

Programmation Par Objets et Java - TD 1

Environnement

Adresse de ce cours : ENT, arche, Sciences et Technologies, IUT Metz Dpt Info, "ppo lp wce". Ce TD peut se faire sous Windows ou sous Linux, en utilisant Eclipse. Faire chaque exercice dans une classe ou procédure de classe (selon ce qui vous semble le plus approprié). **Garder tous les exercices !**

- Créer un projet Java : Menu Fichier, Nouveau... Projet, Projet Java, lui donner un nom (lp_ppo) et cliquer sur Terminer ;
- Dans ce projet, créer un paquetage : clic bouton droit sur le projet, Nouveau, Paquetage, lui donner un nom (td1) et cliquer sur Terminer ;
- Y ajouter une classe : clic bouton droit sur le projet, Nouveau, Classe, lui donner un nom et cliquer sur Terminer.

A rendre

Dépôt à chaque fin de td sur bitbucket, dans votre repository **ppo2017_VotreNom** ; invitation à faire à l'utilisateur **laroche5**. Veillez à bien renseigner le commentaire au moment du commit (Version finale ou pas notamment).

Aucun copier-coller n'est autorisé : vous devez structurer votre code !

1 Chaînes, entrées-sorties, dates

Exercice 1

Créer une classe **Bonjour** qui affiche **Bonjour** !

Donner la liste des classes et méthodes utilisées pour faire cet affichage. Pour répondre à cette question, consulter l'API Java disponible sur le site d'*Oracle* :

<http://docs.oracle.com/javase/8/docs/api/>

Exercice 2

Modifier la classe précédente pour qu'elle réalise les traitements suivants :

- Saisie du prénom et du nom de l'utilisateur ;
- Affichage de **Bonjour *prénom nom***

Saisir le prénom en passant par la classe `Scanner` et le nom en utilisant les classes `BufferedReader` et `InputStreamReader`.

Quels sont les avantages et inconvénients des deux solutions ? Pourquoi la seconde est-elle si compliquée ?

Exercice 3

Modifier l’**affichage** de façon à ce que les noms et prénoms respectent les règles typographiques en vigueur : prénom commençant par une majuscule, et nom tout en majuscules (voir la classe `String` dans l’API).

Exercice 4

Modifier la classe de l’exercice 2 : les règles typographiques doivent être respectées lors de la **saisie** (refaire la saisie en boucle tant que la saisie est incorrecte).

Exercice 5

Repartir de la classe de l’exercice 3. Ecrire deux méthodes (au sens de sous-programme) de saisie d’un **entier** (pas forcément une année, vos variables doivent donc être renommées en conséquence !) :

- avec `BufferedReader` (saisie d’une chaîne puis conversion en entier avec une méthode de la classe enveloppe `Integer`) ;
- avec `Scanner`.

Utiliser successivement les deux méthodes pour saisir l’année de naissance de l’utilisateur, puis calculer l’âge en soustrayant l’année saisie à 2017 et afficher

Bonjour *prénom nom*, tu as *âge* ans

Que se passe-t-il si on saisit une chaîne ou un réel au lieu d’un entier, dans chacune des deux méthodes ?

Exercice 6

Sécuriser la saisie grâce à une gestion d’exception (try...catch) et à une boucle répétant la saisie tant qu’il y a un problème.

Exercice 7

Ajouter une méthode qui lit un réel, avec la méthode qui vous semble la plus simple.

Exercice 8

On a maintenant dans notre classe un programme principal et deux méthodes de saisie (entier et réel). Regrouper ces méthodes au sein d’une classe **Saisie**, et ajoutez-y une méthode de saisie de chaîne, afin de pouvoir les réutiliser à tout moment. Modifier la classe écrite pour l’exercice 5 en appelant ces méthodes.

Exercice 9

Lorsqu'on demande une saisie à l'utilisateur, on le prévient généralement à l'aide d'un message. **Surcharger** les méthodes de la classe **Saisie** pour qu'elles affichent un message passé en paramètre avant la saisie. Utiliser ces méthodes lors de la saisie du nom, du prénom et de l'année de naissance.

Réfléchissez à ce que veut dire "surcharger" !

Exercice 10

Après la saisie de l'année de naissance, affichage de la date courante au format : 07/11/17. Calculer l'âge en fonction de cette date.

Afficher ensuite la date au format : mardi 7 novembre 2017.

Vous devez utiliser les nouvelles classes de Java 8 (voir les liens sur le cours Arche).

Tables

Les exercices 11 à 14 doivent être réalisés sans utiliser de classe de l'API java (à part **Scanner**).

Exercice 11

Ecrire une fonction qui permet de saisir les éléments d'une table d'entiers, et une procédure d'affichage.

```
public static int[] saisieTable()
public static void afficheTable(int[] tab)
```

Exercice 12

Ecrire une fonction qui copie le contenu du premier tableau dans un nouveau tableau.

```
public static int[] copieTable(int[] tab)
```

Exercice 13

Ecrire une procédure de tri de table (tri à bulles).

Exercice 14

Ecrire une fonction qui renvoie la position d'un élément de la table.

Exercice 15

Proposer une procédure englobant les exercices 12 à 14, qui s'écrit en 3 lignes (voir la classe **Arrays**).