DEVELOPPEMENT D'APPLICATION WEB

JAVA - PRISE EN MAIN

OBJECTIFS

A travers cet exercice, vous allez apprendre à :

- Prendre en main la syntaxe du langage Java,
- Manipuler les fondamentaux de la POO en Java,
- Gérer les exceptions d'une application.

PREMIER PROGRAMME

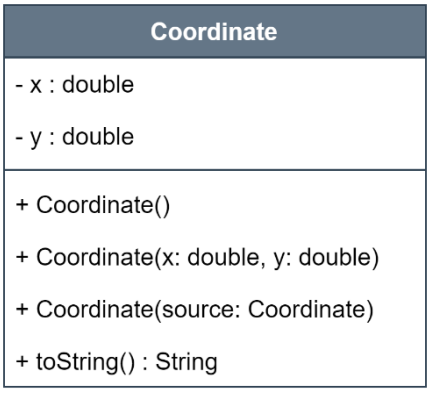
- Créez un fichier « Application.java » qui contiendra une classe **`Application`** avec pour seule fonction la fonction **`main`** (voir le cours)
- Utilisez la fonction **`println`** de l'objet **`out`** du package **`System`** pour afficher :

On fait du Java à Polytech Dijon !

- Exécutez et testez que tout fonctionne correctement.

COORDONNEES

- Créez un fichier « Coordinate.java » qui contiendra une classe… **`Coordinate`** (si vous aviez deviné, c'est que vous avez pigé le truc. Sinon… Pigez le truc !).
- Implémentez la classe **`Coordinate`** selon la modélisation suivante :



La méthode **`toString`** retournera une chaine de caractères au format : « (x ; y) »

- Dans la fonction **`main`**, instanciez un objet de type **`Coordinate`**
- Avec la fonction **`println`**, affichez le résultat de la méthode **`toString`** pour l'objet instancié

```
System.out.println(myCoordinate.toString()) ;
```

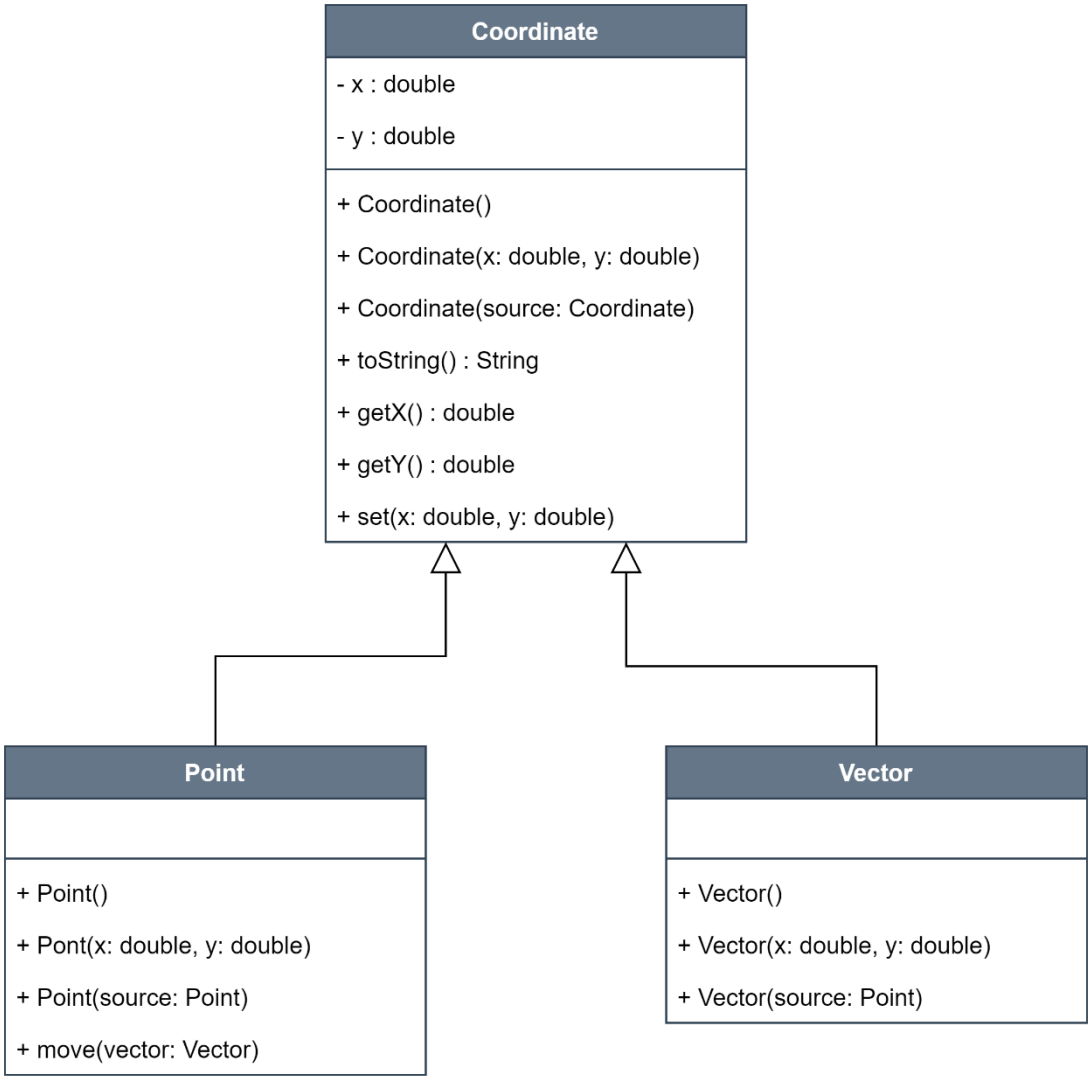
- Exécutez et testez le fonctionnement
- Recommencez, mais cette fois-ci, sans l'appel à **`toString`** :

```
System.out.println(myCoordinate) ;
```

