

Nom : _____
Prénom : _____
Groupe : _____

/ 20

Haute École de Bruxelles
École Supérieure d'Informatique
Bachelor en Informatique

14 novembre 2014
DEV1 – LAJ
CLR

DEV1 – Laboratoires de Java

Interrogation 3

Binaire

- Vous réaliserez votre travail sur **linux1** et le déposerez dans le casier **linux** de votre professeur par la commande **casier**.
- Vous disposez de toutes vos notes ainsi que de l'aide en ligne.
- Vous avez une heure de temps.
- Il ne suffit pas que votre code compile. Testez-le pour identifier d'éventuelles erreurs à l'exécution.
- La cotation tiendra compte aussi du style de programmation que vous avez acquis.
- Respectez bien les noms de package, classe, méthodes demandés dans l'énoncé.
- Vous remplacerez bien sûr **g12345** par votre numéro d'étudiant.

1

Préparation

(sine qua non)

Créez un répertoire **evaluations/i3**. Changez les droits sur votre répertoire **i3** pour donner les permissions de lecture et d'exécution aux professeurs mais aucun droit aux autres étudiants. Appelez votre professeur pour lui montrer que vos permissions ont bien été changées.

Vous ne continuerez pas l'interro tant que cette question n'a pas été validée par votre professeur.

2

Travailler dans un package

(4 points)

Dans la suite de l'interro, votre classe fera partie du package **evaluations.i3**. Votre programme s'appellera **Calculs.java** et sera situé dans **/home/g12345/evaluations/i3/sources/** et **Calculs.class** dans **/home/g12345/evaluations/i3/classes/evaluations/i3**.

Écrivez ici :

- l'instruction qui doit contenir votre classe pour faire partie du package demandé ;

- la commande (complète et précise) que vous allez utiliser pour **compiler** votre classe ;



- la commande (complète et précise) que vous allez utiliser pour **exécuter** votre classe ;

- le contenu minimal de votre variable d'environnement *CLASSPATH*.

3

Obtenir une unité

(5 points)

Écrivez une méthode *getUnit* qui retourne la valeur des unités d'un nombre entier reçu en paramètre.

Nous voulons travailler avec des nombres binaires. Si le chiffre obtenu ne vaut pas 0 ou 1, la méthode affiche un message explicite et arrête le programme grâce à l'instruction *System.exit(1)* ;

Écrivez une méthode *main* qui testera votre méthode.

Écrivez et générez la documentation de la méthode.

4

Convertir un nombre binaire en décimal

(4 points)

Nous avons écrit pour vous une classe *Conversion* qui contient une méthode *binToDec* permettant de convertir un nombre binaire en décimal. Nous savons qu'elle se trouve sur la machine linux1 dans un des sous répertoire de **/eCours**.

La javadoc de la classe est à votre disposition dans un répertoire **doc** situé à côté de la classe.

- Quelle commande allez-vous utiliser pour retrouver ce fichier *Conversion.class* ?

- Que devrez-vous ajouter au contenu de la variable d'environnement *CLASSPATH* pour pouvoir exécuter la classe *Conversion* ?

5

Convertir un nombre binaire en décimal-suite

(2 points)

Attention, à ce stade-ci, si vous n'avez pas respecté scrupuleusement les noms de package, classe et méthode donnés, la classe *Conversion* ne fonctionnera pas.

Vous pouvez tester le bon fonctionnement de la classe *Conversion* en exécutant sa méthode *main*.

Écrivez ici la commande (complète et précise) que vous allez utiliser pour **exécuter** la méthode *main* de la classe *Conversion*.

6

Afficher un nombre binaire en décimal

(5 points)

Écrivez une méthode *afficheDécimal* qui affiche la valeur en décimal d'un nombre binaire reçu en paramètre.

Pour ce faire, vous ferez appel à la méthode *binToDec* de la classe *Conversion*.

Écrivez une méthode *main* qui testera votre méthode.

Écrivez et générez la documentation de la méthode.