Java – Recueil d'exercices

Chapitre 1 : environnement de dev., bases du langage, socle de P.O.O.

ISIE – 2023-2024

Il s'agit d'une version temporaire. Le PDF mis en ligne sera régulièrement mis à jour avec le contenu des séances suivantes.

Préambule

Pour vous aider à y voir clair, les exercices son numérotés pas séance de TP (Exo 1.x = TP1, Eco 2.x = TP2…). Pour certains exercices, des questions bonus sont rajoutées pour sustenter les plus rapides et les plus efficaces parmi vous. Elles peuvent également vous servir à travailler entre les séances, consolider vos acquis et étendre vos compétences.
Des exercices « pour aller plus loin » pourront également apparaître parfois. Ils sont facultatifs et sont proposer pour les plus curieux parmi vous.

Exercice 1.1– Hello Polytech

#helloWorld #outils #premiereClasse #IDE #horsIDE

1. Ecrire le programme de l'exemple 1.0 du cours dans l'IDE IntelliJ.
2. Exécuter le programme depuis l'IDE.

Questions bonus[1] *:*

3. *Recommencer les deux opérations (la compilation et l'exécution) en ligne de commande, i.e. hors de tout IDE, en utilisant explicitement les outils du JDK tels que le compilateur (commande `javac`) ou la JVM (commande `java`)*

Le tutoriel suivant pourrait vous être utile …

- http://docs.oracle.com/javase/tutorial/getStarted/cupojava/win32.html *(ou […]/unix.html)*
- http://docs.oracle.com/javase/tutorial/getStarted/problems/index.html

Si vous êtes dans la situation où vous avez défini explicitement un package (c'est une bonne chose !), la syntaxe est légèrement plus complexe :

Exemples d'appels standards (pour une classe `Test` dans un package `com.polytechtours`) :

```
>javac com\polytechtours\Test.java
>java com.polytechtours.Test
```

4. *Quand nous aurons vu explicitement la notion de package (en fin de séance #1 a priori), recommencez les questions 1, 2 et 3 avec une classe définie dans un package, disons tp1 par exemple.*

[1] les **questions bonus** ne sont à faire que pour les plus rapides, ou pour pratiquer/réviser en dehors des séances.

Exercice 1.2 – mon nom est `Person`

#POJO #defaultConstructor #getters #setters

Nous allons écrire une classe `Person` dont une instance représente une… personne. (Oui, on va la faire en anglais). On est intéressé par les informations suivantes : son nom, son prénom, son age, s'il est étudiant ou pas à Polytech (oui/non).

1. Ecrire la classe `Person` en commençant par y inclure les 4 attributs (à définir privés). Ajouter une méthode publique `addOneYearToAge()` qui augmente l'age de la personne d'un an.

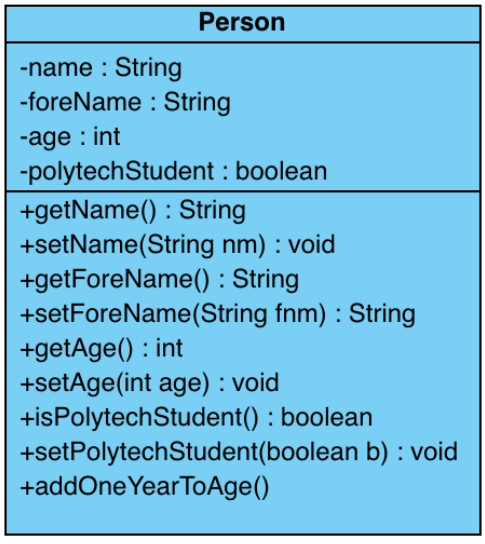
Nous allons tester cette classe avec la méthode `main()` suivante :

```java
public static void main(String[] args) {
    Person p1 = new Person();
    p1.setName("Smith");
    p1.setForeName("Jane");
    p1.setAge(21);
    p1.setPolytechStudent(true);

    System.out.print("Je suis "+p1.getForeName()+" "+p1.getName()+", j'ai "+p1.getAge()+" ans. ");
    if(p1.isPolytechStudent())
        System.out.println("Je suis étudiant(e) de Polytech.");
    else
        System.out.println("Je ne suis pas étudiant(e) de Polytech.");

}
```

2. Ecrire cette méthode `main()`, directement dans votre classe `Person`. Vous devez voir des erreurs s'afficher (en rouge), c'est normal, la question suivante devrait y remédier.

3. Ajouter un getter et un setter publique pour chacun des attributs de votre classe. Vous devriez avoir une classe avec à peu près cette structure :

4. Testez l'exécution du `main()`.



Exercice 1.3 – Un peu de géométrie

#POJO #constructeur #sysout

1. Ecrire une classe `Circle` disposant des attributs privés ci-dessous (de type `float`) ainsi que les accesseurs et modificateurs associés :
   - les coordonnées du centre du cercle en abscisse et en ordonnée,
   - le rayon du cercle,

2. Ajouter une méthode permettant de calculer et retourner le diamètre. Même chose pour le périmètre et pour la surface.
   NB : Utilisez la constante `PI` définie dans la classe `java.lang.Math`

3. Ajouter un constructeur initialisant les coordonnées du centre et le rayon à la création de l'instance.

4. Aurait-ce été pertinent d'ajouter un attribut pour le diamètre, un autre pour le périmètre et un pour la surface ? Pourquoi ?

5. Ecrire une méthode `main()` qui construit un cercle de rayon 18 et dont le centre est de coordonnées (34,12.3). Testez vos méthodes.

6. Ecrire dans `Circle` une méthode (d'instance) `public void printMe()` qui affiche sur la sortie standard les caractéristiques de l'instance.
   Exemple de résultat attendu :

```
Je suis un cercle dont voici les caractéristiques :
 coordonnées de mon centre : (34.0, 12.3)
 rayon : 18.0
 périmètre : 113,097
 surface : 1017,876
```

Questions bonus :

7. De la même manière, écrire une classe Rectangle définie par les coordonnées de son coin inférieur gauche, sa largeur et sa hauteur. Intégrez également le calcul de son périmètre et de sa surface. Complétez votre classe avec les accesseurs et modificateurs requis, et la méthode `printMe()`.