IFT1176 - Session Automne, Intra

Mohamed Lokbani

IFT1176 - INTRA

Nom: _____| Prénom(s): _____|

Signature: _____| Code perm: _____|

Section: _____|

Date:20 octobre 2001

Durée: 2 heures (de 10h30 à 12h30) Local: P-310 du Pavillon Principal (UdM) et L-604 à Longueuil.

Directives:

- Toute documentation permise.
- Calculatrice (simple) permise.

- Répondre directement sur le questionnaire.

1._____/12

2._____/20

3._____/20

4._____/18

5._____/30

Total:_____/100

Question 1 (12 points)

Que va afficher en sortie, le programme suivant. Justifiez votre réponse.

```
public class IExo1 {

    public static void main(String args[]) {
      B b = new B("Exo1");
    }
}

class A {

    A() { this("IFT1176");}
    A(String a) {System.out.println(a);}
}

class B extends A {

    B(String s) { System.out.println(s);}
    B(String s,String t) {this(s+t+"9");}
    B() { super("18");}

}
```

Question 2 (20 points)

La méthode parseInt() de la classe Integer lève une exception si la valeur passée comme paramètre à l'appel ne peut pas être convertie en int. Par exemple:

A = Integer.parseInt("ma valeur");

lève une exception de type NumberFormatException.

De même, quand on essaie d'accéder à un tableau en dehors de ses limites une méthode interne de Java signale une exception de type ArrayIndexOutOfBoundsException.

Soit IExo2 la classe suivante:

```
1  class IExo2{
2        public static void main(String [] arg){
3              String premierArgument, deuxièmeArgument;
4              int a,b,c;
5              premierArgument = arg[0];
6              deuxièmeArgument = arg[1];
7              a=Integer.parseInt(premierArgument);
8              b=Integer.parseInt(deuxièmeArgument);
9              c = a*b;
10             System.out.println("Le produit de "+a+" et "+b+" est égal à "+c);
11       }
12 }
```

-Q2.1- En exécutant la classe IExo2, en ne donnant aucune valeur d'entrée :

```
prompt> java IExo2
```

Nous obtenons une erreur d'exécution. Précisez la ligne où l'erreur s'est produite, la raison, et le type d'exception levée.

```
numéro de la ligne qui génère l'exception:

nom de l'exception:

pourquoi?:
```

-Q2.2- Que sera affiché en sortie, suite à l'exécution de la classe IExo2, en donnant comme valeurs d'entrée, les deux mots: `Bonjours Hello`, comme suit:

```
prompt> java IExo2 Bonjours Hello
```

S'il y a une erreur à l'exécution, précisez ce qui suit:

numéro de la ligne qui génère l'exception:

nom de l'exception:

pourquoi?:

-Q2.3- Que sera affiché en sortie, suite à l'exécution de la classe IExo2, en donnant comme valeurs d'entrée, 5, comme suit:

```
prompt> java IExo2 5
```

S'il y a une erreur à l'exécution, précisez ce qui suit:

numéro de la ligne qui génère l'exception:

nom de l'exception:

pourquoi?:

-Q2.4- Modifier la classe IExo2 afin de capturer les exceptions de Type NumberFormatException pour que le programme ne finisse pas de façon anormale. En cas de mauvais typage, afficher à l'écran le message: "*Cette donnée ne peut pas être convertie en int*"

-Q2.5- Modifier la classe IExo2 pour tenir compte de l'exception générée lors de l'omission d'un argument sur la ligne de commande.

Question 3 (20 points)

Soit le programme suivant:

```
1   public class IExo3 {
2       private ThreadHaut haut;
3       private ThreadBas bas;
4
5       public void start() {
6          haut = new ThreadHaut();
7          haut.start();
8          bas = new ThreadBas();
9          bas.start();
10      }
11      public static void main( String args[] ) {
12         IExo3 app = new IExo3();
13         app.start();
14         System.exit(0);
15      }
16  }
17
18  class ThreadHaut extends Thread {
19
20      public ThreadHaut() {}
21
22      public void run() {
23             System.out.println( "Thread de haute priorité!!!\n" );
24      }
25  }
26
27  class ThreadBas extends Thread {
28
29      public ThreadBas() {}
30
31      public void run() {
32         System.out.println( "Thread de faible priorité!!!\n" );
33      }
34  }
```

-Q3.1- Expliquez brièvement ce que fait ce programme.

