

Java Avancé
Master 1 Informatique UPEC 2015/2016
TD 2 : Vers les lambda

Considérez les classes suivantes (dont on omet les constructeurs et les getteurs) :

```
public class Apple {
    private final Integer weight;
    private final String color;
    private final Farm origin;
}

public class Farm {
    private Integer nTrees;
    private Integer price;
    private List<Market> destinations;
}

public class Market {
    private String city;
    private Integer nStalls;
}
```

Exercice 1: *Listes et Ensembles*

Dans une classe TD2M1 écrivez

- une méthode qui prend en entrée une liste de pommes et renvoie l'ensemble des couleurs des pommes de la liste
- une méthode qui prend en entrée une liste de pommes et renvoie l'ensemble des fermes d'origine
- une méthode qui prend en entrée une liste de pommes et renvoie l'ensemble des villes de destination
- une méthode qui prend en entrée une liste de pommes et renvoie le prix total de toutes les pommes
- une méthode qui prend en entrée une liste de pommes et renvoie la somme de tous les stands qui se trouvent dans tous le marché de destination

Exercice 2: *Paramétrisation et lambda (1)*

Créez une interface `Inter1<T>` avec une méthode abstraite `T f (Apple a);`

- Dans la classe TD2M1 écrivez une méthode générique en T qui prend en entrée une liste de pommes et une instance de `Inter1<T>` et renvoie l'ensemble des éléments de type T qui correspondent aux pommes de la liste selon la méthode f.
- Réalisez les deux premières méthodes de l'exercice précédent en instanciant l'interface `Inter1<T>` avec une classe anonyme.
- Faites la même chose avec des lambdas

Exercice 3: *Paramétrisation et lambda (2)*

Généralisez l'exercice précédent pour qu'il s'applique à d'autres types que les pommes. Avec une classe anonyme réalisez une méthode qui prend en entrée une liste de String et qui renvoie l'ensemble des troisièmes caractères. Faites la même chose avec un lambda

Exercice 4: *Paramétrisation et lambda (3)*

Créez une interface **Inter2<T>** avec une méthode abstraite **Set<T> g (Apple a);**

- Dans la classe **TD2M1** écrivez une méthode générique en **T** qui prend en entrée une liste de pommes et une instance de **Inter2<T>** et renvoie l'ensemble des éléments de type **T** qui correspondent aux pommes de la liste selon la méthode **g**. Attention : cette méthode doit renvoyer un ensemble de **T**, et non pas un ensemble d'ensembles.
- Réalisez la troisième méthode du premier exercice en instanciant l'interface **Inter2<T>** avec un **lambda**.
- Généralisez l'interface et la méthode pour qu'elles s'appliquent à d'autres types que les pommes. À l'aide de cette interface réalisez une méthode qui prend en entrée une liste de **String** et qui renvoie l'ensemble des tous les caractères qui apparaissent dans la liste.

Exercice 5: *Paramétrisation et lambda (4)*

Créez une interface **Inter3** avec une méthode abstraite **Integer h (Apple a);**

- Dans la classe **TD2M1** écrivez une méthode qui prend en entrée une liste de pommes et une instance de **Inter3** et renvoie la somme de tous les entiers qui correspondent aux pommes de la liste selon la méthode **h**.
- Réalisez la quatrième méthode du premier exercice en instanciant l'interface **Inter3** avec un **lambda**.
- Comment généraliser cet exercice pour qu'on puisse appliquer d'autres opérations que la somme ? Pour qu'elle s'applique à d'autres types que **Integer** ?

Exercice 6: *Paramétrisation et lambda (5)*

Comment généraliser la cinquième méthode ?