Université Lyon 1

LIF 13 : TP de manipulation Java/UML sous NetBeans

Compétences indispensables à acquérir :

- Savoir mettre en place un projet Java sous l'IDE (Integrated Development Environment) NetBeans (création projet, définition d'un *main,* exécution paramétrée ou non)
- Savoir se documenter : utiliser les API de java, utiliser les fonctions de documentation de NetBeans

!!! N'hésitez pas à questionner vos encadrants si vous rencontrez des difficultés à vous approprier une des compétences décrites !!!

Informations complémentaires :

Création d'un projet NetBeans :

Fichier -> New Project

Sélectionner dans la catégorie Java un projet de type « Java Application ». Choisir le nom de la classe contenant le « main » afin de répondre à l'énoncé.

Partie I : Premier Programme

1. Ecrire le programme java *Somme (classe Somme)* qui permet de calculer et d'afficher à l'écran la somme des 10 premiers entiers

Ce programme affiche à l'écran: La somme des 10 premiers entiers est : 55

2. Modifier ce programme pour que celui-ci puisse prendre en paramètre le nombre d'entiers à sommer.

2.1 : Modification de la fonction main :

Dans la fonction « public static void main(String[] args) », args[0] représente le premier paramètre passé au programme sous la forme d'une chaîne de caractères.

Transformation d'une chaîne en entier :

int monEntier = Integer.parseInt(maChaîne); // par exemple, maChaîne = « 4 »

2.2 : Appeler la fonction main avec paramètre :

- *depuis NetBeans (ou eclipse) :* explorer les paramètres d'exécution du projet (click droit sur le nom du projet depuis l'arborescence, puis sélectionner les propriétés)
- en ligne de commande : trouver le répertoire contenant les classes compilées (Byte-Code), et exécuter : « java monPackage.Somme 55 »

2.3 : Modifier le programme de façon à ce que la somme soit désormais celle de la série suivante : 1 + 1/2 + 1/3 + 1/4 + ... + 1/n
Veuillez à ce qu'il n'y ait pas de perte de précision.

Partie 2 : Premier objet composite Moto

Soit une moto une composition d'un moteur, de deux phares et de deux roues.

1. Proposer un diagramme de classes et un diagramme d'objets. Valider ceux-ci avec votre encadrant.

2. Implémenter les différentes classe de façon simplifiée (développer uniquement le constructeur de chaque classe dans un premier temps)

Le constructeur de la Moto se charge de préparer l'ensemble des objets composants utiles (phares, etc.), et d'en enregistrer les références.

3. Ajouter les fonctionnalités suivantes :

- un moteur peut être dans un état démarré ou éteint (par défaut). Les phares peuvent aussi être éteints ou allumés. Proposer les fonction permettant de faire varier ces états depuis un objet (ou une instance) de la classe Moto.

- une moto peut accélérer. Pour cela, lorsque l'on demande l'exécution de la fonction accélérer de la classe Moto, l'ordinateur de bord de la moto vérifie que le moteur est disponible (allumé) et que les phares sont allumés avant d'appeler la fonction d'accélération propre à l'objet Moteur.

à chaque fois qu'un des objets du modèle effectue une action, il l'affiche grâce à la fonction :

System.out.println(« Message à afficher ») ;

4. Développer un scénario dans une fonction Main permettant de tester le bon fonctionnement de votre programme

5. Comprendre les références mémoire utilisées en Java utilisées pour manipuler les objets

Soit le code suivant dans votre Main :
Moto maMoto = ….

Moto maMoto 2 = maMoto ;

maMoto et maMoto2 sont-il différents ? Pourquoi ? Comment pourrions-nous créer une copie indépendante de ma maMoto (copie en cascade des phares, du moteur, etc.) ? Implémentez votre proposition et testez.

Partie 3 : Manipulation de l'IDE

Utiliser l'exercice de la partie 2 pour les manipulations ci-dessous :

1. Trouver comment modifier le nom d'une classe/fonction/etc. Dans tous les fichiers concernés d'un projet automatiquement.

2. Trouver le raccourci permettant d'afficher les différentes fonctions disponible pour un objet (lorsque le curseur est sur l'objet)

3. Trouver comment accéder par un clic (avec un bouton maintenu) au code d'une fonction appelée

4. A quel endroit définit-on le JDK utilisé ?

5. Familiarisez vous avec la fonction de débogage

Partie 4 :

1. Pour chaque cas ci-dessous, le code est-il correct ? Si oui, pourquoi ?

- String s = « j'ai mangé » + 4 + « pommes » ;

- int somme = «quatre » + 2 ;

2. Soit les instructions : « int a = 2 ; int b = a ; a = a + 1 ;», que vaut b ? pourquoi ?

3. Soit les instructions « Integer a = new Integer(2) ; Integer b = a ; a = null ; » que vaut b ? pourquoi ?