

Structures de base [bs]

Exercices de cours

Karine Zampieri, Stéphane Rivière

Unisciel  algoprogram  Version 12 mai 2018

Table des matières

1	Appréhender le cours	2
1.1	Simplification de programme / pgsimplific	2
1.2	Réutilisation d'une variable / pgaffected3	5
1.3	Exponentiation 16-ème / pgexponent1	7
2	Appliquer le cours	8
2.1	Quelques calculs / pgcalcul1	8
2.2	Durée de trajet / pgtrajet1	10
2.3	Un peu de trigo / pgtrigo1	12
2.4	Date de Pâques / pgpagues1	13
3	Approfondir le cours	15
3.1	Moyennes de deux entiers / pgmoyenne1	15
3.2	Codage d'une date / pgdate1	16

Java - Exercices de cours (Solution)



Mots-Clés Structures de base ■

Difficulté ●○○ (1 h 30) ■

1 Appréhender le cours

1.1 Simplification de programme / pgsimplific



Soit le programme suivant :

```
import java.util.Scanner;

class PGSimplific1a {

public static void main(String[] args)
{
    Scanner input = new Scanner(System.in);
    a = input.nextInt();
    b = a;
    c = b + b * b;
    x = input.nextInt();
    e = 4 * x;
    s = e + b;
    s = s * s + c;
    f = s;
    System.out.println("Le resultat est " + f);
}
}
```



Déclarez les variables de type entier.



Quel est le résultat avec 3 et -2 ?
Justifiez.

Solution simple

Le résultat est 37. En effet :

- On saisit 3 dans **a**.
- **b** vaut **a** qui vaut 3.
- **c** vaut $3+3*3=12$.
- On saisit -2 dans **x**.
- **e** vaut $4*(-2)=-8$.
- **s** vaut $-8+3=-5$ puis $(-5)*(-5)+12=25+12=37$.
- Finalement **f** vaut **s** qui vaut 37.



Les variables **b** et **f** étant inutiles,
Écrivez une première version simplifiée.



Validez votre programme avec la solution.

Solution Java

```
import java.util.Scanner;

class PGSimplific1b {

public static void main(String[] args)
{
    Scanner input = new Scanner(System.in);
    int a, c, x, e, s;
    a = input.nextInt();
    c = a + a * a;
    x = input.nextInt();
    e = 4 * x;
    s = e + a;
    s = s * s + c;
    System.out.println("Le resultat est " + s);
}
}
```



Quelles sont les variables incompressibles ?

Solution simple

Ce sont les données et les résultats :

- Données : `a` et `x`
- Résultats : `s`



En ne déclarant que les données et les résultats,
Écrivez une deuxième version simplifiée.



Validez votre programme avec la solution.

Solution Java

```
import java.util.Scanner;

class PGSimplific1c {

public static void main(String[] args)
{
    Scanner input = new Scanner(System.in);
    int a, x, s;
    a = input.nextInt();
    x = input.nextInt();
    s = 4 * x + a;
    s = s * s + a + a * a;
    System.out.println("Le resultat est " + s);
}
}
```



Vérifiez en recalculant le résultat.

Solution simple

Voici le résultat d'exécution :

```
3
```

```
-2
```

```
Le resultat est 37
```

1.2 Réutilisation d'une variable / pgaffected3



Écrivez un programme qui ne fait rien.



Déclarez deux entiers x et y .



Affectez 3 à x .



Affectez x à y .



Affectez $3 * x + 1$ à x .



Affectez $2 * x - 5 * y$ à x .



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

x vaut $[x]$ et y vaut $[y]$



Donnez la valeur des variables à l'issue de votre programme.

Solution simple

On a :

- Initialement x vaut 3
- Puis y vaut x qui vaut 3
- Puis x vaut $3*3+1=10$
- Puis x vaut $2*10-5*3 = 20-15 = 5$
- Finalement :

x vaut 5 et y vaut 3



Testez.



Validez votre programme avec la solution.

