

Initiation à la programmation en Java

Jean-Luc Baril, Elisabeth Gavignet, Richard Genestier, Joel Savelli,
Vincent Vajnovszki
LIB, Université de Bourgogne
B.P. 47 870, 21078 DIJON-Cedex France

October 14, 2020

Ce document propose d'aider le lecteur à apprendre les bases de programmation en langage Java. Il est constitué de plusieurs chapitres développant chacun un concept fondamental: Entrées/Sorties, Conditionnelles, Répétitives, Tableaux, Chaînes de caractères, Fonctions, Récursivité. Pour chaque concept, un *minimum à savoir* et une liste d'exercices corrigés sont proposés. Il s'agit d'un document complémentaire au cours d'Initiation à la programmation Java accessible en cliquant sur le lien <http://jl.baril.u-bourgogne.fr/coursinfo.pdf>. Avant de commencer, vous devez télécharger le fichier Lire.java à l'adresse <http://jl.baril.u-bourgogne.fr/licence1.html> et le compiler dans le répertoire de travail en exécutant la commande `javac Lire.java`.

1 Les entrées/sorties

1.1 Le minimum à savoir

Pour afficher à l'écran le message *Bonjour* :

```
System.out.print("Bonjour");//sans retour à la ligne  
System.out.println("Bonjour");//avec retour à la ligne
```

Pour afficher à l'écran le contenu de la variable *x* :

```
System.out.print(x);//sans retour à la ligne  
System.out.println(x);//avec retour à la ligne
```

Pour afficher *Bonjour* et le contenu de la variable *x* :

```
System.out.print("Bonjour"+x);//sans retour à la ligne  
System.out.println("Bonjour"+x);//avec retour à la ligne
```

Pour saisir un entier au clavier et le mettre dans la variable *x* :

```
x=Lire.i();
```

Pour saisir un réel au clavier et le mettre dans la variable *x* :

```
x=Lire.d();
```

Pour saisir un caractère au clavier et le mettre dans la variable *x* :

```
x=Lire.c();
```

1.2 Exercices corrigés

Exercice 3: Périmètre. Ecrire un programme qui calcule le périmètre d'un triangle rectangle dont on connaît l'hypothénuse et l'un des côtés de l'angle droit.

```
public class Ex3
{
    public static void main(String args[])
    {
        double x,y,z;
        x = Lire.d();
        z = Lire.d();
        if(z>=x)
        {
            y = Math.sqrt(z*z-x*x);
            System.out.println("périmètre : "+(x+y+z));
        }
    }
}
```

Exercice 4: Conversion secondes. Ecrire un programme qui permet d'exprimer un nombre de secondes saisi par l'utilisateur en heures, minutes et secondes.

```
public class Ex4
{
    public static void main(String args[])
    {
        System.out.println("Donner un nb de secondes : ");
        int s = Lire.i();

        int h = s/3600;
        int tmp = s%3600;
        int m = tmp/60;
        s = tmp%60;
        System.out.println("Ce nb de secondes vaut : "+h+" h "+m+" min "+s+"
            secondes");

        /* autre solution
        int h = s/3600;
        System.out.print(h+" h ");
        h = s%3600;
        int m = h/60;
        s = h%60;
        System.out.println(m+" min "+s+" s");
        */
    }
}
```

Exercice 5: Utilisation d'un boolean. Ecrire un programme qui permet de saisir un nombre entier puis qui affiche **true** si le nombre appartient à l'intervalle [1, 10] et **false** sinon (ne pas utiliser de conditionnelle).

```
public class Ex5
{
    public static void main(String args[])
    {
```

```

    int x;
    boolean bool;
    System.out.println("Entrer un entier");
    x = Lire.i();
    bool=x >= 1 && x <= 10;
    System.out.println(bool);
}
}

