Question	Points	Score
1	10	
2	10	
3	10	
4	20	
5	20	
6	30	
Total:	100	

DIRECTIVES PÉDAGOGIQUES: - Documentation n'est pas permise.
- Echanger des informations lors d'un examen (ou autres formes de tricherie) est du **plagiat**, qui est passible de sanctions allant jusqu'au l'exclusion du programme.

---

(10) 1. Considérez la programme suivante:

```
class Casting {
    public static void main(String [] args) {
        Enfant e = new Enfant();
        System.out.println("A " + e.getVal());
        System.out.println("B " + ((Parent)e).getVal());
        System.out.println("C " + ((Parent)e).getVal2());
        System.out.println("D " + e.val);
        System.out.println("E " + ((Parent)e).val);
    }
}
class Parent {
    int val = 10;
    int getVal()  { return val; }
    int getVal2() { return val; }
}
class Enfant extends Parent {
    int val = 20;
    int getVal() { return val; }
}
```

Écrivez le résultat de ce programme (après avoir taper `java Casting`)

Solution:

```
A 20
B 20
C 10
D 20
E 10
```

(10) 2. Considérez le code suivant:

```
class Person {
    String nom;
    int age;
    Person (String nom, int age) {
        this.nom = nom;
        this.age=age;
    }

    public String toString() {
        return "Nom=" + nom + " age=" + age;
    }

    public static void main(String[] args) {
        Person p1 = new Person("Carla Bruni",40);
        Person p2 = new Person("Carla Bruni",40);
        Object o1 = (Object)p1;
        System.out.println("p1.equals(p2)?" + p1.equals(p2));
        System.out.println("p1.equals(o1)?" + p1.equals(o1));
        System.out.println("p2.equals(o1)?" + p2.equals(o1));
    }
}
```

(a) Écrivez le résultat de ce programme (après avoir taper `java Person`)

(b) Modifiez la classe `Person` pour que deux instances de `Person` soient égales si et seulement s'ils partagent les même nom et âge. Vous n'avez pas besoin de recopier la classe. Il suffit d'écrire le code pour les méthodes `public int hashCode()` et `public boolean equals(Object o)`.

Solution:

```
p1.equals(p2)?false
p1.equals(o1)?true
p2.equals(o1)?false

_____-
```

```
class  Person2{
    String nom;
    int age;
    Person2 (String nom, int age) {
        this.nom = nom;
        this.age=age;
    }
    public boolean equals(Object o) {
        if (!(o instanceof Person2)) {
            return false;
        }
        Person2 p = (Person2) o;
        return p.nom.equals(nom) && p.age==age;
    }
    public int hashCode() {
        return nom.hashCode() + age;
    }
    public String toString() {
        return "Nom=" + nom + "␣age=" + age;
    }
}
```

(10) 3. Considérez le code partiel suivant:

```
class MonPanel extends JPanel {
    private int numClicks=0;
    JButton b;
    public MonPanel() {
        b=new JButton("Cliquez-moi");
        b.addMouseListener((MouseListener) new MyMouseListener()));
        add(b);

    }

    //A implementer MyMouseListener comme classe interieur

}

interface MouseListener {
  void mouseClicked(MouseEvent e);
  void mouseEntered(MouseEvent e);
  void mouseExited(MouseEvent e);
  void mousePressed(MouseEvent e);
  void mouseReleased(MouseEvent e);
}
```

(a) Implémentez le code pour la partie `A`, une "inner" `class MyMouseListener`. L'interface `MouseListener` est indiquée en haut. Votre code pour `MyMouseListener` doit incrémenter `numClicks` chaque fois que `void mouseClicked(MouseEvent e)` est appellée. Faites attention á bien définir `MyMouseListener` comme une classe intérieur ("inner class") et pour qu'elle implémente `MouseListener`.

