

Structures de base [bs] - Sujets d'examens

Karine Zampieri, Stéphane Rivière

Unisciel  algoprogram  Version 14 mai 2018

Table des matières

1	Les ardoises d'un toit / pardoises (4 points)	2
2	Force d'attraction / pattraction (3 points)	4
3	Troupeau vaches-moutons / ptroupeau (4 points)	6
4	Références générales	7

Java - Sujets d'examens (Solution)

1 Les ardoises d’un toit / pardoises (4 points)



Objectif

Calculer le nombre de paquets d’ardoises pour couvrir un toit constitué de deux rectangles identiques. Les ardoises sont vendues par paquet de 12.

Voici un exemple du résultat attendu :

```
Hauteur et largeur d’un versant du toit? 3.5 10
Et celles d’une ardoise? 0.20 0.5
==> Nombre de paquets : 60
```



(0.5 point) Définissez les constantes entières :

- `NRECTANGLES=2` (nombre de rectangles).
- `NARDPAQUET=12` (nombre d’ardoises par paquet).



(1 point) Écrivez un programme qui saisit :

- Les dimensions d’un versant de toit dans `httoit` et `lrtoit`.
- Celles d’une ardoise dans `htard` et `lrard`.

Toutes les variables sont réelles.

Affichez les invites :

```
Hauteur et largeur d’un versant du toit?
Et celles d’une ardoise?
```



(1 point) Pour un versant, calculez l’arrondi supérieur du nombre d’ardoises :

- En hauteur dans `nardht` (entier) : $nardht = \lceil httoit/htard \rceil$
- En largeur dans `nardlr` (entier) : $nardlr = \lceil lrtoit/lrard \rceil$

($\lceil x \rceil$ désigne l’entier supérieur du réel x)

Outil Java

L’opération $\lceil x \rceil$ s’écrit `Math.ceil(x)`.



(0.5 point) Dans l’entier `nardoises`, calculez le nombre total d’ardoises :

$$nardoises = nardht \times nardlr \times NRECTANGLES$$



(0.5 point) Dans l’entier `npaquets`, calculez le nombre de paquets d’ardoises.

Attention, ici aussi il n’y a pas de demi-paquet d’où l’arrondi supérieur :

$$npaquets = \lceil nardoises/NARDPAQUET \rceil$$



(0.5 point) Affichez (où [x] désigne le contenu de x) :

```
==> Nombre de paquets : [npaquets]
```



Testez.



Validez votre programme avec la solution.

Solution Java @[pgardoises1.java]

```
import java.util.Scanner;

class PGArdoises1 {

    final static int NRECTANGLES = 2;
    final static int NARDPAQUET = 12;

    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        double httoit, lrtoit;
        System.out.print("Hauteur et largeur d'un versant du toit? ");
        httoit = input.nextDouble();
        lrtoit = input.nextDouble();
        double htard, lrard;
        System.out.print("Et celles d'une ardoise? ");
        htard = input.nextDouble();
        lrard = input.nextDouble();
        int nardht = (int)Math.ceil(httoit / htard);
        int nardlr = (int)Math.ceil(lrtoit / lrard);
        int nardoises = NRECTANGLES * nardht * nardlr;
        int npaquets = (int)Math.ceil(nardoises / NARDPAQUET);
        System.out.println("Nombre de paquets : " + npaquets);
    }
}
```

2 Force d’attraction / pgattraction (3 points)



Objectif

Déterminer la force d’attraction entre deux corps.

Voici un exemple du résultat attendu :

```
Masses des deux corps? 1.5 2
Distance? 10
Force d’attraction vaut 0.2943
```



(1 point) Écrivez un programme qui saisit :

- La masse de deux corps dans `m1` et `m2`.
- La distance qui les sépare dans `r`.

Toutes les variables sont des réels.

Affichez les invites :

```
Masses des deux corps?
Distance?
```



(0.5 point) Définissez la constante `GTERRE=9.81` (gravitation g).



(1 point) Dans le réel `f`, calculez la force d’attraction définie par :

$$F = g \cdot m_1 \cdot m_2 / r^2$$



(0.5 point) Affichez (où `[x]` désigne le contenu de `x`) :

```
Force d’attraction vaut [f]
```



Testez.



Validez votre programme avec la solution.

Solution Java @[pgattraction1.java]

```
import java.util.Scanner;

class PGAttraction1 {

    final static double GTERRE = 9.81;

    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
```

```
double m1, m2;
System.out.print("Masses des deux corps? ");
m1 = input.nextDouble();
m2 = input.nextDouble();
double r;
System.out.print("Distance? ");
r = input.nextDouble();
double f = GTERRE * m1 * m2 / (r * r);
System.out.println("Force d'attraction vaut " + f);
}
}
```

3 Troupeau vaches-moutons / pgtroupeau (4 points)



Objectif

Déterminer le prix d'un troupeau composé de vaches et de moutons.
Voici un exemple du résultat attendu :

```
Nombre de vaches? 3
Nombre de moutons? 7
Prix des vaches est 750.0 euros
Celui des moutons est 840.0 euros
Total du troupeau est 1590.0 euros
Nombre d'animaux est 10
```



(1 point) Écrivez un programme qui saisit :

- Le nombre de vaches dans `nv` (entier).
- Celui des moutons dans `nm` (entier).

Affichez les invites :

```
Nombre de vaches?
Nombre de moutons?
```



(0.5 point) Définissez les **constantes réelles** :

- `PRIX_VACHE=250.0` (prix d'une vache) en €.
- `PRIX_MOUTON=120.0` (prix d'un mouton) en €.



(0.5 point) Calculez le prix :

- Des vaches dans `prv` (réel).
- Celui des moutons dans `prm` (réel).



(1 point) Affichez (où `[x]` désigne le contenu de `x`) :

```
Prix des vaches est [prv] euros
Celui des moutons est [prm] euros
```



(1 point) Calculez et affichez :

```
Total du troupeau est ... euros
Nombre d'animaux est ...
```



Testez.



Validez votre programme avec la solution.

Solution Java @[pgtroupeau1.java]

```
import java.util.Scanner;

class PGTroupeau1 {

    final static double PRIX_VACHE = 250.0;
    final static double PRIX_MOUTON = 120.0;

    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int nv, nm;
        System.out.print("Nombre de vaches? ");
        nv = input.nextInt();
        System.out.print("Nombre de moutons? ");
        nm = input.nextInt();
        double prv = PRIX_VACHE * nv;
        double prm = PRIX_MOUTON * nm;
        System.out.println("Prix des vaches est " + prv + " euros");
        System.out.println("Celui des moutons est " + prm + " euros");
        System.out.println("Total du troupeau est " + (prv + prm) + " euros");
        System.out.println("Nombre d'animaux est " + (nv + nm));
    }
}
```

4 Références générales

Comprend ■