```

Exercice 6: Majeurs. Ecrire un programme qui permet de saisir les âges de 5 personnes puis qui affiche **true** si au moins 4 personnes sont adultes (âge> 18) et **false** sinon (ne pas utiliser de conditionnelle).

```

public class Ex6
{
    public static void main(String args[])
    {
        int a1,a2,a3,a4,a5;
        boolean bool;
        System.out.println("age 1 : ");
        a1 = Lire.i();
        System.out.println("age 2 : ");
        a2 = Lire.i();
        System.out.println("age 3 : ");
        a3 = Lire.i();
        System.out.println("age 4 : ");
        a4 = Lire.i();
        System.out.println("age 5 : ");
        a5 = Lire.i();
        bool=(a1>18 && a2>18 && a3>18 && a4>18) || (a1>18 && a2>18 && a3>18
            && a5>18) || (a1>18 && a2>18 && a4>18 && a5>18) || (a1>18 && a3>18
            && a4>18 && a5>18) || (a2>18 && a3>18 && a4>18 && a5>18);
        System.out.println(bool);
    }
}

```

2 Les conditionnelles

2.1 Le minimum à savoir

La structure **if** permet de discuter en fonction d'une condition (expression booléenne). Le **else** est facultatif.

```
if(condition)
{ //instructions si la condition est true }
```

```
if(condition)
{ //instructions si la condition est true }
else
{ //instructions si la condition est false }
```

La structure **switch** permet de discuter en fonction du contenu d'une variable. L'ajout d'un **break** permet de sortir immédiatement de la structure.

```
switch(x)
{ case 3 : //instructions si la variable x vaut 3
  case 5 : //instructions si la variable x vaut 5
}
```

```
switch(x)
{ case 3 : //instructions si la variable x vaut 3 ;break
  case 5 : //instructions si la variable x vaut 5 ;
}
```

2.2 Exercices corrigés

Exercice 1: Maxi. Ecrire un programme qui

- affiche le maximum de deux entiers saisis par l'utilisateur,
- affiche le maximum de trois entiers saisis par l'utilisateur,
- affiche en ordre croissant trois entiers saisis par l'utilisateur.

```
public class Max
{
    public static void main(String args[])
    {
        int x,y,z,max;
        System.out.println("Donner nb 1 :");
        x = Lire.i();
        System.out.println("Donner nb 2 :");
        y = Lire.i();
        System.out.println("Donner nb 3 :");
        z = Lire.i();

        //2
        if(x>y) max=x;
        else max=y;
        if (z>max) max=z;
        System.out.println("max = "+max);
    }
}
```

```

//3
if(x == max) {
    if (y <= z) System.out.println("ordre croissant = "+y+" "+z+" "+x);
    else System.out.println("ordre croissant = "+z+" "+y+" "+x);
}
else {
    if(y == max) {
        if (x <= z) System.out.println("ordre croissant = "+x+" "+z+" "+y);
        else System.out.println("ordre croissant = "+z+" "+x+" "+y);
    }
    else {
        if (x <= y) System.out.println("ordre croissant = "+x+" "+y+" "+z);
        else System.out.println("ordre croissant = "+y+" "+x+" "+z);
    }
}
}
}
}

```

Exercice 2: Age. Ecrire un programme qui permet de saisir l'âge de 5 personnes puis qui affiche **il n'y a pas de mineurs** si les 5 âges sont supérieurs ou égaux à 18, qui affiche **il y a un seul mineur** si un seul âge est inférieur à 18, et qui affiche **il y a au moins deux mineurs** sinon.

```

public class Mineur
{
    public static void main(String args[])
    {
        int age, mineur=0;
        System.out.println("Donner un age :");
        a = Lire.d();
        if(a<18)
            mineur = mineur +1;
        System.out.println("Donner un age :");
        a = Lire.d();
        if(a<18)
            mineur = mineur +1;
        System.out.println("Donner un age :");
        a = Lire.d();
        if(a<18)
            mineur = mineur +1;
        System.out.println("Donner un age :");
        a = Lire.d();
        if(a<18)
            mineur = mineur +1;
        System.out.println("Donner un age :");
        a = Lire.d();
        if(a<18)
            mineur = mineur +1;
        if(mineur==0)
            System.out.println("Il n y a pas de mineur");
        else
            if(mineur==1)
                System.out.println("Il y a 1 seul mineur");
            else