Solution:

```
class MouseTester extends JPanel {
    private int numClicks=0;
    JButton b;
    JLabel l;
    public MouseTester() {
        super(new BorderLayout());
        MouseAdapter m = new MyMouseListener();
        addMouseListener(m);
        b=new JButton("Cliquez-moi");
        b.addMouseListener(m);
        l=new JLabel("Clicks = 0");
        l.setLabelFor(b);
        add(b, BorderLayout.CENTER);
        add(l, BorderLayout.NORTH);
    }

    //class MyMouseListener extends MouseAdapter {
    //  public void mouseClicked(MouseEvent e) {
    //      if (e.getSource()==b) {
    //          l.setText("Clicks =" + ++numClicks);
    //      }
    //  }
    //}
    class MyMouseListener implements MouseListener {
        public void mouseClicked(MouseEvent e) {
            if (e.getSource()==b) {
                l.setText("Clicks =" + ++numClicks);
            }
        }
        void mouseEntered(MouseEvent e) {}
        void mouseExited(MouseEvent e) {}
        void mousePressed(MouseEvent e) {}
        void mouseReleased(MouseEvent e) {}
    }

}
```

(20) 4. Pour cette question, vous devez écrire une classe qui s'appelle Intra. Suivez les instructions avec soins.

(a) La classe doit posséder comme variable un `ArrayList` qui stock des éléments de type `Integer`.

(b) Dans le constructeur, créer l' `ArrayList` et le remplir avec 10 `Integer`s aléatoires. Utilisez la méthode `nextInt()` de `java.util.Random` pour obtenir une valeur de type primitif `int`. Utilisez le constructeur `Integer(int)` de `java.lang.Integer` pour créer le `Integer` á partir du type primitif `int`.

(c) Écrire la méthode `public String toString()` qui imprime les entiers en utilisant un `Iterator` obtenu de l' `ArrayList`. Cette méthode *doit utiliser un Iterator*.

(d) Écrire la méthode `public int recursiveSum()` qui utilise la récursion pour calculer et imprimer la somme des entiers contenus dans le tableau. Cette méthode *doit être récursive*.

(e) Écrire la méthode `public void sortArray()` qui tri les valeurs du tableau de la plus grande á la plus petite. Votre tri n'a pas besoin d'être efficace mais il doit fonctionner. Cette méthode *doit faire le tri elle-même.* Par exemple, vous n'avez pas le droit d'utiliser `java.util.Collections.sort(List)`.

Solution:

```
class Intra {
    ArrayList<Integer> a;
    Random rand;
    Intra() {
        rand = new Random();
        a = new ArrayList<Integer>();
        for (int i=0;i<10;i++) {
            a.add(rand.nextInt());
        }
    }

    public String toString() {
        Iterator it = a.iterator();
        StringBuffer b = new StringBuffer();
        while (it.hasNext()) {
            b.append(" " + it.next());
        }
        return b.toString();
    }

    public int recursiveSum() {
        return recSub(0);
    }

    public int recSub(int idx) {
        if (idx>=a.size()) {
            return 0;
        } else {
            return a.get(idx) + recSub(++idx);
        }
    }

    public void sortArray() {
        //primitive bubble sort
        boolean modified=true;
        while (modified) {
            modified=false;
            for (int i=0;i<a.size()-1;i++) {
                if (a.get(i)>a.get(i+1)) {
                    Integer tmp=a.get(i);
                    a.set(i,a.get(i+1));
                    a.set(i+1,tmp);
                    modified=true;
                }
```

```
            }
        }
    }

    public static void main(String [] args) {
        Intra intra = new Intra();
        System.out.println("Unsorted " + intra);
        intra.sortArray();
        System.out.println("Sorted   " + intra);
        System.out.println("Sum is " + intra.recursiveSum());
    }
}
```