- Que constatez-vous ?

HERITAGE

- Ajoutez les accesseurs **`getX`**, **`getY`**, et **`set`** à la classe **`Coordinate`**, et implémentez les classes ci-dessous (si vous avez « pigé le truc », une classe = un fichier) :



Note : en UML, la flèche fermée vide représente une relation d'héritage. Ici, **`Point`** hérite de **`Coordinate`**, tout comme **`Vector`**.

- Dans la fonction main, instanciez un **`Point`** et un **`Vector`**.

Note : la méthode **`super`** permet d'appeler l'un des constructeurs de la classe mère depuis le constructeur de la classe fille.

- Déplacez le point avec le vector créé.
- Affichez les coordonnées du point avant et après son déplacement.
- Exécutez et testez le bon fonctionnement.

SAISIE UTILISATEUR

La classe Scanner (package **`java.util.Scanner`**) permet d'extraire des données à partir d'un flux d'entrée. Connectée à l'entrée standard (ici le clavier), elle pourra être utiliser pour effectuer des saisie utilisateur au clavier.

- Videz le code de la fonction **`main`**
- Instanciez un objet de type **`Scanner`** :

```
Scanner keyboard = new Scanner(System.in);
```

La classe Scanner fournit plusieurs méthodes pour extraire différents types de données :

- nextLine : **`String`**
- nextInt : **`int`**
- nextDouble : **`double`**

- Faites saisir différents types d'information à l'utilisateur et affichez les valeurs obtenues.
- Exécutez et testez le fonctionnement.
- Que se passe-t-il si vous saisissez une donnée qui ne correspond pas au format attendu (par exemple « toto » pour la saisie d'un entier) ?

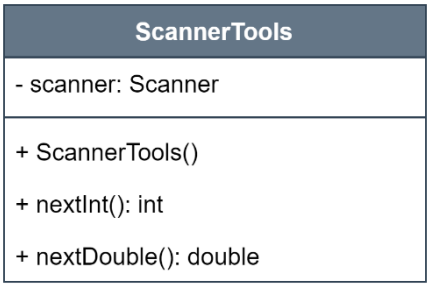
GESTION DES EXCEPTIONS

Lorsqu'une exception est levée, deux scénarios sont possibles :

- Soit elle est capturée à l'aide d'un bloc **`try/catch`**
- Soit elle n'est pas capturée et l'application s'arrête avec perte et fracas.

Or, les utilisateurs n'aiment pas le fracas et encore moins la perte de leur travail. Il est donc nécessaire de capturer les exceptions lorsqu'elles se produisent et d'offrir à l'utilisateur une alternative au crash de son application.

- Créez la classe **`ScannerTools`** suivante :



Les fonctions **nextInt** et **nextDouble** permettront de saisir respectivement un **int** ou un **double**.

Tant que l'utilisateur effectuera une saisie incorrecte, le message « Saisie incorrecte » s'affichera puis une nouvelle saisie sera demandée.

Attention : lorsque le format de données présent dans le flux d'entrée ne correspond pas au format demandé, le contenu du flux reste inchangé. Pensez à videz le flux en utilisant la méthode **nextLine** avant de redemander une saisie à l'utilisateur.

- Testez l'utilisation de **ScannerTools** dans la fonction main.

LISTE

La classe **ArrayList** est un équivalent Java à la classe **vector** en C++ :

```
ArrayList<int> integers = new ArrayList<int>() ;
```

- A partir des éléments précédents, créez une application qui
  - demandera à l'utilisateur combien de points ils souhaitent saisir
  - demandera à l'utilisateur les coordonnées x et y de chaque point
  - créera un point avec les informations saisies par l'utilisateur et l'ajoutera à la liste
  - enfin, affichera la liste de tous les points saisis.

BONUS DU DEVELOPPEUR

La classe **ScannerTools** présente de la redondance de code (les méthodes **nextInt** et **nextDouble** sont quasiment similaire).

- Trouvez une solution pour supprimer cette redondance de code.