```

```
System.out.println("Il y au moins deux mineur");
```

Exercice 3: Equation. Ecrire un programme qui trouve les solutions de l'équation $ax^2 + bx + c = 0$.

```
public class Equation
{
    public static void main(String args[])
    {
        double a, b, c, discri;
        System.out.println("Donner a :");
        a = Lire.d();
        System.out.println("Donner b :");
        b = Lire.d();
        System.out.println("Donner c :");
        c = Lire.d();
        if (a == 0 && b == 0 && c == 0)
            System.out.println("infinité de solutions réelles");
        else
            if (a == 0 && b == 0) {
                System.out.println("pas de solution");
            }
            else
                if (a == 0)
                    System.out.println("x vaut "+ (-c / b));
                else
                    {
                        discri = b * b - 4 * a * c;
                        if (discri < 0)
                            System.out.println("pas de solution réelle");
                        else
                            if (discri == 0)
                                System.out.println("x a comme solution réelle "+(-b / (2 *
                                    a)));
                            else
                                System.out.println("x a comme solutions réelles "+(-b +
                                    Math.sqrt(discri)) / (2 * a) +" et "+(-b -
                                    Math.sqrt(discri)) / (2 * a));
                    }
            }
    }
}
```

Exercice 4: Chiffre romain. Ecrire un programme qui convertit un entier entre 1 et 5 dans son écriture romaine (1 → I, 2 → II, 3 → III, 4 → IV, 5 → V).

```
public class Romain
{
    public static void main(String args[])
    {
        int x;
        do
        {
            System.out.println("Entrer un entier entre 1 et 5");
            x = Lire.i();
        }while(x<1 || x>5);
        switch(x)
    }
}
```

```

        {case 1: {System.out.println("I");break;}
        case 2: {System.out.println("II");break;}
        case 3: {System.out.println("III");break;}
        case 4: {System.out.println("IV");break;}
        case 5: {System.out.println("V");break;}
        }
    }
}

```

Exercice 5: Calculatrice. Ecrire un programme qui simule le fonctionnement d'une calculatrice à 4 opérations en utilisant l'instruction `switch ...case` pour le choix de l'opérateur.

```

public class Calcul
{
    public static void main(String args[])
    {
        double x,y;
        char o;
        System.out.println("Donner nb 1 :");
        x = Lire.d();
        System.out.println("Donner nb 2 :");
        y = Lire.d();
        System.out.println("Donner l'opérateur :");
        o = Lire.c();
        switch(o) {
            case '+' : System.out.println("le résultat est "+(x+y)); break;
            case '-' : System.out.println("le résultat est "+(x-y)); break;
            case '*' : System.out.println("le résultat est "+(x*y)); break;
            case '/' : if (y == 0)
                System.out.println("impossible");
                else
                System.out.println("le résultat est "+(x/y));
                break;
            default : System.out.println("opérateur incorrect");
        }
    }
}

```

Exercice 6: Cible. Proposer un programme permettant le tirage aléatoire de 4 points (x, y) avec $x, y \in [0, 1[$, et l'affichage du nombre de ces points situés dans un disque de centre $(0.5, 0.5)$ et de rayon 0.5.

```

public class Cible
{
    public static void main(String args[])
    {
        double x,y,d;
        int s = 0;
        x = Math.random();
        y = Math.random();
        System.out.println("x = "+x+" y = "+y);
        d = Math.sqrt((x-0.5)*(x-0.5) + (y-0.5)*(y-0.5));
        if(d < 0.5)
            s=s+1;
        x = Math.random();
    }
}

