UNIVERSITÉ LE HAVRE

UFR
Sciences et
Techniques

Licence 2

Java

19 mars 2019
Durée : 2 heures
Documents, téléphones et calculatrices non autorisés.
Il est possible d'utiliser les réponses à une question non traitée pour résoudre les autres questions.

I *Exceptions*

I.1 Précisez la sortie de ce programme et donnez une explication.

```
class Main {
   public static void main(String args[]) {
      try {
         throw 10;
      }
      catch(int e) {
         System.out.println("Houla exception " + e);
      }
  }
}
```

1. Houla exception 10 ;
2. Houla exception 0 ;
3. Erreur de compilation.

I.2 Expliquez le fonctionnement du programme suivant :

```
class Test extends Exception { }

class Main {
   public static void main(String args[]) {
      try {
         throw new Test();
      }
      catch(Test t) {
         System.out.println("Exception capturee");
      }
      finally {
         System.out.println("Dans le bloc finally ");
      }
  }
}
```

25 Rue Philippe Lebon

B.P. 540

76058 Le Havre Cedex

I.3 Si ce programme compile expliquez son fonctionnement, sinon précisez pourquoi il ne compile pas.

```
class Base extends Exception {}
class Derivee extends Base  {}

public class Main {
  public static void main(String args[]) {
   // Des instructions
   try {
       // D'autres instructions
       throw new Derivee();
    }
    catch(Base b)     {
       System.out.println("Exception de base capturee");
    }
    catch(Derivee d)  {
       System.out.println("Exception derivee capturee");
    }
  }
}
```

I.4 Quelle est la sortie de ce programme ?

```
class Test
{
    String str = "a";

    void A()
    {
        try
        {
            str +="b";
            B();
        }
        catch (Exception e)
        {
            str += "c";
        }
    }

    void B() throws Exception
    {
        try
        {
            str += "d";
            C();
        }
        catch(Exception e)
        {
            throw new Exception();
        }
        finally
        {
            str += "e";
        }
        str += "f";
    }

    void C() throws Exception
    {
        throw new Exception();
    }

    void affiche()
    {
        System.out.println(str);
    }

    public static void main(String[] args)
    {
        Test object = new Test();
        object.A();
        object.affiche();
    }
}
```

I.5 Quelle est la sortie de ce programme ?

```
class Test
{   int count = 0;

    void A() throws Exception
    {
        try
        {
            count++;

            try
            {
                count++;

                try
                {
                    count++;
                    throw new Exception();

                }

                catch(Exception ex)
                {
                    count++;
                    throw new Exception();
                }
            }

            catch(Exception ex)
            {
                count++;
            }
        }

        catch(Exception ex)
        {
            count++;
        }

    }
```

```
    void affiche()
    {
        System.out.println(count);
    }

    public static void main(String[] args) throws Exception
    {
        Test obj = new Test();
        obj.A();
        obj.affiche();
    }
}
```

II *Zéro d'une fonction - méthode de Newton*

On cherche à écrire les classes nécessaires pour déterminer une bonne approximation de la valeur de x tel que $f(x) = 0$, sachant que f est une fonction d'une variable réelle et dérivable. Pour cela on utilise l'algorithme de Newton. Tout d'abord, on part d'un point x_0 proche du zéro à trouver. Ceci est généralement possible en faisant des estimations grossières (ce n'est pas demandé ici). Partant de ce point, on construit par récurrence la suite :

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)},$$

et l'on arrête lorsque $f(x_k) = 0$ ou lorsque $f'(x_k) = 0$. Pour la mise en œuvre, on prendra comme critère d'arrêt : $\frac{f(x_n)}{f'(x_n)} < 10^{-p}$, p correspondant à la précision, c'est un paramètre fourni en entrée. De plus, on lèvera une exception si la dérivée est nulle.

II.1 Écrire une classe `ExceptionDeriveeNulle`.

II.2 Écrire les classes nécessaires (au sens large) pour rechercher le zéro d'une fonction qui sera passée en paramètre.

II.3 L'exemple choisi par Newton était $f(x) = x^3 - 2x - 5$ et il a recherché le zéro entre 2 et 3. Écrire une classe "principale" recherchant le zéro de cette fonction à l'aide des classes que vous avez écrites.

III *Polymorphisme*

Écrire les classes nécessaires au fonctionnement du programme suivant (en ne fournissant que les méthodes nécessaires à ce fonctionnement).

```
public class TestMetiers {
  public static void main(String[] argv) {
     Personne[ ] personnes = new Personne[3] ;
     personnes [ 0 ] = new Menuisier( "Paul" ) ;
     personnes [ 1 ] = new Plombier( "Jean" ) ;
    personnes [ 2 ] = new Menuisier( "Adrien" ) ;
    for( int i=0 ; i< personnes.length ; i++ ) personnes[i].affiche() ;
  }
}
```

Sortie de ce programme :

```
Je suis Paul le Menuisier
Je suis Jean le Plombier
Je suis Adrien le Menuisier
```