le nom et l'attaque passés en paramètre. Il ne doit rien afficher.
4. Ecrire le destructeur pour qu'il affiche "Le *nomDuPokémon* est vaincu…".
5. Ecrire la méthode getCount():int qui renvoie le nombre de Pokémon en jeu.
6. Ecrire la méthode attacks(Pokemon):void avec les règles de gestion suivantes :
    1. En début de combat, on affichera "Le *nomDuPokémonAttaqué* attaque !" et "EN AVANT *nomDuPokémonAttaquant* !".
    2. A chaque attaque, on affiche "*nomDuPokémonAttaquant* lance *nomDeSonAttaque*…".
    3. La puissance d'une attaque est aléatoire et inflige de 1 à 40 PV de dommage.
    4. Si une attaque inflige plus de 35 PV de dommage, on affiche : "COUP CRITIQUE !!!"
    5. On affichera le nombre de PV perdus et la vie restante du Pokémon attaqué.
    6. Si le Pokémon attaqué n'a plus de PV, il est supprimé.
7. Ecrire le programme principal qui :
    1. crée un Pokémon sauvage et un Pikachu avec l'attaque Eclair
    2. les font s'attaquer à tour de rôle tant que les deux Pokémon sont en vie, le Pikachu attaquera toujours en premier (parce qu'il est super rapide !).

Résultat attendu :

La classe et le programme seront rendus dans un seul fichier .cpp compatible avec le compilateur g++ v8.1.0 minimum.

La lisibilité et la documentation du code seront appréciées.

Exemple de résultat d'exécution :

```
Un Nosferapti sauvage apparaît !
Le Nosferapti sauvage attaque !
EN AVANT Pikachu !

Pikachu lance Eclair...
COUP CRITIQUE !!!
Nosferapti sauvage perd 38 pv (62/100)

Nosferapti sauvage lance Morsure...
```

```
Pikachu perd 22 pv (78/100)

Pikachu lance Eclair...
Nosferapti sauvage perd 12 pv (50/100)

Nosferapti sauvage lance Morsure...
Pikachu perd 32 pv (46/100)

Pikachu lance Eclair...
Nosferapti sauvage perd 26 pv (24/100)

Nosferapti sauvage lance Morsure...
Pikachu perd 21 pv (25/100)

Pikachu lance Eclair...
Nosferapti sauvage perd 3 pv (21/100)

Nosferapti sauvage lance Morsure...
Pikachu perd 3 pv (22/100)

Pikachu lance Eclair...
COUP CRITIQUE !!!
Nosferapti sauvage perd 37 pv (0/100)

Nosferapti sauvage est vaincu...
```

Astuces

- Pour afficher les caractères accentués, on n'oubliera pas de lancer system("chcp 65001"); en début de progamme (bibliothèque windows.h).
- Pour calculer des nombres aléatoires, on incluera la bibliothèque ctime et on lancera une fois `srand((unsigned) time(0));` en début de programme, puis, après l'avoir intégrée au code, on utilisera la fonction suivante pour obtenir un nombre aléatoire :

```c
/**
 * Simple random generator
 * (do not forget to launch srand((unsigned) time(0)); at the beginning of the program).
 * @param min The lowest value included.
 * @param max The highest value included.
 * @return A random int between the lowest and the highest values provided.
 */
int random(int min, int max) {
    return (rand() % (max + 1)) + min;
}
```