```

```

y = Math.random();
System.out.println("x = "+x+" y = "+y);
d = Math.sqrt((x-0.5)*(x-0.5) + (y-0.5)*(y-0.5));
if(d < 0.5)
    s=s+1;
x = Math.random();
y = Math.random();
System.out.println("x = "+x+" y = "+y);
d = Math.sqrt((x-0.5)*(x-0.5) + (y-0.5)*(y-0.5));
if(d < 0.5)
    s=s+1;
x = Math.random();
y = Math.random();
System.out.println("x = "+x+" y = "+y);
d = Math.sqrt((x-0.5)*(x-0.5) + (y-0.5)*(y-0.5));
if(d < 0.5)
    s=s+1;
System.out.println("nb de pts dans la cible : "+s);
}
}

```

Exercice 7: Lendemain. Ecrire un programme qui calcule la date du lendemain d'une date saisie par trois entiers, **jour** pour le numéro de jour dans la semaine, **mois** pour le numéro de mois, et **annee** pour l'année. Le mois devra être affiché en toutes lettres. On ne tiendra pas compte des années bissextiles.

```

public class Lendemain2
{
    public static void main(String args[])
    {
        int jour,mois,annee;

        System.out.println("Jour?");
        jour = Lire.i();
        System.out.println("Mois?");
        mois = Lire.i();
        System.out.println("Annee?");
        annee = Lire.i();

        int jourLendemain = jour;
        int moisLendemain = mois;
        int anneeLendemain = annee;

        if (mois == 1 || mois == 3 || mois == 5 || mois == 7 || mois == 8 ||
            mois == 10 || mois == 12)
            if (jour < 31)
                jourLendemain = jour + 1;
            else
                if (jour == 31)
                {
                    jourLendemain = 1;
                    if (mois == 12)
                    {
                        anneeLendemain = annee+1;
                    }
                }
            }
    }
}

```



```

        moisLendemain = 1;
    }
    else
        moisLendemain = mois + 1;
    }
    else
        System.out.println("jour incorrect");
if (mois == 4 || mois == 6 || mois == 9 || mois == 11)
    if (jour < 30) jourLendemain = jour + 1;
    else
    {
        if (jour == 30)
        {
            jourLendemain = 1;
            moisLendemain = mois + 1;
        }
        else
            System.out.println("jour incorrect");
    }
if (mois == 2)
    if (jour < 28)
        jourLendemain = jour + 1;
    else
    {
        if(jour==28)
        {
            jourLendemain = 1;
            moisLendemain = mois + 1;
        }
        else
            System.out.println("jour incorrect");
    }
}
}
}

```

3 Les répétitives

3.1 Le minimum à savoir

La boucle for permet de répéter plusieurs instructions en faisant varier un indice d'une valeur donnée à une autre valeur donnée en incrémentant ou décrémentant cet indice.

```
for(i=1;i<=10;i++)
{ \\Instructions }
for(i=10;i>=1;i--)
{ \\Instructions }
```

La boucle while permet de répéter plusieurs instructions tant qu'une condition est vraie. On évalue la condition avant d'entrer dans la boucle.

```
while(condition)
{ \\Instructions }
```

La boucle do...while permet de répéter plusieurs instructions tant qu'une condition est vraie. On évalue la condition à la fin de la boucle.

```
do
{ \\Instructions }
}while(condition);
```

3.2 Exercices corrigés

Exercice 1: Majeurs. Ecrire un programme qui permet de saisir l'âge de 20 personnes puis qui indique le nombre d'adultes et d'enfants.

```
public class Majeurs
{
    public static void main(String args[])
    {
        int a, c = 0;
        for (int i = 0; i < 20; i++) {
            System.out.println("age : ");
            a = Lire.i();
            if (a >= 18) c = c+1;
        }
        System.out.println("Il y a : "+c+" adultes et "+(20-c)+" enfants");
    }
}
```

