Java : Partie 1

Acquérir les concepts fondamentaux et la syntaxe de base en Java

|  |
| --- |
| ***Mini-Projet 1 : Emergence des concepts objet en Java*** |

**Objectifs :**

* Emergence des concepts fondamentaux de la « Programmation Orientée Objet » déjà appréhendés lors des sessions précédentes (PHP) par des manipulations concrètes (ordinateur éteint)
* Formulation en langage simple des notions de classe ; classe abstraite ; objet ; attribut ; méthode ; héritage ; polymorphisme…

**Modalités :**

* Travail par îlot, puis par classe mixée
* Production individuelle
* Durée ~2 heures

**Consignes**

|  |
| --- |
| * Faire une liste de caractéristiques vous définissant bien (qualités, défauts, caractéristiques physiques ou intellectuelles), et des capacités et/ ou aptitudes personnelles (mode de transport pour venir au campus ? Je fais du vélo, je code, j’aime l’informatique…) de chaque élève de l'îlot. Utiliser des post-its. * Synthèse par salle : Au tableau, les élèves viennent coller leurs post-its, et effectuent un regroupement par thème… * À partir de cette liste, l’animateur fait émerger les différents concepts objet : les élèves effectuent des regroupements selon des critères ou catégories. Les concepts objet sont amenés en faisant l'analogie avec les regroupements effectués par les élèves. |

|  |
| --- |
| **Livrable :**   * Document proposant un premier niveau de définition « en langage simple » des différents concepts objets (environ 2 lignes par définition). A déposer sous le nom ***livrable1,***  dans votre répertoire individuel sous ***java→livrables*** |

|  |
| --- |
| ***Projet 1 : Affichage des profils*** |

### **Modalités :**

* Travail en autonomie
* Production individuelle

### **Objectifs de l’activité :**

* Prendre en main l’environnement de développement NetBeans
* Manipuler la syntaxe de base de Java
* Manipuler les concepts objets du langage Java
* Générer la Javadoc
* Validation des compétences n° 1, 2 et 14

### **Consignes :**

1. **Etude de tutoriels :**

|  |
| --- |
| * Codecademy : Introduction to Java, Conditional and Control, Object-oriented Java ou Sololearn : Basic Concepts, Conditionals and Loops, Arrays * OpenClassrooms : « Les variables et les opérateurs », « Lire les entrées clavier », «Votre première classe», « Les tableaux» |

1. **Manipulation de la syntaxe de base de Java**

|  |
| --- |
| * Dans la classe « Main » : renseigner et afficher des informations de profil qui sont fournies dans des variables (Nom, prénom, année de naissance..). L’affichage se fait via des println. * Même consigne mais cette fois les informations seront saisies via lecture des entrées clavier (classe Scanner). * Faire la saisie de plusieurs utilisateurs (boucle) et les afficher au fur et à mesure jusqu’à ce que l’utilisateur choisisse volontairement de quitter le programme. |

Servez-vous de ce que vous avez appris sur cet exercice pour créer le programme “Reseau\_social”. Ce premier programme constituera le squelette de votre réseau social. Vous l’enrichirez au cours des projets proposés tout au long de la formation.

1. **Spécifications du programme “Reseau\_social”**

|  |
| --- |
| * Le programme commence par vous demander les informations pour créer l’utilisateur correspondant. * Il propose ensuite plusieurs choix d’actions :   + Afficher votre profil ? (nom, prénom, etc.)   + Modifier vos infos personnelles ?   + Ecrire un message ? (on suppose que l’on a un tableau de messages pour chaque utilisateur, qui est un tableau de String)   + Afficher un message ?   + Ajouter un ami ? (on suppose que l’on a un tableau d’amis, qui est un tableau de String)   + Voulez-vous afficher le nom d’un ami ? * Faire la saisie de plusieurs utilisateurs (boucle) jusqu’à ce que l’utilisateur choisisse de quitter le programme. |

1. **Etude de tutoriels :**

|  |
| --- |
| * Codecademy : Object-oriented Java ou Sololearn : Classes and Objects * OpenClassrooms : « Votre première classe» |

1. **Evolution du programme :**

|  |
| --- |
| * Dans une seconde version de ce programme, vous utiliserez une classe “Utilisateur” qui aura 3 constructeurs (sans paramètres, avec nom/prénom, et avec toutes les infos) |

**Ne pas oublier**

|  |
| --- |
| * Testez votre programme de manière exhaustive * Documentez vos classes et générez la Javadoc |

Travail complémentaire : créer de même une classe « Post » (indépendante de la classe “Utilisateur”) et ses méthodes associées. Créer une classe menu pour manipuler les menus.

|  |
| --- |
| **Livrable :**   * Codes commentés des classes « Main », « Utilisateurs » (et éventuellement « Post »). A déposer sous git. * Javadoc de votre projet à déposer dans ***java→livrables*** * Glossaire des différentes syntaxes utilisées. A déposer sous le nom ***glossaire***, dans votre répertoire individuel sous ***java→livrables*** |

### **Ressources :**

* *Codecademy :*
  + - Introduction to Java, Conditionals and Control flow, Object-Oriented Java
    - <https://www.codecademy.com/learn/learn-java>

ou

* *Sololearn:*
  + - Basic Concepts, Conditionals and Loops, Arrays, Classes and Objects
    - <https://www.sololearn.com/Course/Java/>
* *Openclassrooms :* 
  + - <https://openclassrooms.com/courses/apprenez-a-programmer-en-java/les-variables-et-les-operateurs>
    - <https://openclassrooms.com/courses/apprenez-a-programmer-en-java/lire-les-entrees-clavier>
    - [https://openclassrooms.com/courses/apprenez-a-programmer-en-java/les-tableaux-3](https://openclassrooms.com/courses/apprenez-a-programmer-en-java/les-tableaux-3*)
    - <https://openclassrooms.com/courses/apprenez-a-programmer-en-java/votre-premiere-classe>
* *Tutoriel d’Oracle pour Java :* 
  + - <http://docs.oracle.com/javase/tutorial/java/index.html>
* *Astuces NetBeans :* 
  + - <http://es.ece.fr/wp/wp-content/uploads/ESRaccourcis-java_NetBeans.pdf>

|  |
| --- |
| ***Synthèse (activité de groupe)*** |

### **Modalités :**

* Travail en groupe

### **Objectifs de l’activité :**

* Repérer les erreurs classiques dans le code Java

### **Déroulement :**

* Slides fournis pour la recherche d’erreurs
* Recherche individuelle puis mise en commun