(20) 5. Considérez la classe partielle `class GPSLocation` qui stocke les coordonnés pour un point sur la terre avec sa latitude et sa longitude (en radians)

```
class GPSLocation {
    String loc;  //location
    double lat;  //latitutde
    double lng;  //longitude
    GPSLocation(String loc, double lat, double lng) {
        this.loc=loc;
        this.lat=lat;
        this.lng=lng;
    }
    GPSLocation(String rec) {
        //reads in values such as: Quebec 46 50 N 71 15 W
        StringTokenizer st = new StringTokenizer(rec);
        loc = st.nextToken();
        lat = Double.parseDouble(st.nextToken());
        lat += Double.parseDouble(st.nextToken())/60.0;
        if (st.nextToken().equals("S")) lat = -lat;
        lng = Double.parseDouble(st.nextToken());
        lng += Double.parseDouble(st.nextToken())/60.0;
        if (st.nextToken().equals("W")) lng = -lng;
        lng *= Math.PI/180.0;
        lat *= Math.PI/180.0;
    }

    public String toString() {
        return loc + " lat=" + lat + " lng=" + lng;
    }

    double distance(GPSLocation desired) {
      //A Calculer distance
    }

    public static void main (String [] args) {
      ArrayList<GPSLocation> pts = new ArrayList<GPSLocation>();
      //B Lire ficher
   }
}
```

Voir page suivante

(a) Ecrivez le code pour la partie `A`: calculez la distance entre deux pointes en km en utilisant la formule suivante:

$$a = \sin((\mathbf{lat}_1 - \mathbf{lat}_2)/2)^2 + \cos(\mathbf{lat}_1) * \cos(\mathbf{lat}_2) * \sin((\mathbf{lng}_1 - \mathbf{lng}_2)/2)^2 \quad (1)$$
$$c = 2 * \text{atan2}(\sqrt{a}, \sqrt{1-a}) \quad (2)$$
$$k = c * 6371 \quad (3)$$

Ou 6371 est la circonférence de la Terre en km et k est donc la distance entre les points en km. Vous pouvez utiliser les fonctions: `Math.sin(double x)`, `Math.cos(double x)`, `Math.sqrt(double x)`, `Math.atan2(double x, double y)`

(b) Écrivez le code pour la partie `B`: ouvrez le fichier texte avec le nom fourni dans `args[1]`. Pour chaque ligne du fichier, créez une nouvelle instance de `GPSLocation` en utilisant le constructeur approprié et ajoutez le `GPSLocation` au `ArrayList<GPSLocation> pts`. Le fichier contient des lignes telles que :

```
Guadalajara 20 40 N 103 20 W
Montreal 45 28 N 73 45 W
Vancouver 49 16 N 123 07 W
Winnipeg 49 53 N 97 09 W
```

Utilisez `new BufferedReader(new FileReader (String fName))` pour lire le fichier. `BufferedReader` offre une méthode `readLine()` qui retourne la prochaine ligne du fichier dans un `String` jusqu'á ce que le fichier soit vide. Puis elle retourne `null`. `FileReader` lance une `IOException` qui doit être attrapée. Il suffit d'imprimer l'exception quand vous l'attrappez.