Exercice 2: Moyenne pondérée. Ecrire un programme qui calcule la moyenne pondérée d'une suite de notes. Les notes ainsi que leur coefficients sont saisis au clavier jusqu'à ce que la note -1 soit entrée. Les notes sont des réels entre 0 et 20, les coefficients sont des entiers entre 1 et 10.

```
public class Notes
{
    public static void main(String args[])
    {
        double n=0, snp=0;
        int c=0, sc=0;
```

```

do {
  do {
    System.out.println("note entre 0 et 20 ou -1 pour quitter: ");
    n = Lire.d();
  }
  while (n != -1 && (n < 0 || n > 20));
  if (n != -1)
  {
    do {
      System.out.println("coef entre 1 et 10: ");
      c = Lire.i();
    }
    while (c < 0 || c > 10);
    snp = snp+c*n;
    sc = sc+c;
  }
}
while (n != -1);
if (snp != 0)
  System.out.println("moyenne : "+snp/sc);
else
  System.out.println("pas de note");
}
}

```

Exercice 3: Loto. Ecrire un programme qui génère aléatoirement six nombres entiers différents entre 1 et 49 inclus.

```

public class Loto
{
  public static void main(String args[])
  {
    int x1, x2, x3, x4, x5, x6;
    x1 = (int)(Math.random()*49+1);
    do x2 = (int)(Math.random()*49+1); while (x2 == x1);
    do x3 = (int)(Math.random()*49+1); while (x3 == x2 || x3 == x1);
    do x4 = (int)(Math.random()*49+1); while (x4 == x3 || x4 == x2 || x4 ==
      x1);
    do x5 = (int)(Math.random()*49+1); while (x5 == x4 || x5 == x3 || x5 ==
      x2 || x5 == x1);
    do x6 = (int)(Math.random()*49+1); while (x6 == x5 || x6 == x4 || x6
      == x3 || x6 == x2 || x6 == x1);
    System.out.println(x1+" "+x2+" "+x3+" "+x4+" "+x5+" "+x6);
  }
}

```

Exercice 4: Pi. Ecrire un programme qui calcule pour n donné la valeur de $1 + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \frac{1}{25} + \dots + \frac{1}{n}$ en utilisant successivement les boucles **for**, **while** et **do ...while**. En déduire une valeur approchée de π sachant que $\frac{\pi^2}{6} = \sum_{i=1}^{\infty} \frac{1}{i^2}$.

```

public class Pi1
{
  public static void main(String args[])

```

```

{
    int n;
    double x;
    x = 0;
    System.out.println("Donner n");
    n = Lire.i();
    for (int i = 1; i <= n; i++) {
        x = x + 1.0/(i*i); // Ne pas faire la division entière 1/(i*i)
    }
    System.out.println("Valeur approchée de pi : " + Math.sqrt(6*x));
}
}

public class Pi2
{
    public static void main(String args[])
    {
        int n,i;
        double x;
        x = 0;
        System.out.println("Donner n");
        n = Lire.i();
        i=1;
        while(i<=n)
        {
            x = x + 1.0/(i*i); // Ne pas faire la division entière 1/(i*i)
            i=i+1;
        }
        System.out.println("Valeur approchée de pi : " + Math.sqrt(6*x));
    }
}

public class Pi3
{
    public static void main(String args[])
    {
        int n,i;
        double x;
        x = 0;
        System.out.println("Donner n");
        n = Lire.i();
        i=1;
        do
        {
            x = x + 1.0/(i*i); // Ne pas faire la division entière 1/(i*i)
            i=i+1;
        }
        while(i<=n);
        System.out.println("Valeur approchée de pi : " + Math.sqrt(6*x));
    }
}

```

Exercice 5: PGCD. Ecrire un programme pour calculer, par la méthode d'Euclide, le PGCD de deux nombres entiers.

```
public class pgcd
{
    public static void main(String args[])
    {
        int a,b;
        System.out.println("Entrer a");
        a = Lire.i();
        System.out.println("Entrer b");
        b = Lire.i();
        while(a%b != 0)
        {r=a%b;
          a=b;
          b=r;
        }
        System.out.println("pgcd = "+b);
    }
}
```

