

Interface Homme-Machine en mode textuel

TD Java n°10 – Eisti 2009-2010

Le but de ce TD est de saisir interactivement des données pour créer des objets en mode console. Il est conseillé d'exécuter votre programme directement à partir d'une console Linux.

1 – Utilisation des paramètres en ligne de commande

- utiliser les paramètres en ligne de commande de la méthode main de votre application pour récupérer le nom et les coordonnées d'un Point ;
- créer un point à partir de ces paramètres ;
- afficher le point créé ;
- signaler tout problème en affichant un message sur la sortie erreur avant de quitter l'application.

Les exercices 2 et 3 devront être réalisés 3 fois en utilisant les 3 possibilités de lecture suivantes :

- lecture avec la classe BufferedReader (historique) ;
- lecture avec la classe Scanner (Java 5) ;
- lecture avec la classe Console (Java 6).

2 – Lecture clavier d'une chaîne quelconque :

- écrire une fonction lireChaine qui prend en paramètre un message d'invite (prompt), permet la saisie d'une chaîne de caractère au clavier par l'utilisateur et renvoie la chaîne lue ;
- tout problème éventuel sera propagé au-delà de la fonction
- tester cette fonction dans une 2ème application qui demande de saisir le nom d'un Point à l'utilisateur et affiche ce nom. Signaler tout problème en affichant un message sur la sortie erreur avant de quitter l'application.

3 – Lecture clavier d'une valeur numérique :

- écrire une fonction lireEntier qui prend en paramètre un message d'invite (prompt), permet la saisie d'un entier au clavier par l'utilisateur et renvoie l'entier lu ;
- les problèmes liés à la saisie d'un texte erroné ne correspondant pas à une valeur entière devront être repris dans la fonction en repropoant à l'utilisateur une autre saisie jusqu'à ce que celle-ci soit correcte. Tout autre type de problème sera propagé au-delà de la fonction ;
- tester cette fonction en ajoutant à votre 2ème application la saisie des coordonnées d'un Point à l'utilisateur. L'application crée ensuite un objet Point et l'affiche. Signaler tout problème en affichant un message sur la sortie erreur avant de quitter l'application.

4 – Sauvegarde d'un Point dans un fichier texte :

- écrire une fonction sauvegarderPoint qui prend en paramètre un Point et un nom de fichier. Cette fonction ajoute à la fin du fichier 3 lignes contenant respectivement le nom, l'abscisse et l'ordonnée du Point. Tout problème éventuel sera propagé au-delà de la fonction ;
- ajouter à l'application précédente la sauvegarde du point saisi après avoir proposé à l'utilisateur de rentrer le nom du fichier ;
- en cas de problème de sauvegarde, afficher à l'utilisateur le problème constaté et proposez-lui une autre sauvegarde optionnelle (il se peut qu'on ne puisse vraiment pas écrire) ;

- vérifier le contenu du fichier de sauvegarde après l'exécution de votre application.

5 – Lecture de Points dans un fichier texte

- écrire une fonction lirePoints qui prend en paramètre un nom de fichier et renvoie la liste des points lus dans ce fichier ;
- écrire une application qui utilise la fonction précédente pour lire une liste de points dans un fichier dont le nom est passé en ligne de commande et affiche cette liste. Le programme signalera sur la sortie erreur tout problème éventuel.