|  |  |
| --- | --- |
| **Nom :**Pas le nom ou pas respecté la consigne -2 | **Prénom :** |

## **2 heures**

## **Tourcoing le 16 mars 2018**

Vous renommerez ce fichier : Examen2\_java\_prenom\_nom.odt

que vous enverrez complété dans la fenêtre prévue à cet effet.

## **Item 4 : Maîtrise de l’EDI**

A l’aide du logiciel de votre choix, testez ce programme et apportez la preuve de son fonctionnement en fournissant une copie d’écran

**import** **java.util.\***;

**public** **class** **CompareTableaux** {

 **public** **static** void main(String[] args) {

 HashSet<String> hs1 = **new** HashSet<String>(Arrays.asList(**new** String[] { "2", "3", "5", "7", "11" }));

 HashSet<String> hs2 = **new** HashSet<String>(Arrays.asList(**new** String[] { "2", "4", "6", "8", "11", "12" }));

 HashSet<String> similarites = **new** HashSet<String>(hs1);

 HashSet<String> differences = **new** HashSet<String>(hs1);

 similarites.retainAll(hs2);

 hs2.removeAll(similarites);

 differences.removeAll(similarites);

 differences.addAll(hs2);

 System.out.println( similarites.toString() ); *//[11, 2]*

 System.out.println( differences.toString() ); *//[12, 3, 4, 5, 6, 7, 8]*

 }

}

Mettez ici une copie d’écran du résultat de votre travail

|  |
| --- |
| /3  |

A l’aide du logiciel de votre choix, testez ce package et apportez la preuve de son fonctionnement en fournissant une copie d’écran

**import** javax.swing.JFrame;

**public** **class** Fenetre **extends** JFrame

{

**public** Fenetre()

 {

 **this**.setTitle("Un rond");

 **this**.setSize(300, 300);

 **this**.setLocationRelativeTo(**null**);

 **this**.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

 **this**.setContentPane(**new** Panneau());

 **this**.setVisible(**true**);

 }

}

**import** java.awt.Graphics;

**import** javax.swing.JPanel;

**public** **class** Panneau **extends** JPanel

 {

 **public** **void** paintComponent(Graphics g){

 **int** x1 = **this**.getWidth()/8;

 **int** y1 = **this**.getHeight()/8;

 g.fillOval(x1, y1, **this**.getWidth()/4, **this**.getHeight()/4);

 x1 = **this**.getWidth()\*5/8;

 g.fillOval(x1, y1, **this**.getWidth()/4, **this**.getHeight()/4);

 x1 = **this**.getWidth()\*7/16;

 y1 = **this**.getHeight()\*8/16;

 g.fillOval(x1, y1, **this**.getWidth()/8, **this**.getHeight()/8);

 x1 = **this**.getWidth()/4;

 y1 = **this**.getHeight()/4;

 g.drawOval(x1, y1, **this**.getWidth()/2, **this**.getHeight()/2);

 }

}

**import** javax.swing.JFrame;

**public** **class** Test

 {

 **public** **static** **void** main(String[] args)

 {

 Fenetre fen = **new** Fenetre();

 }

}

|  |
| --- |
|  /6 |

## **Item 5 : Testez ce programme**

**public** **class** Overflow {

 **public** **static** **void** main(String[] args) {

 **int** grand = 0x7fffffff; // valeur maxi d’un int

 *prt*("grand = " + grand);

 **int** plusgrand = grand \* 4;

 *prt*("plusgrand = " + plusgrand);

 }

 **static** **void** prt(String s) {

 System.***out***.println(s);

 }

}

Mettez ici une copie d’écran du résultat de votre travail

|  |
| --- |
| /3 |

Expliquez le résultat !

|  |
| --- |
| 0x7fffffff est le plus grand nombre entier positif codable en hexadécimal ce qui fait 2147483647 en décimalSi je multiplie par 4 (100 en binaire) je décale tous les « 1 » de deux vers la gauche, le bit de poids fort devient donc un « 1 » ce qui en fait un nombre négatif0111 1111 1111 1111 1111 1111 1111 1111X 0100= 1 1111 1111 1111 1111 1111 1111 1111 11001111 1111 1111 1111 1111 1111 1111 1100 vaut -4Le « 1 » à gauche est perdu, c’est pourquoi cette classe a été nommée « Overflow)./4 |

## **Item 6 : Interface graphique**

**Créer une fenêtre comportant un bouton marqué « π » et une zone d’affichage…**

**En cliquant sur le bouton, faire apparaître le texte « 3,14159 »**

π

3 ,14159

Mettez ici une copie d’écran du résultat de votre travail

|  |
| --- |
|  /4 |