Exercice 6: Pierre - Feuille - Ciseau. Ecrire un programme qui permet de jouer à Pierre - Feuille - Ciseau contre l'ordinateur. Le jeu d'arrête lorsque l'un des deux joueurs a 5 points; le nom du gagnant doit s'afficher à l'écran.

```
public class pierrefeuilleciseau
{
    public static void main(String args[])
    {
        int joueur, ordi, ptjoueur=0,ptordi=0;
        do
        { ordi=(int)(1+3*Math.random());
          System.out.println("Tapez 1 pour pierre, 2 pour feuille et 3 pour
            ciseau");
          joueur=Lire.i();
          if((ordi==1 && joueur==3)|| (ordi==2 && joueur==1)|| (ordi==3 &&
            joueur==2))
            ptordi=ptordi+1;
          if((joueur==1 && ordi==3)|| (joueur==2 && ordi==1)|| (joueur==3 &&
            ordi==2))
            ptjoueur=ptjoueur+1;
        }
        while(ptjoueur<5 && ptordi<5);
        if(ptjoueur==5)
            System.out.println("Vous avez gagné");
        else
            System.out.println("L'ordinateur a gagné");
    }
}
```

4 Révisions

4.1 Le minimum à savoir

Conseil du vieux sage :

Le meilleur moment pour planter un arbre était il y a 20 ans.

Le deuxième meilleur moment est maintenant!!!

Moralité : On bosse!

4.2 Exercices corrigés

Exercice 1: Joli dessin. On souhaite afficher pour tout $n \geq 4$ le dessin suivant:

```
* * * ...
 * * ...
* * * ...
 * * ...
* * * ...
. . .
. . .
```

```
public class Dessin
{
    public static void main (String [] args)
    {
        int n,i,j;
        do
        {
            System.out.println("Entrer la valeur de n:");
            n=Lire.i();
        }
        while(n<4);
        for (i=1;i<=n;i++)
        {
            for(j=1;j<=n;j++)
                if((i+j)%2==0)
                    System.out.print("*");
                else
                    System.out.print(" ");
            System.out.println();
        }
    }
}
```

Exercice 2: Combinaisons C_n^p . Ecrire un programme permettant de calculer $n!$. En déduire un programme permettant de calculer $C_n^p = \frac{n!}{p!(n-p)!}$.

```
public class Combinaison
{
    public static void main (String [] args)
    {
        int n,p,i,j,Cnp,factn,factp,factnp;
```

```

System.out.println("Entrer la valeur de n:");
n=Lire.i();
System.out.println("Entrer la valeur de p:");
p=Lire.i();

factn=1;
for(i=1;i<=n;i++)
    factn=factn*i;

factp=1;
for(i=1;i<=p;i++)
    factp=factp*i;

factnp=1;
for(i=1;i<=n-p;i++)
    factnp=factnp*i;

Cnp=factn/(factp*factnp);
System.out.println(Cnp);
}
}

```

Vérifier alors la formule (sans utiliser *Math.pow*) $\sum_{p=0}^n C_n^p = 2^n$

```

public class Verif
{
    public static void main (String [] args)
    {
        int n,p,i,j,Cnp,factn,factp,factnp,Somme,puis;
        do
        {System.out.println("Entrer la valeur de n:");
          n=Lire.i();
        }
        while(n<0);
        factn=1;
        for(i=1;i<=n;i++)
            factn=factn*i;

        Somme=0;
        for(p=0;p<=n;p++)
        {
            factp=1;
            for(i=1;i<=p;i++)
                factp=factp*i;

            factnp=1;
            for(i=1;i<=n-p;i++)
                factnp=factnp*i;

            Cnp=factn/(factp*factnp);
            Somme=Somme+Cnp;
        }
    }
}

```

```

}
puis=1;
for(i=1;i<=n;i++)
    puis=puis*2;
if(Somme==puis)
    System.out.println("Formule juste");
else
    System.out.println("Formule fausse");
}
}

