

L2 TP Noté C++

ASSIETTE DE CRUDITES

L'objectif est d'écrire quelques classes simples permettant de représenter des **crudités** afin de constituer une **assiette de crudités**.

Les lignes de code fournies sont à écrire dans des fonctions externes, telle que la fonction principale ***main***.

PARTIE I : Tomates, poivrons

- 1) Ecrire une **classe Tomate** représentant une tomate avec son poids en kilogrammes et son diamètre (calibre) en centimètres de sorte que :

les lignes de code ci-dessous	produisent l'affichage suivant:
Tomate t = Tomate(0.25, 4) cout << t << endl;	tomate: [0.25 kg – 4 cm]

- 2) Ecrire une **classe Poivron** représentant un poivron avec son poids en kilogrammes et un nombre de graines de sorte que:

les lignes de code ci-dessous	produisent l'affichage suivant:
Poivron p = Poivron(0.3, 10); cout << p << endl;	poivron: [0.3 kg - 10 graines]

- 3) Le poivron ci-dessus n'a pas beaucoup de graines (10 pour un poids de 300 grammes).

En règle générale, les poivrons ont par défaut une graine par tranche de 10 grammes.

Ajoutez à la **classe Poivron** ce qu'il faut pour créer des poivrons qui respectent cette règle générale sans avoir à spécifier explicitement le nombre de graines de sorte que :

les lignes de code ci-dessous	produisent l'affichage suivant :
Poivron p1 = Poivron(0.3, 10); Poivron p2 = Poivron 0.3); Poivron p3 = Poivron (0.75); cout << p1 << endl; cout << p2 << endl; cout << p3 << endl;	poivron: [0.3 kg - 10 graines] poivron: [0.3 kg – 30 graines]] poivron: [0.75 kg – 75 graines]]

PARTIE II : Assiette de crudités

- 1) Comme on risque d'avoir plein de crudités (tomates, poivrons, poireaux...) mais que toutes auront un poids, on souhaite créer une **classe Crudite** pour "**factoriser**" cet attribut, de sorte que le code suivant puisse compiler et que

les lignes de code ci-dessous	produisent l'affichage suivant:
<pre>Crudite * c1 = new Poivron(0.3, 10); Crudite * c2 = new Tomate(0.25, 4); cout << *c1 << endl; out << *c2 << endl;</pre>	<pre>poivron [0.3 kg - 10 graines] tomate: [0.25 kg - 4 cm]</pre>

- 2) On veut maintenant définir une **classe AssietteCrudites** représentant une assiette de crudités par la liste de ses ingrédients (crudités de **type Crudite**).

Vous pouvez utiliser une **classe container de la STL**, ou par défaut un tableau.

Une fois qu'une assiette de crudités est créée, on lui ajoute chaque crudité une par une, grâce à la méthode **add()** oomme illustrée ci dessous. A tout moment, on peut demander l'affichage de l'assiette de crudités.

Écrire la **classe AssietteCrudites** de sorte que le code suivant s'exécute :

```
AssietteCrudites a = AssietteCrudites();  
a.add(new Poivron(0.3, 10));  
a.add(new Tomate(0.25, 4))  
cout << a << endl;  
a.add(new Poivron(0.500));  
a.add(new Poivron(0.150));  
a.add(new Tomate(0.20, 5));  
a.add(new Poivron(0.450, 200));  
cout << a << endl;
```

- 3) Toujours dans la **classe AssietteCrudites**, on souhaite avoir une **méthode `getPeelingWeight()`** qui donne le poids des épluchures générées lors de la préparation de cette assiette de crudités.

Par défaut, le poids d'épluchure des crudités est de 10% de leur poids.

Néanmoins, dans le cas particulier des poivrons, le poids d'épluchure est ce pourcentage (10%) auquel il faut ajouter 1 gramme par graine.

Modifiez les classes nécessaires pour réaliser ces fonctionnalités et pouvoir exécuter le code suivant :

```
AssietteCrudites a = AssietteCrudites();  
a.add(new Poivron(0.3, 10));  
a.add(new Tomate(0.25, 4));  
cout << a.getPeelingWeight() << endl;
```

- 4) On souhaite maintenant que le pourcentage du poids de la crudité perdu en épluchure (10% par défaut) puisse être paramétré globalement, pour toutes les crudités.

Ainsi, par exemple, sur la base du code précédent, on souhaite voir la différence de poids d'épluchure d'une assiette de crudités entre le paramétrage par défaut (10%) et un paramétrage à 5%.

Modifiez les classes nécessaires pour réaliser ces fonctionnalités et pouvoir exécuter le code suivant :

```
AssietteCrudites a = AssietteCrudites();  
a.add(new Poivron(0.3, 10));  
a.add(new Tomate(0.25, 4));  
cout << a.getPeelingWeight() << endl;  
Crudite::setPeelingWeight(0.05);  
cout << a.getPeelingWeight() << endl;
```

PARTIE III : On surveille ses calories

La plupart des aliments, qu'ils soient crus, cuits, nature ou transformés, contiennent des calories.

On souhaite maintenant disposer d'une méthode indiquant le nombre de calories d'un aliment donné.

On suppose que:

- les tomates contiennent 21 calories pour 100 grammes;
- les poivrons contiennent 19 calories pour 100 grammes, moins 1 calorie par graine;
- le calcul du nombre de calories se fait sur chaque crudité sans tenir compte du poids des épluchures, et produit un nombre entier (pas un nombre flottant);
- on ne peut pas définir le nombre de calories d'une crudité sans savoir précisément de quelle crudité il s'agit;
- une assiette de crudités contient autant de calories que les crudités qui entrent dans sa composition.

Modifiez les classes nécessaires pour réaliser ces fonctionnalités et pouvoir exécuter le code suivant (attention: toutes les fonctionnalités décrites dans les deux exercices précédents doivent être maintenues) :

```
Poivron p = Poivron(0.3, 10));
cout << p.getCalories() << endl;
Tomate t = Tomate(0.25, 4))
cout << t.getCalories() << endl;
AssietteCrudites a = AssietteCrudites();
a.add(&p);
a.add(&t);
cout << a.getCalories() << endl;
```