

TD1 (exercices corrigés)

**Exercice 1:** Calcul et affichage de la somme, de la différence, du produit, du quotient et du reste de la division entière de 2 nombres entiers.

**Spécifications complémentaires:**

- Il s'agit de définir une application en Java qui affichera la somme, la différence, le produit, le quotient et le reste de la division entière de 2 nombres entiers > 0 fournis préalablement (sans erreur) par l'utilisateur.
- Par exemple, si l'utilisateur fournit les nombres 11 puis 2, l'affichage réalisé sera:  
11 + 2 = 13  
11 - 2 = 9  
11 \* 2 = 22  
11 / 2 = 5  
11 % 2 = 1

**Algorithme:**

Demander à l'utilisateur de fournir 2 nombres entiers (grâce au clavier).  
Calculer la somme, la différence, le produit, le quotient et le reste de la division entière de 2 nombres.  
Afficher ces informations en respectant le format décrit ci-dessus.

Remarque: cet algorithme ne fait appel qu'à des opérations primitives (acquisition de nombres, calculs élémentaires, affichages d'informations connues), il n'est donc pas nécessaire de le détailler davantage.

**Analyse de l'algorithme pour sa transformation en un programme (Java):**

- Bilan des informations manipulées:
  - les 2 nombres entiers à acquérir,
  - la somme, la différence, le produit, le quotient et le reste de la division (toutes ces informations sont également des entiers).D'où:

```
int  nombre1, nombre2;  
int  somme, difference, produit, quotient, reste;
```
- 1<sup>ère</sup> phrase de l'algorithme:  
il s'agit de l'acquisition de 2 nombres entiers, il convient donc de renseigner l'utilisateur par l'affichage d'un message avant de procéder à chacune d'elles.  
D'où:

```
System.out.println(" Donnez moi un nombre ");  
nombre1 = Clavier.readInt();  
System.out.println(" Donnez moi un autre nombre ");  
nombre2 = Clavier.readInt();
```
- 2<sup>ème</sup> phrase de l'algorithme:  
il s'agit d'une suite de 5 calculs élémentaires.  
D'où:

```
somme      = nombre1 + nombre2;  
difference = nombre1 - nombre2;  
produit    = nombre1 * nombre2;  
quotient   = nombre1 / nombre2;  
reste      = nombre1 % nombre2;
```

→ 3<sup>ème</sup> phrase de l'algorithme:

il s'agit d'une suite de 5 affichages (selon le format précisé auparavant).D'où:

```
System.out.println( nombre1 + " + " + nombre2 + " = " + somme);  
System.out.println( nombre1 + " - " + nombre2 + " = " + difference);  
System.out.println( nombre1 + " * " + nombre2 + " = " + produit);  
System.out.println( nombre1 + " / " + nombre2 + " = " + quotient);  
System.out.println( nombre1 + " % " + nombre2 + " = " + reste);
```

**Programme Java résultant:**

```
class TdlExo1  
{  
    public static void main ( String args[])  
    {  
        int     nombre1, nombre2;  
        int     somme, difference, produit, quotient, reste;  
  
        System.out.println(" Donnez moi un nombre ");  
        nombre1 = Clavier.readInt();  
        System.out.println(" Donnez moi un autre nombre ");  
        nombre2 = Clavier.readInt();  
  
        somme      = nombre1 + nombre2;  
        difference = nombre1 - nombre2;  
        produit   = nombre1 * nombre2;  
        quotient  = nombre1 / nombre2;  
        reste     = nombre1 % nombre2;  
  
        System.out.println( nombre1 + " + " + nombre2 + " = " + somme);  
        System.out.println( nombre1 + " - " + nombre2 + " = " + difference);  
        System.out.println( nombre1 + " * " + nombre2 + " = " + produit);  
        System.out.println( nombre1 + " / " + nombre2 + " = " + quotient);  
        System.out.println( nombre1 + " % " + nombre2 + " = " + reste);  
    }  
}
```

**Exercice 2:** Rendre de la monnaie, i.e. calcul puis affichage du nombre (minimal) de pièces de 20, 10, 5, 2 et 1 Francs correspondant à un montant donné en Francs.

**Spécifications complémentaires:**

- L'utilisateur fournira un montant en Francs (nombre entier positif ou nul).
- On fera l'hypothèse qu'il n'y aura pas d'erreur de saisie
- Par exemple, si l'utilisateur fournit comme montant 69 F, l'affichage réalisé sera:  
$$69F = 3 \times 20F + 0 \times 10F + 1 \times 5F + 2 \times 2F + 0 \times 1F$$

**Algorithme (ébauche):**

Demander à l'utilisateur de fournir un montant.

Calculer le nombre de pièces de 20F, 10F, 5F, 2F et 1F correspondant à ce montant, de façon à obtenir le moins de pièces possibles.

Afficher ces informations en respectant le format décrit ci-dessus.

Remarque: la 2<sup>ème</sup> phrase de cet algorithme indique des calculs à réaliser qui ne sont pas exprimés en termes d'opérations primitives, il est donc nécessaire de préciser comment ces calculs doivent être faits.