```

Refaire le programme du calcul de C_n^p pour qu'il fonctionne pour des valeurs de n plus grandes. Remarquons que : $C_n^p = \frac{n!}{p!(n-p)!} = \frac{n(n-1)\dots(n-p+1)(n-p)\dots 3 \ 2 \ 1}{p!(n-p)(n-p-1)\dots 3 \ 2 \ 1}$. Donc on a : $C_n^p = \frac{n}{1} \cdot \frac{n-1}{2} \cdot \frac{n-2}{3} \dots \frac{n-i+1}{i} \dots \frac{n-p+1}{p}$.

```

public class Rapide
{
    public static void main (String [] args)
    {
        int n,p,i;
        double Cnp;

        System.out.println("Entrer la valeur de n:");
        n=Lire.i();

        System.out.println("Entrer la valeur de p:");
        p=Lire.i();

        Cnp=1;
        for(i=1;i<=p;i++)
            Cnp=Cnp*(n-i+1)/i;

        System.out.println(Cnp);
    }
}

```

Exercice 3: Base 2. Ecrire un programme qui réalise la décomposition binaire d'un entier positif. Exemple: la décomposition binaire de 26 est 11010. On pourra utiliser une chaîne de caractères pour stocker la décomposition binaire.

```

public class Binaire
{
    public static void main (String [] args)
    {
        int n;
        String binaire;
        System.out.println("Entrer la valeur de n:");
        n=Lire.i();

        binaire="";
        do
        {
            if(n%2==1)
                binaire="1"+binaire;

```



```

        else
            binaire="0"+binaire;
        n=n/2;
    }
    while (n!=0);
    System.out.println(binaire);
}
}

```

Exercice 4: Jeu. Ecrire un programme qui permet à un utilisateur de rechercher un nombre entier aléatoire entre 1 et 100 généré par la machine. On imposera au plus 7 coups et on affichera à chaque coup si le nombre proposé par l'utilisateur est trop grand ou trop petit.

```

public class Jeu
{
    public static void main (String [] args)
    {
        int ordi ,joueur ,coups;
        ordi=(int) (1+100*Math.random());
        coups=0;
        do
        {
            System.out.println("Proposer votre nombre");
            joueur=Lire.i();
            coups=coups+1;
            if(joueur>ordi)
                System.out.println("Votre nbre est trop grand");
            if(joueur<ordi)
                System.out.println("Votre nbre est trop petit");
        }
        while(joueur!=ordi && coups<7);

        if(joueur==ordi)
            System.out.println("Bravo vous avez trouve en "+coups+"coups");
        else
            System.out.println("Perdu");
        }
}

```

Exercice 5: Approximation de π . Nous présentons dans cet exercice deux formules permettant d'obtenir une approximation de π .

a) La formule de Wallis (1665) $\frac{\pi}{2} = \frac{2}{1} \cdot \frac{2}{3} \cdot \frac{4}{3} \cdot \frac{4}{5} \cdot \frac{6}{5} \cdot \frac{6}{7} \dots$. Ecrire un programme permettant de trouver une approximation de π en utilisant les 1000 premiers facteurs de ce produit.

```

public class Wallis
{
    public static void main (String [] args)
    {
        int n,i,num,denom;
        double p;
        System.out.println("Entrer la valeur de n:");
        n=Lire.i();

        p=1;
    }
}

```

```

num=0;
denom=1;
for(i=1;i<=n;i++)
{
    if(num<denom)
        num=num+2;
    else
        denom=denom+2;
    p=p*num/denom;
}

System.out.println(2*P);
}
}

```

b) La formule de Viète (1593) $\frac{2}{\pi} = \frac{\sqrt{2}}{2} \cdot \frac{\sqrt{2+\sqrt{2}}}{2} \cdot \frac{\sqrt{2+\sqrt{2+\sqrt{2}}}}{2} \dots$

```

public class Viète
{
    public static void main (String [] args)
    {
        int n,i,denom;
        double P,num;
        System.out.println("Entrer la valeur de n:");
        n=Lire.i();

        P=1;
        num=0;
        denom=2;
        for(i=1;i<=n;i++)
        {
            num=Math.sqrt(2+num);
            P=P*num/denom;
        }

        System.out.println(2/P);
    }
}