Solution:

```
import java.io.*;
import java.util.*;
class GPS {
    public static void main(String [] args) {
        ArrayList<GPSLocation> pts = new ArrayList<GPSLocation>();
        try {
            BufferedReader f =
            new BufferedReader(new FileReader(args[0]));
            String l = f.readLine();
            while (!(l==null)) {
                System.out.println(l);
                pts.add(new GPSLocation(l));
                l=f.readLine();
            }
        } catch (IOException e) {
            System.out.println(e);
```

```
        }
        for (GPSLocation i : pts) {
           for (GPSLocation j : pts) {
                if (!j.equals(i)) {
                    System.out.println(i.loc + " to " +
                    j.loc + " " + (int)i.distance(j) + "kms");
                }
           }
        }
    }
}


class GPSLocation {
    String loc;  //location
    double lat;  //latitutde
    double lng; //longitude
    GPSLocation(String loc, double lat, double lng) {
        this.loc=loc;
        this.lat=lat;
        this.lng=lng;
    }

    GPSLocation(String rec) {
        //reads in values such as
        //Quebec                46 50 N         71 15 W
        StringTokenizer st = new StringTokenizer(rec);
        loc = st.nextToken();
        lat = Double.parseDouble(st.nextToken());
        lat += Double.parseDouble(st.nextToken())/60.0;
        if (st.nextToken().equals("S")) lat = -lat;
        lng = Double.parseDouble(st.nextToken());
        lng += Double.parseDouble(st.nextToken())/60.0;
        if (st.nextToken().equals("W")) lng = -lng;
        lng *= Math.PI/180.0;
        lat *= Math.PI/180.0;
    }

    public String toString() {
        return loc + " lat=" + lat + " lng=" + lng;
    }

    double distance(GPSLocation desired) {
        double dlon = desired.lng - lng;
        double dlat = desired.lat - lat;
```

```
        double a = Math.pow(Math.sin(dlat/2),2) + Math.cos(lat) *
        Math.cos(desired.lat) *  Math.pow(Math.sin(dlon/2),2);
        double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
        return 6371 * c;  //6371 is the radius of the Earth in kms
    }
}
```

(30) 6. Créez les trois classes suivantes:

(a) `class Item`: Un item a une vidéothèque (video rental store). Cette classe est *abstraite*.

Variables:

```
String name                // le nom de l'item
int id                     // commence a 1 et incremente
                           // automatiquement avec chaque nouvel Item
                           // (Vous pouvez utiliser une autre variable
                           // pour tenir l'id du derniere Item cree)
```

Constructeurs:

```
Item(String name);
```

Méthodes:

```
public String toString()   // imprime le nom et l'id
double cost()              // methode abstraite qui retourne le cout
                           // de l'item
```

(b) `class DVD` (sous-classe d'Item): Un DVD

Variables:

```
double dvdCost=4.50;       // le cout du DVD
```

Constructeurs:

```
DVD(String name);
```

Méthodes:

```
double cost()              // retourne dvdCost
public String toString()   // imprime le cout et puis
                           // imprime le nom and id via Item.toString()
```

(c) `class Game` (sous-classe d'Item) : Un jeux video

Variables:

```
double gCost=8.00;         // le cout du base d'un jeux
String platform;           // la platte-forme de jeux
```

Constructeurs:

```
Game(String name, String platform);
```

Méthodes:

```
double cost()              // si la plateforme est ``XBOX'' retourne
                           // gCost*1.5, sinon retourne gCost
public String toString()   // imprime le cout et la plateforme et
                           // le nom et id via Item.toString()
```

Solution:

```
import java.util.*;
class DVDGameItem {
    public static void main (String [] args) {
        Game g1 = new Game("Assassins Creed","XBOX");
        Game g2 = new Game("Assassins Creed","PS3");
        DVD d1 = new DVD("Juno");
        ArrayList<Item> a = new ArrayList<Item>();
        a.add(g1);
        a.add(g2);
        a.add(d1);
        for (Item item : a) {
            System.out.println(item);
        }
    }
}
abstract class Item {
    protected String nom;
    protected int id;
    private static int ctr=1;
    Item (String nom) {
        this.nom=nom;
        this.id=ctr++;
    }
    public String toString() {
        return "name=" + nom + " id=" + id;
    }
    public abstract double cost();
}
class DVD extends Item {
    double dvdCost = 4.50;
    DVD(String name) {
        super(name);
    }
    public String toString() {
        return "DVD cost=" + cost() + " " + super.toString();
    }
    public double cost() {
        return dvdCost;
    }
}
class Game extends Item {
    double gCost = 8.0;
    String platform;
    Game(String name, String platform) {
```

```
        super(name);
        this.platform=platform;
    }
    public String toString() {
        return "Game cost=" + cost() + " platform=" +
        platform + " " + super.toString();
    }
    public double cost() {
        if (platform.equals("XBOX")) {
            return gCost * 1.5;
        } else {
            return gCost;
        }
    }
}
```