

## Travaux pratiques d'informatique N° 4

Le but de cette séance est de vous permettre d'appliquer vos connaissances concernant les structures de contrôle de flux (les instructions structurées de sélection et de répétition) et d'écrire, de manière indépendante, des programmes Java interactifs simples.

### Indications générales :

- *utiliser une mise en forme cohérente et suggestive dans la rédaction des fichiers sources **.java** ;*
- *faire des commentaires pertinents, sans charger inutilement les sources ;*
- *choisir des noms évocateurs pour les variables utilisées ;*
- *consulter l'**API Java** afin d'obtenir des informations concernant les packages et les classes prédéfinies ;*
- *afficher des messages et des résultats à l'écran (dans la fenêtre console) à l'aide des méthodes **print( )** et **println( )** appelées pour le champ public, statique et final **out** (de type **java.io.PrintStream**) de la classe **System** du package **java.lang** ;*
- *lire des informations introduites au clavier à l'aide des méthodes appropriées appelées pour un objet "filtre" ad-hoc de type **java.util.Scanner** associé au champ public, statique et final **in** (de type **java.io.InputStream**) de la classe **System** du package **java.lang**.*

1. Quels sont les résultats fournis par les programmes suivants :

a)

```
package cms_tp4;

import java.util.Scanner;

public class CP_TP4Exo1a
{
    public static void main(String args[ ])
    {
        Scanner scan = new Scanner(System.in);
        int i ;
        System.out.println("Introduire un nombre entier compris entre 1 et 5 !");
        i=scan.nextInt( );
        switch(i)
        {
            case 1 : System.out.println("Début !");
            case 2 :
            case 3 : System.out.println("Milieu !");
                    break;
        }
    }
}
```

```

        default : System.out.println("Par défaut !");
        case 4 :
        case 5 : System.out.println("Fin !");
    } //fin de l'instruction structurée (du bloc) switch
} //fin de la méthode main
} //fin de la classe principale CP_TP8Exo1a

```

Remarque : pour ce point a), on considère les cas où l'utilisateur introduit comme valeur 1, 2, 3, 4, 5, 100 et -11.

**b)**

```

public class CP_TP4Exo1b
{
    public static void main(String args[ ])
    {
        int n, m ;

        for(n=0, m=0; n<4; n++)
            m=m+n;
        System.out.println("Premier cas : n="+n+" et m="+m+".");

        for(n=0, m=10; m>7; m=m-++n);
        System.out.println("\n2-ème cas : n="+n+" et m="+m+".");

        for(n=0, m=0; n<3; n++, m=m+n,
            System.out.println("\n3-ème cas : n="+n+" et m="+m+"."))
        {
        };

        System.out.println("\n4-ème cas : n="+n+" et m="+m+".");
    } //fin de la méthode main
} //fin de la classe principale CP_TP8Exo1b

```

**c)**

```

public class CP_TP4Exo1c
{
    public static void main(String args[ ])
    {
        int p=1, q=1;

        while(p<6) p+=2; q++;
        System.out.println("Premier cas : p="+p+" et q="+q+".");

        p=1; q=1;
        while(p<6) {p+=2; q++;}

        System.out.println("\nDeuxième cas : p="+p+" et q="+q+".");
    } //fin de la méthode main
} //fin de la classe principale CP_TP8Exo1c

```

**2. Ecrire une application Java interactive qui respecte les consignes suivantes :**

- a)** on demande à l'utilisateur d'introduire au clavier une lettre (majuscule ou minuscule, mais sans accent ni cédille) ;
- b)** si l'utilisateur introduit une lettre (sans accent, ni cédille), on affiche d'abord un message qui précise si la lettre introduite est une majuscule ou une minuscule, ensuite un message final et le programme s'arrête ;
- c)** autrement (c'est-à-dire si l'utilisateur introduit un caractère qui n'est pas une lettre), on affiche le code Unicode du caractère introduit et un message d'erreur approprié et on revient au point **a)** (pas plus de deux fois !) ;
- d)** en outre, si la situation décrite au point **c)** intervient trois fois de suite, la dernière fois on annonce à l'utilisateur qu'il a fait trop de fautes, on affiche le même message final qu'au point **b)** et le programme s'arrête.