```

Exercice 6: Les nombres parfaits, amis et d'Amstrong. a) Un entier naturel est un nombre parfait s'il est égal à la somme de ses diviseurs propres. Par exemple, les diviseurs de 6 sont 1,2,3,6; ses diviseurs propres sont 1,2,3; la somme de ses diviseurs propres est $1 + 2 + 3 = 6$; 6 est donc un nombre parfait. Ecrire un programme qui affiche tous les nombres parfaits inférieurs à 10000.

```

public class Parfait
{
    public static void main (String [] args)
    {int n,s,i;

        for(n=1;n<=10000;n++)
        {
            s=0;
            for(i=1;i<=n/2;i++)

```

```

        if (n%i==0)
            s=s+i;
    if (s==n)
        System.out.print(n+" ");
    }

System.out.println();
}
}

```

b) Deux entiers naturels sont des nombres amis si chacun d'entre eux est égal à la somme des diviseurs propres de l'autre. Ecrire un programme qui affiche la liste des couples de nombres amis inférieurs à 10000.

```

public class Amis
{
    public static void main (String [] args)
    {
        int n,s,i,somme;
        for(n=1;n<=10000;n++)
        {
            s=0;
            for(i=1;i<=n/2;i++)
                if (n%i==0)
                    s=s+i;
            somme=0;
            for(i=1;i<=s/2;i++)
                if (s%i==0)
                    somme=somme+i;
            if (somme==n)
                System.out.println(n+" est "+s+" sont amis");
        }
    }
}

```

b) On appelle nombre d'Armstrong, un entier naturel autre que 0 et 1 qui a la propriété d'être égal à la somme des cubes de ses chiffres. Ecrire un programme permettant d'afficher tous les nombres d'Armstrong compris entre 2 et 1000.

```

public class Armstrong
{
    public static void main (String [] args)
    {
        int n,somme,unite,dizaine,centaine;
        for(n=2;n<=1000;n++)
        {
            unite=n%10;
            dizaine=((n-unite)/10)%10;
            centaine=n/100;
            somme=unite*unite*unite+dizaine*dizaine*dizaine+centaine*centaine*centaine;
            if (somme==n)
                System.out.print(n+" ");
        }
        System.out.println();
    }
}

```

```
}
```

Exercice 7: Evolution d'une urne. Soit une urne contenant deux boules, une noire et une rouge. On veut simuler le schéma suivant: on tire une boule, si celle-ci est rouge on remet cette boule dans l'urne et on ajoute une autre boule rouge; si on tire une boule noire on la remet dans l'urne et on ajoute deux autres boules noires. Le but de l'exercice est d'écrire un programme qui affiche à chaque étape le nombre de boules de chaque couleur.

```
public class Urne
{
    public static void main (String [] args)
    {
        int n,i,rouge,noire,alea;
        System.out.println("Entrer le nombre de tirages");
        n=Lire.i();
        rouge=1;
        noire=1;
        for(i=1;i<=n;i++)
        {
            alea=(int)(1+(rouge+noire)*Math.random());
            if(alea<=rouge)
                rouge=rouge+1;
            else
                noire=noire+2;
            System.out.println("Rouge="+rouge+" et Noire="+noire);
        }
    }
}
```

5 Les tableaux.

5.1 Le minimum à savoir

Déclaration d'un tableau 1D de 10 cases contenant des entiers:

```
int tab[];
tab=new int[10];
```

Accès à la i -ième case du tableau tab: tab[i-1]

Déclaration d'un tableau 2D 2 lignes et 10 colonnes contenant des entiers:

```
int tab[][];
tab=new int[2][10];
```

Accès à la case située sur la i -ième ligne et j -ième colonne: tab[i-1][j-1]

Attention : Les indices pour repérer les cases (resp. lignes, colonnes) des tableaux commencent à 0.

5.2 Exercices corrigés

Exercice 1 : Tableau à une dimension. Soit Tab un tableau à une dimension de taille 10 contenant des entiers. On souhaite écrire un programme qui propose à l'utilisateur un menu composé de la liste des opérations ci-dessous:

- 1) Remplir aléatoirement le tableau Tab avec 10 entiers compris entre 1 et 10 inclus.