TD 6 : RPG / JdR

Exercice 1 : `IntegerItem`

Ajouter deux méthodes dans `IntegerItem` :

1. `getMaxValue` : qui renvoie la valeur maximale du `IntegerItem`,
2. `getMinValue` : qui renvoie la valeur minimal du `IntegerItem`.

Exercice 2 : Race

Définir la classe Race qui possède deux attributs :

1. `physicFormula` de type `IntegerItem`,
2. `mentalFormula` de type `IntegerItem`.

et deux constructeurs :

1. Un constructeur d'initialisation avec comme paramètre les deux attributs,
2. et un constructeur de copie.

Les attributs `physicFormula`, `mentalFormula` doivent posséder un accesseur.

Exercice 3 : Operator+

Écrire l'opérateur de concaténation + pour `String`. Cet opérateur doit permettre de faire :

1. `String + String → String`
2. `String + const char * → String`
3. `const char * + String → String`

Écrire l'opérateur d'affectation (=) si cela n'a pas été encore fait.

Exercice 4 : Operator+

Nous voulons mettre en place l'addition (opérateur(s) +) de deux `IntegerItem` ou d'un entier pour former un nouvel `IntegerItem`.

Le but étant de pouvoir écrire :

```
Dice d6(6);
Race humanRace(d6+d6+d6, d6+d6+d6 ); // 3d6,   3d6
Race orcRace(d6+d6+6, d6+d6);        // 2d6+6, 2d6
```

Réfléchir, tester et proposer une solution.

Il faut vérifier

- que `d6+d6+d6` a une valeur minimale de 3 et une valeur maximale de 18
- et que `d6+d6+6` a une valeur minimale de 8 et une valeur maximale de 18
- Que demander de multiples évaluations de la formule pourront produire des résultats différents.

Exercice 5 : Creature

Modifier la créature pour qu'elle possède une Race et que ses attributs `physic`, `mental` et `hitPoints` (de type `int`) soient initialisés par l'évaluation des attributs `physicFormula`, `mentalFormula` de la Race (de type `IntegerItem`). L'attribut `physic` et l'attribut `hitPoints` ont la même valeur à la création (l'évaluation de `physicFormula`).

```
Dice d6(6);

Race humanRace(d6+d6+d6, d6+d6+d6); // 3d6,   3d6
Race orcRace(d6+d6+6, d6+d6);       // 2d6+6, 2d6

Creature King("Thingol", &humanRace);
Creature orcRace("Boldog",&orcRace);
```

Exercice 5 : `Weapon`

Définir la notion d'arme (`Weapon`) comme étant une classe possédant un attribut `damage` de type `IntegerItem`. `Weapon` appartient au namespace `rpg`.

Exercice 4 : `CraftedWeapon`

Définir la notion d'arme fabriquée/forgée (`CraftedWeapon`). `CraftedWeapon` est à la fois une arme (`CraftedWeapon`) et un objet (`Item`). `CraftedWeapon` appartient au namespace `rpg`.

Écrire deux constructeurs :

3. Un constructeur d'initialisation,
4. et un constructeur de copie.