Solution Java @[pgaffect3.java]

```
class PGAffect3 {  
  
public static void main(String[] args)  
{  
    int x, y;  
    x = 3;  
    y = x;  
    x = 3 * x + 1;  
    x = 2 * x - 5 * y;  
    System.out.println("x vaut " + x + " et y vaut " + y);  
}  
}
```

1.3 Exponentiation 16-ème / pgexponent1



Écrivez un programme qui saisit un entier X dans `x`.
Affichez l'invite :

Valeur de X?



Calculez le **plus rapidement** X^{16} dans `x`.

Aide simple

Calculez X^2 puis X^4 puis X^8 puis X^{16} .



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

La valeur de X^{16} est `[x]`



Testez. Exemple d'exécution :

Valeur de X? 3

La valeur de X^{16} est 43046721



Validez votre programme avec la solution.

Solution Java @pgexponent1.java

```
import java.util.Scanner;

class PGExponent1 {

    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int x;
        System.out.print("Valeur de X? ");
        x = input.nextInt();
        x = x * x;
        x = x * x;
        x = x * x;
        x = x * x;
        System.out.println("La valeur de X^16 est " + x);
    }
}
```

2 Appliquer le cours

2.1 Quelques calculs / pgcalcul1



Écrivez un programme qui saisit deux nombres dans les variables `n1` et `n2` de type entier.



Calculez la somme, la différence, le produit et le quotient de `n1` et `n2` et affecte les résultats respectivement aux variables de type réel `a`, `b`, `c` et `d`.



Affichez les valeurs de `a`, `b`, `c` et `d`.



Compilez et exécutez le programme.
Testez avec 14 et 3 (par exemple).



Modifiez le programme en déclarant `n1` et `n2` avec le type réel.
Recompilez et exécutez.
Que constatez-vous ?



Même question avec `n1` de type entier et `n2` de type réel.
Expliquez.

Solution Java

Lorsque dans l'expression `n1/n2` les deux opérandes (`n1` et `n2`) sont de type entier, l'opération qui s'effectue est la division *entière* (ou division « euclidienne ») : le résultat est le *quotient*. Dans le cas où `n1` vaut 14 et `n2` vaut 3, on obtient donc 4 qui est ensuite convertit en réel (d'où 4.0). Par contre, lorsqu'un opérande au moins est un réel, c'est la division *réelle* qui est effectuée.



Validez votre programme avec la solution.

Solution Java

@[pgcalcul1.java]

```
import java.util.Scanner;

class PGCalcul1 {

    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int n1, n2;
        System.out.print("Deux nombres? ");
        n1 = input.nextInt();
        n2 = input.nextInt();
        double a = n1 + n2;
        double b = n1 - n2;
```

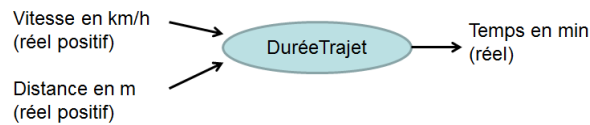
```
double c = n1 * n2;  
double d = n1 / n2;  
System.out.println("La somme est " + a);  
System.out.println("La difference est " + b);  
System.out.println("Le produit est " + c);  
System.out.println("La division est " + d);  
}  
}
```

2.2 Durée de trajet / pgtrajet1



Objectif

Cet exercice calcule la durée en minutes du trajet d'un véhicule :



Écrivez un programme qui saisit la vitesse moyenne en **km/h** d'un véhicule dans `vmoy` (réel) et la distance parcourue en **m** par ce véhicule dans `dist` (réel). Affichez les invites :

```
Vitesse moyenne (en km/h)?
Distance en m?
```



La formule qui lit les trois éléments est :

$$vitesse = \frac{distance}{temps} \quad \text{d'où} \quad temps = \frac{distance}{vitesse}$$

pour autant que les unités soient compatibles. Dans notre cas, on peut d'abord convertir la distance en km d'où le temps sera exprimée en heure que l'on divisera par 60 pour l'exprimer en minutes. Par conséquent :

Calculez la distance en km dans `km` (réel) puis affichez :

```
==> Distance en km : ...
==> Duree en minutes : ...
```



Autre solution : Calculez la vitesse en m/s dans `vt` (réel) et la durée en secondes du trajet de ce véhicule dans `nsecs` (entier).



