TD de Java n°1 : Attrapez-les tous !

Et voilà ! Après des années d'entraînement, vous êtes enfin devenu un maître Pokemon ! Votre Pokédex est presque rempli, et quasiment plus personne ne peut vous tenir tête. Cependant, vous vous rendez compte qu'avec tous ces Pokemons à gérer, il vous faut un moyen pratique pour organiser votre bestiaire. En effet, avec tous les défis et les aventures que vous vivez en ce moment, vous gagnez et perdez souvent des Pokemons... Comment gérer tout cela de façon efficace ?

Pour commencer, nous allons imaginer que chaque Pokemon est un ensemble de données qui se présente de la façon suivante :

. Une chaîne de caractère **nom** vous donne le nom du Pokemon,

. Une autre chaîne de caractère, **type**, vous donne le type du Pokemon (électrique, eau, etc),

. Un entier **puissance** vous donne la force du Pokemon.

En Java, le type "chaîne de caractère" s'appelle **String**.

Un ensemble de données est appelé une structure de données. En pseudo langage, on déclarera une structure de données ainsi :

```
structure Pokemon
{
      String nom;
      String type;
      int puissance;
}
```

Une fois que l'on a déclaré ceci, on peut construire des variables de type Pokemon (tout comme on pouvait, avant, créer des variables de type int, double, char, etc...). Pour ce faire, on utilise une syntaxe proche de la création de tableau.

```
Pokemon p1;
p1 = new Pokemon();
```

p1 est alors une variable de type Pokemon. Si je souhaite donner un nom à p1, je ferai

```
p1.nom = "Pikachu";
```

Si l'on souhaite créer un tableau **tab** de Pokemon (disons de trois cases), on fera

```
Pokemon tab[]; //on déclare tab comme un tableau de Pokemons
tab = new Pokemon[3];  //tab fera 3 cases.
tab[0] = new Pokemon(); //Sur chaque case du tableau, on créé un nouveau Pokemon
tab[1] = new Pokemon();
tab[2] = new Pokemon();
```

1. Pouvez-vous créer une variable **p2** de type Pokemon, dont le nom sera "Porygon", le type "Epileptique", et la force d'attaque 42 ?

2. Maintenant, nous devons essayer d'organiser nos Pokemons. On peut commencer par faire un tableau de Pokemons : ce tableau permettra de stocker tous les Pokemons que l'on possède. Écrivez le code permettant de construire un tableau de 100 Pokemons.

3. Imaginons que l'on possède un tableau (nommé **collection**) de Pokemons. Ce tableau est sensé lister tous les Pokemons que vous possédez. Écrivez un morceau de code permettant d'afficher le nom de tous les Pokemons de votre tableau (réfléchissez au prototype de la fonction d'abord).

4. Pour effacer un Pokemon du tableau, on changera son nom par "" (la chaîne vide). Écrivez un algorithme qui prend en paramètre un tableau de Pokemon et une position (un entier) et qui efface le Pokemon placé à cette position (réfléchissez au prototype de la fonction d'abord).

5. On souhaite que le tableau ne possède pas de trou, c'est à dire de case vide en plein milieu. Recommencez la question précédente en prenant cette contrainte en considération.

6. De plus, le tableau est trié, et l'on souhaite conserver l'ordre établi entre les Pokemons. Recommencez la question 4 en prenant cette contrainte supplémentaire en considération.

7. Un maître Pokemon perd des Pokemons, mais il lui arrive d'en gagner. Comment feriez-vous pour ajouter un Pokemon à votre collection ?

8. Que pensez-vous des tableaux pour gérer une collection d'éléments qui peut changer de taille (ajout et suppression d'éléments) ? Voyez-vous une solution plus élégante pour gérer ce problème ?

9. Utilisez la solution plus élégante pour répondre aux questions 4 à 7.