-Q3.2- Il y a des situations dans lesquelles l'exécution de la classe IExo3 (i.e. java IExo3) ne produit aucun affichage en sortie malgré la présence des instructions *System.out.println* dans le programme! Expliquez dans quel cas de figure ce phénomène peut se produire.

-Q3.3- Écrire l'instruction (une seule pour chaque cas) nécessaire qui permet au:

-a- ThreadHaut (i.e. haut) de s'exécuter en haute priorité

-b- ThreadBas (i.e. bas) de s'exécuter en basse priorité

-Q3.4- On suppose que les modifications apportées dans la question -3- ont été introduites dans les constructeurs des classes (-a- dans ThreadHaut, -b- dans ThreadBas). L'exécution de la classe IExo3 (en tenant donc compte des modifications), a produit en sortie l'affichage suivant:

```
-----------
Thread de haute priorité!!!

-----------
```

-a- Expliquez ce qu'il est arrivé au ThreadBas?

-b- Quelle instruction (une seule) faudra-t-il ajouter dans la méthode main, pour permettre l'affichage suivant:

```
-----------
Thread de haute priorité!!!

Thread de faible priorité!!!

-----------
```

Réponse:

```
dans la méthode main entre la ligne numéro ______ et numéro ________

instruction:
```

Question 4 (18 points)

```
1   class Province extends Region {
2       protected String message;
3       Province() {
4         message = "est à voir.";
5       }
6       public void affiche(String mess) {
7         System.out.println("Province "+ mess);
8       }
9   }
10  class Region extends Pays {
11      protected String message;
12      Region() {
13        message = "est à localiser.";
14      }
15      public void affiche(String mess) {
16        System.out.println("Region "+ mess);
17      }
18  }
19  class Pays extends Object {
20      protected String message;
21      Pays() {
22        message = "est à découvrir.";
23      }
24      public void affiche(String mess) {
25        System.out.println("Pays " + mess);
26      }
27  }
28
29  public class Quebec extends Province {
30      public String message;
31      Quebec() {
32        message = "est à visiter.";
33      }
34      public void affiche(String mess) {
35        System.out.println("Quebec "+ mess);
36      }
37      public static void main(String[] args) {
38        Region mid = new Province();
39        Province va = new Quebec();
40        Object obj = new Pays();
41        Pays canada = new Region();
42        va.affiche(va.message);
43        mid.affiche(mid.message);
44        ((Pays) obj).affiche(((Pays)obj).message);
45        obj = va;
46        ((Quebec) obj).affiche(((Quebec)obj).message);
47        obj = canada;
48        ((Pays) obj).affiche(((Pays)obj).message);
49        canada = va;
50        ((Pays) canada).affiche(((Pays)canada).message);
51      }
52  }
```

Question: Que vont afficher l'exécution des lignes suivantes (du précédant programme):

```
42          va.affiche(va.message);

43          mid.affiche(mid.message);

44          ((Pays) obj).affiche(((Pays)obj).message);

46          ((Quebec) obj).affiche(((Quebec)obj).message);

48          ((Pays) obj).affiche(((Pays)obj).message);

50          ((Pays) canada).affiche(((Pays)canada).message);
```

Question 5 (30 points)

première partie

Écrire un programme qui réalise ce qui suit:

- accepte sur la ligne de commande, une suite de chaînes de caractères et qui stocke dans un ArrayList (a) celles qui contiennent au moins une fois le caractère 'a'.

- affiche à l'écran:

    - Le nombre de chaînes lues sur la ligne de commande,

    - Le nombre d'éléments dans l'ArrayList (a),

    - Toutes les chaînes (ainsi) stockées dans la structure ArrayList, en utilisant pour cela :

        - une boucle sur la taille de la liste (méthode size()),

        - puis en utilisant un itérateur (méthode iterator() se trouvant dans l'interface Collection)

deuxième partie

En utilisant cette fois-ci une LinkedList à la place d'un ArrayList, faites en sorte que les chaînes ajoutées dans la LinkedList le soient dans l'ordre lexicographique (pas besoin de réécrire tout le code, ajoutez juste les modifications nécessaires dans le programme de la première partie).