## Bases de l'algorithmique et de la programmation en Java

En analysant des exemples traités "à la main", on s'aperçoit facilement que pour obtenir un nombre minimal de pièces il faut tout d'abord choisir un maximum de pièces de 20F, puis de 10F avec le montant restant, puis de 5F et ainsi de suite jusqu'aux pièces de 1F. L'ordre des opérations est donc ici fondamental pour obtenir le bon résultat.

Le nombre de pièces de 20F est obtenu simplement en divisant le montant initial par 20. Il convient alors de calculer le montant restant qui est le reste de la division précédente. On procède ensuite de façon identique à partir du montant restant pour les pièces de 10F, puis de 5F, puis de 2F; le nombre de pièces de 1F est le montant restant alors.

### Algorithme détaillé:

Demander à l'utilisateur de fournir un montant initial.  
Calculer le nombre de pièces de 20F en divisant le montant initial par 20.  
Calculer le montant restant qui est le reste de la division précédente.  
Calculer le nombre de pièces de 10F en divisant le montant restant par 10.  
Calculer le montant restant qui est le reste de la division précédente.  
Calculer le nombre de pièces de 5F en divisant le montant restant par 5.  
Calculer le montant restant qui est le reste de la division précédente.  
Calculer le nombre de pièces de 2F en divisant le montant restant par 2.  
Calculer le montant restant qui est le reste de la division précédente.  
Le nombre de pièces de 1F est égal au montant restant.  
Afficher ces informations en respectant le format décrit ci-dessus.

### Analyse de l'algorithme pour sa transformation en un programme (Java):

- Bilan des informations manipulées:
  - Le montant initial (nombre entier) à acquérir,
  - le nombre de pièces de 20F, le montant restant,
  - le nombre de pièces de 10F, le nombre de pièces de 5F,
  - le nombre de pièces de 2F, le nombre de pièces de 1FCeci permet de déclarer les variables correspondantes en Java.
  
- 1<sup>ère</sup> phrase de l'algorithme:  
il s'agit de l'acquisition d'un nombre entier positif ou nul, il convient donc de renseigner préalablement l'utilisateur par l'affichage d'un message.  
La traduction est immédiate.
  
- 2<sup>ème</sup> phrase de l'algorithme:  
il s'agit d'un calcul élémentaire, l'information à calculer est le nombre de pièces de 20, elle est obtenue en divisant le montant initial par 20, d'où:  
$$\text{nbPieces20F} = \text{montantInitial} / 20;$$
  
- 3<sup>ème</sup> phrase de l'algorithme:  
il s'agit d'un calcul élémentaire, l'information à calculer est le montant restant, il est obtenu en prenant le reste de la division du montant initial par 20, d'où:  
$$\text{montantRestant} = \text{montantInitial} \% 20;$$
  
etc.

**Programme Java résultant:**

```
class Td1Exo2
{
    public static void main (String args[])
    {
        int    montantInitial, montantRestant;
        int    nbPieces20F, nbPieces10F, nbPieces5F, nbPieces2F, nbPieces1F;

        System.out.println(" Donnez moi le nombre de francs");
        montantInitial    = Clavier.readInt();

        nbPieces20F       = montantInitial / 20;
        montantRestant    = montantInitial % 20;
        nbPieces10F       = montantRestant / 10;
        montantRestant    = montantRestant % 10;
        nbPieces5F        = montantRestant / 5;
        montantRestant    = montantRestant % 5;
        nbPieces2F        = montantRestant / 2;
        montantRestant    = montantRestant % 2;
        nbPieces1F        = montantRestant;

        System.out.println(montantInitial + "F = " +
                           nbPieces20F + "x20F + " +
                           nbPieces10F + "x10F + " +
                           nbPieces5F  + "x5F + " +
                           nbPieces2F  + "x2F + " +
                           nbPieces1F  + "x1F");
    }
}
```

Cette version en Java est la traduction fidèle “bête” et systématique de l’algorithme détaillé donné précédemment. Ce n’est évidemment ni la seule ni la meilleure solution. D’autres solutions existent. Le point de départ choisi ici consistait à calculer dans un premier temps tous les résultats puis à les afficher; une autre approche possible serait d’afficher au fur et à mesure les résultats calculés sans avoir à les “mémoriser”. Par exemple, dès lors que le nombre de pièces de 20 Francs est affiché, cette information n’est plus utile pour la suite des calculs; on peut donc afficher directement le résultat du calcul donnant cette information en se dispensant de la “variable” nommée nbPieces20F. Ceci conduit à l’écriture de la 2<sup>ème</sup> version (plus courte) donnée ci-dessous.

```
class Td1Exo2v2
{
    public static void main (String args[])
    {
        int    montantInitial, montantRestant;

        System.out.println(" Donnez moi le nombre de francs");
        montantInitial = Clavier.readInt();

        System.out.print(montantInitial + "F = ");
        System.out.print(montantInitial / 20 + "x20F + ");
        montantRestant  = montantInitial % 20;
        System.out.print(montantRestant / 10 + "x10F + ");
        montantRestant  = montantRestant % 10;
        System.out.print(montantRestant / 5 + "x5F + ");
        montantRestant  = montantRestant % 5;
        System.out.print(montantRestant / 2 + "x2F + ");
        montantRestant  = montantRestant % 2;
        System.out.println(montantRestant + "x1F");
    }
}
```