Affichez finalement :

```
==> Vitesse en m/s : ...
==> Duree en secondes : ...
==> Duree en minutes : ...
```



Testez. Exemple d'exécution :

```
Vitesse moyenne (en km/h)? 50
Distance en m? 5000
==> Distance en km : 5
==> Duree en minutes : 6
==> Vitesse en m/s : 13.8888888889
==> Duree en secondes : 360
==> Duree en minutes : 6
```



Validez votre programme avec la solution.

Solution Java @[pgtrajet1.java]

```
import java.util.Scanner;

class PGTrajet1 {

public static void main(String[] args)
{
    Scanner input = new Scanner(System.in);
    double vmoy;
    System.out.print("Vitesse moyenne (en km/h)? ");
    vmoy = input.nextDouble();
    double dist;
    System.out.print("Distance en m? ");
    dist = input.nextDouble();
    double km = dist / 1000;
    System.out.println("==> Distance en km : " + km);
    System.out.println("==> Duree en minutes : " + (km / vmoy * 60));
    double vt = vmoy * 1000.0 / 3600.0;
    int nsecs = (int)(dist / vt);
    System.out.println("==> Vitesse en m/s : " + vt);
    System.out.println("==> Duree en secondes : " + nsecs);
    System.out.println("==> Duree en minutes : " + (nsecs / 60));
}
}
```

2.3 Un peu de trigo / pgtrigo1



Écrivez un programme qui saisit un entier dans `n`.
Affichez l'invite :

Valeur de n?



Calculez et affichez les valeurs de $\cos(n * \pi/2)$ et de $\sin(n * \pi/2)$.

Outil Java

La constante mathématique π est définie par `Math.PI`.



Testez avec plusieurs valeurs de `n`.
Concluez.

Solution simple

On remarque que les fonctions mathématiques sont des approximations.
Exemple d'exécution :

Valeur de n? 3
==> -1.8369701987210297E-16
==> -1.0



Validez votre programme avec la solution.

Solution Java @[pgtrigo1.java]

```
import java.util.Scanner;

class PGTrigo1 {

public static void main(String[] args)
{
    Scanner input = new Scanner(System.in);
    int n;
    System.out.print("Valeur de n? ");
    n = input.nextInt();
    double alpha = n * Math.PI / 2;
    System.out.println("==> " + Math.cos(alpha));
    System.out.println("==> " + Math.sin(alpha));
}
}
```

2.4 Date de Pâques / pgpagues1



Objectif

Depuis le concile de Nicée (325 ap.J.-C.) et la mise en place du calendrier grégorien (en 1591), la date de Pâques est le premier dimanche après la première pleine lune qui suit l'équinoxe de printemps (du 21 mars). Il existe plusieurs formules permettant de calculer la date de Pâques pour une année donnée dans le calendrier grégorien.

Voici les calculs permettant de connaître cette date [$x \bmod p$ dénote le reste de x dans la division euclidienne par p] :

$$\begin{aligned} a &= \text{annee} \bmod 19 \\ b &= \text{annee} \bmod 4 \\ c &= \text{annee} \bmod 7 \\ d &= (19a + 24) \bmod 30 \\ e &= (2b + 4c + 6d + 5) \bmod 7 \\ n &= 22 + d + e \end{aligned}$$

où n est le numéro du jour du mois de mars (ou avril si n est supérieur à 31) correspondant au dimanche de Pâques.



Écrivez un programme qui demande une année (qui est supposée comprise entre 1591 et 2099) puis calcule et affiche le « rang » de la date de Pâques correspondant.



Vérifiez qu'en 2002, Pâques est tombé le 31 mars, et qu'en 1996, le 7 avril.

