Rapport Projet rattrapage Scala

# Rappel sujet

Le typeclass BiFunctor est globalement la même chose que Functor, mais pour les type constructors à 2 trous F[\_, \_] (exemple : Either) où tu veux pouvoir transformer la gauche et la droite.

1. Ecrire une structure de données Or[A, B] qui permet de représenter un Sum type, qui est soit First[A], soit Second[B]. En Scala tu pourras écrire indifféremment "A Or B" ou "Or[A, B]"
2. Ecrire un typeclass BiFunctor, qui contient une seule fonction bimap, et qui permet de transformer A en B, et C en D, pour un type F[A, C], et retourne F[B, D]
3. Ecrire l'instance de BiFunctor pour le type Or
4. Ecrire une fonction "convertToInt" qui prend une chaine de caractères et retourne Or[String, Int] en fonction de si la chaine de caractères peut-être transformée en Int ou non
5. Je veux pouvoir écrire : convertToInt("1").bimap(\_.length, \_ + 1). Fais ce qu'il faut pour.
6. Ajoute des tests unitaires à tout ça.

# Réalisations

**Structure de données**

J’ai créé un « sealed abstract class Or[+A , +B] » qui a deux case classe :

**final case class** First[A](value: A) **extends** Or[A , Nothing] {  
  
}  
  
  
**final case class** Second[B](value: B) **extends** Or[Nothing , B]

Ensuite j’ai ajouté un « Campanion object » à ma structure de données.

**Typeclass BiFunctor**

J’ai défini un trait Bifunctor qui a une seule méthode « bimap » définie comme suit :

**def** bimap[A, B, C, D](fab: F[A,B])(f: A => C , g: B => D): F[C,D]

J’ai aussi défini un Bifunctor pour mon type Or, ceci en définissant un implicit comme suit :

**implicit val** *OrBifunctor*: Bifunctor[Or] = **new** Bifunctor[Or]

**Fonction de conversion**

J’ai défini une fonction de conversion d’une chaîne de caractères en entier. Elle renvoie First d’un entier si la chaîne est convertible sinon Second de la même chaîne en entrée.

**Ajout de la méthode au type de données existant**

Pour ajouter la méthode « bimap » à mon type « Or » que j’ai défini, j’ai utilisé un implicit. J’ai défini un « object Implicits » qui a un « implicit class OrPlus » définissant une fonction « bimap »

# Difficultés rencontrées et solutions

Tout au long du projet j’ai fait face à des difficultés :

**Comprendre exactement ce qui est demandé (ce qu’il faut faire)** : Au début et à chaque fois que j’avançais le contexte n’était pas trop clair pour moi et pour ce fait je demandais des explications et je posais des questions pour être sûre de bien comprendre ce qui est demandé.

**Corriger les erreurs du code**: j’avais de temps en temps eu besoin d’être orientée pour corriger les erreurs dans mon code.

**Test unitaire** : J’ai eu besoins d’aide pour comprendre que la version de Scala utilisée peut favoriser ou non l’ajout des dépendances afin d’importer les packages nécessaires. En plus je ne maîtrise pas encore bien la définition d’un unit test en scala (c’est la partie que j’ai le moins discuté avec le professeur).