***Indication :** concevoir quatre solutions différentes (en utilisant, respectivement, une boucle **do ... while** qui ne contient pas d'instruction **break**, une boucle **while** qui contient des instructions **break**, une boucle **for** qui ne contient pas d'instruction **break** et, finalement, une boucle **for** qui contient des instructions **break**).*

A l'exécution, les résultats du programme doivent respecter la mise en forme présentée dans les exemples ci-dessous :

```
Introduire une lettre (sans accent ni cédille et suivie par
ENTER) :
P
La lettre P est une majuscule !

Au revoir !
```

```
Introduire une lettre (sans accent ni cédille et suivie par
ENTER) :
&
Le caractère & n'est pas une lettre !
Son code Unicode est 38.
Recommencez s.v.p. !

Introduire une lettre (sans accent ni cédille et suivie par
ENTER) :
p
La lettre p est une minuscule !

Au revoir !
```

```
Introduire une lettre (sans accent ni cédille et suivie par
ENTER) :
1
Le caractère 1 n'est pas une lettre !
Son code Unicode est 49.
Recommencez s.v.p. !

Introduire une lettre (sans accent ni cédille et suivie par
ENTER) :
ç
Le caractère ç n'est pas une lettre !
Son code Unicode est 231.
Recommencez s.v.p. !

Introduire une lettre (sans accent ni cédille et suivie par
ENTER) :
{
Le caractère { n'est pas une lettre !
Son code Unicode est 123.
Désolé ! Vous faites trop de fautes !

Au revoir !
```

### 3. Ecrire une **application Java interactive** qui :

- a. dans la partie "*introduction des données*", affiche un message de bienvenue, demande à l'utilisateur son nom et son prénom, ainsi que son année de naissance ;
- b. dans la partie "*traitement des données*", calcule l'âge *actuel* de l'utilisateur ;
- c. dans la partie "*présentation des résultats*", affiche l'âge *calculé* de l'utilisateur *courant*, ainsi qu'un message final.

Comme dans l'exercice numéro **3.** des Travaux pratiques d'informatique **N° 3**, le projet Java qui correspond à l'application détaillée ci-dessus doit prévoir **deux classes**, à savoir :

- une classe publique *Aux\_TP4Exo3*, munie d'une méthode publique et statique *calculerAge* ;
- la classe publique "principale" *CP\_TP4Exo3*, munie de la méthode *main* qui utilise (appelle) la méthode *calculerAge*.

De plus, si dans la partie "*introduction des données*", l'utilisateur introduit au clavier une année de naissance invalide (i.e. plus petite que 1910 ou plus grande que l'année courante), le programme affiche un message d'erreur et invite l'utilisateur à réintroduire une nouvelle valeur.

Cependant, si l'utilisateur se trompe trois fois de suite, le programme n'exécute plus les parties "*traitement des données*" et "*présentation des résultats*" mais affiche un message "d'erreur fatale" et se termine inopinément.

La sortie du programme doit respecter la mise en page donnée dans les exemples ci-dessous.

```
Bonjour !
Introduisez votre nom, svp :
Blanc
Introduisez votre prénom, svp :
Bonnet
Introduisez votre année de naissance, svp :
1998
Bonjour Bonnet Blanc !
Vous avez 20 ans !
Au revoir !
```

```
Bonjour !
Introduisez votre nom, svp :
Blanc
Introduisez votre prénom, svp :
Bonnet
Introduisez votre année de naissance, svp :
1890
Année de naissance non valide !
Introduisez votre année de naissance, svp :
2025
Année de naissance non valide !
Introduisez votre année naissance, svp :
1998
Bonjour Bonnet Blanc !
Vous avez 20 ans !
Au revoir !
```

```
Bonjour !
Introduisez votre nom, svp :
Blanc
Introduisez votre prénom, svp :
Bonnet
Introduisez votre année de naissance, svp :
1890
Année de naissance non valide !
Introduisez votre année de naissance, svp :
2025
Année de naissance non valide !
Introduisez votre année de naissance, svp :
-1998
Année de naissance non valide !
Désolé ! Vous avez fait trop de fautes ! Au revoir !
```