Solution simple

Pour $\text{annee} = 2002$, on a :

$$\begin{aligned} a &= 2002 \bmod 19 = 7 \\ b &= 2002 \bmod 4 = 2 \\ c &= 2002 \bmod 7 = 0 \\ d &= (19 \times 7 + 24) \bmod 30 = 7 \\ e &= (2 \times 2 + 4 \times 0 + 6 \times 7 + 5) \bmod 7 = 2 \\ n &= 22 + 7 + 2 = 31 \end{aligned}$$

Donc Pâques est en mars et le quantième est le 31. Pour $\text{annee} = 1996$, les calculs indiquent que $n = 38$, ce qui signifie que le dimanche de Pâques est le 38^e jour à partir du 1^{er} mars, i.e. le 7 avril.



Validez votre programme avec la solution.

Solution Java @[pgpaques1.java]

```
import java.util.Scanner;

class PGPaques1 {

public static void main(String[] args)
{
    Scanner input = new Scanner(System.in);
    int annee;
    System.out.print("Votre annee dans [1592..2099]? ");
    annee = input.nextInt();
    int a = annee % 19;
    int b = annee % 4;
    int c = annee % 7;
    int d = (19 * a + 24) % 30;
    int e = (2 * b + 4 * c + 6 * d + 5) % 7;
    int n = 22 + d + e;
    System.out.println("Numero de jour = " + n);
}
}
```

3 Approfondir le cours

3.1 Moyennes de deux entiers / pgmoyenne1



Écrivez un programme qui saisit deux entiers dans `n1` et `n2`.
Affichez l'invite :

```
Deux entiers?
```



Calculez et affichez (où `[x]` désigne le contenu de `x`) :

```
Pour les entiers [n1] et [n2]:
```

```
    Moyenne decimale = ...
```

```
    Moyenne entiere  = ...
```



Testez. Exemple d'exécution :

```
Deux entiers? 3 8
```

```
Pour les entiers 3 et 8:
```

```
    Moyenne decimale : 5.5
```

```
    Moyenne entiere  : 5
```



Validez votre programme avec la solution.

Solution Java @[pgmoyenne1.java]

```
import java.util.Scanner;

class PGMoyenne1 {

public static void main(String[] args)
{
    Scanner input = new Scanner(System.in);
    int n1, n2;
    System.out.print("Deux entiers? ");
    n1 = input.nextInt();
    n2 = input.nextInt();
    System.out.println("Pour les entiers " + n1 + " et " + n2 + ":");
    System.out.println("    Moyenne decimale = " + ((n1 + n2) / 2.0));
    System.out.println("    Moyenne entiere  = " + ((n1 + n2) / 2));
}
}
```

3.2 Codage d'une date / pgdate1



Objectif

On choisit de coder une date par un entier composé de la façon suivante :

$$a \times 10000 + m \times 100 + j$$

où a est l'année ($a \geq 1$), m le mois ($1 \leq m \leq 12$) et j le jour ($1 \leq j \leq 31$).

Exemple : La date « 14 juillet 1789 » sera codée par l'entier 17890714.

Cet exercice décode une telle date (dont on supposera qu'elle est valide).



Écrivez un programme qui saisit une date codée dans un entier `dt`.

Affichez l'invite :

Votre date codee?



Calculez, les trois variables étant de type entier :

- Le numéro du jour dans `j`.
- Le numéro du mois dans `m`.
- Et l'année dans `a`.



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

jour=[j] mois=[m] annee=[a]



Testez. Exemple d'exécution :

Votre date codee? 20161206

jour=6 mois=12 annee=2016



Validez votre programme avec la solution.

Solution Java @pgdate1.java

```
import java.util.Scanner;

class PGDate1 {

    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int dt;
        System.out.print("Votre date codee? ");
        dt = input.nextInt();
        int j = dt % 100;
        int m = (dt / 100) % 100;
        int a = dt / 10000;
        System.out.println("jour=" + j + " mois=" + m + " annee=" + a);
    }
}
```

}

}