

MOOC Init Prog Java

Corriges semaine 3

Les corrigés proposés correspondent à l'ordre des apprentissages : chaque corrigé correspond à la solution à laquelle vous pourriez aboutir au moyen des connaissances acquises jusqu'à la semaine correspondante.

Exercice 9: Tables de multiplications (boucles for)

```
class Tables {  
  
    public static void main(String[] args) {  
        System.out.println("    Tables de multiplication");  
        for (int i = 2; i <= 10; ++i) {  
            // \n est un saut de ligne  
            System.out.println("\n Table de " + i + " :");  
            for (int j=0; j <= 10; ++j) {  
                System.out.println("    " + i + " * " + j + " = " + i*j);  
            }  
        }  
    }  
}
```

Exercice 10: Plus Grand Diviseur Commun (Algorithme, if, boucles)

```
import java.util.Scanner;

class PGDC {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.println("Calcul du plus grand diviseur commun de deux nombres");
        // Entrée des données
        System.out.print("Entrez un nombre positif : ");
        int nb1 = scanner.nextInt();
        System.out.print("Entrez un nombre positif : ");
        int nb2 = scanner.nextInt();

        /*
         * A chaque passage de la boucle while, on modifie le plus grand
         * de a et b en déduisant le nombre plus petit, comme indiqué par
         * la formule d'Euclide. La boucle se terminera quand a et b sont
         * égaux (au pire des cas quand ils valent 1). A ce moment-là, on
         * retourne la valeur de a (on aurait aussi pu retourner b).
         */
        int a = nb1;
        int b = nb2;

        while (a != b) {
            if (a > b) {
                a = a - b;
            } else {
                b = b - a;
            }
        }

        System.out.println("Le plus grand diviseur commun de " + nb1 + " et " + nb2 + " est " + a);

        scanner.close();
    }
}
```

Il existe une solution dite *récursive* du calcul du PGDC un peu plus intuitive à écrire.

Exercice 11: Rebonds de balles (for, do while)

```
import java.util.Scanner;

public class Rebonds1 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        //Constante de gravité
        final double G = 9.81;

        // Déclarations
        // vitesses avant et après le rebond
        double v = 0.0;
        double v1 = 0.0;
        // hauteur avant le rebond, hauteur de remontée
        double h = 0.0;
        double h1 = 0.0;
        // hauteur initiale
        double H0 = 0.0;
        // valeur epsilon
        double eps = 0.0;
        // nombre de rebonds
        int NBR = 0;

        /*
         * Entrée des valeurs par l'utilisateur,
         * avec test de validité
         */

        do {
            System.out.print("Coefficient de rebond (0 <= coeff < 1) : ");
            eps = scanner.nextDouble();
        } while ((eps < 0.0) || (eps >= 1.0));
        do {
            System.out.print("Hauteur initiale (0 <= H0) : ");
            H0 = scanner.nextDouble();
        } while (H0 < 0.0);
        do {
            System.out.print("Nombre de rebonds (0 <= N ) : ");
            NBR = scanner.nextInt();
        } while (NBR < 0);

        // Boucle de calcul
        h = H0;
        // on fait une itération par rebond
        for (int nombre = 0; nombre < NBR; ++nombre) {
            v = Math.sqrt(2.0 * G * h);
            // vitesse après le rebond
            v1 = eps * v;
            // la hauteur à laquelle elle remonte...
            h1 = (v1 * v1) / (2.0 * G);
            // ...qui devient la nouvelle hauteur initiale
            h = h1;
            System.out.println("rebond " + (nombre+1) + " : " + h);
        }
    }
}
```

```

        /* Affichage du résultat */
        System.out.println("Au " + NBR + "eme rebond, la hauteur sera de " + h);

        scanner.close();
    }
}

import java.util.Scanner;

public class Rebonds2 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        //Constante de gravité
        final double G = 9.81;
        // Déclarations
        // vitesses avant et après le rebond
        double v = 0.0;
        double v1 = 0.0;
        // hauteur avant le rebond, hauteur de remontée
        double h = 0.0;
        double h1 = 0.0;
        double h0 = 0.0;
        double eps = 0.0;
        double h_fin = 0.0;

        /*
         * Entrée des valeurs par l'utilisateur,
         * avec test de validité
         */
        do {
            System.out.print("Coefficient de rebond (0 <= coeff < 1) : ");
            eps = scanner.nextDouble();
        } while ((eps < 0.0) || (eps >= 1.0));
        do {
            System.out.print("Hauteur initiale (0 <= H0) : ");
            h0 = scanner.nextDouble();
        } while (h0 < 0.0);
        do {
            System.out.println("Hauteur finale (0 <= h_fin) : ");
            h_fin = scanner.nextDouble();
        } while (h_fin < 0.0 || h_fin > h0);

        // Boucle de calcul
        h = h0;
        int count = 0;
        do {
            v = Math.sqrt(2.0 * G * h);
            // vitesse après le rebond
            v1 = eps * v;
            // incrémente le nombre de rebonds
            count++;
            // la hauteur à laquelle elle remonte...
            h1 = (v1 * v1) / (2.0 * G);
            // ...qui devient la nouvelle hauteur initiale

```

```
        h  = h1;

        System.out.println("rebond " + count + " : " + h );
    } while (h1 > h_fin);

    // Affichage du résultat
    System.out.println("Nombre de rebonds :" + count);

    scanner.close();
}
}
```
