

Master bio-info : Java

Année 2010-2011

TP1

Exercice 1 Tapez le programme suivant dans un fichier appelé `Bonjour.java`.

```
public class Bonjour {  
    public static void main(String [] args) {  
        System.out.println("Bonjour !");  
    }  
}
```

Compilez-le et exécutez-le.

Rappels : Un programme doit toujours commencer par

```
public class MonProgramme
```

et doit être contenu dans un fichier qui doit avoir pour nom `MonProgramme.java`.
Le nom du fichier doit reprendre le nom donné dans la classe.

La méthode principale s'appelle `main` et doit être déclarée comme suit :

```
public static void main(String [] args)
```

Pour compiler le programme tapez sur le terminal :

```
>javac Programme.java
```

(">" représente l'invite d'Unix) : La compilation produit un fichier `Programme.class`, l'exécution du programme se fera avec la commande :

```
>java Programme
```

Comment faire une Lecture au clavier ? Voici un exemple de lecture au clavier :

```
import java.util.Scanner; //on va utiliser la classe Scanner
```

```
public class LireEntiers{  
    public static void main(String [] args) {  
  
        // initialisation lecture  
        Scanner sc = new Scanner(System.in);  
  
        System.out.print("Donner un entier");  
  
        //lecture de la reponse  
        int r = sc.nextInt();  
  
        System.out.println("Vous avez donné l'entier "+ r);  
    }  
}
```

La ligne

```
import java.util.Scanner;
```

doit être mise en début de fichier. La ligne

```
Scanner sc = new Scanner(System.in);
```

doit être mise dans la méthode qui va utiliser la lecture, elle sert à indiquer qu'on va lire au clavier.

Pour lire un `double`, on utilisera `sc.nextDouble()`, pour un `boolean`, `sc.nextBoolean()`, etc.

La lecture d'un mot se fera par `sc.next()`, le retour à la ligne ou un espace étant considérés comme une fin de mot.

Exercice 2 [boucle while] Écrivez un programme qui choisit deux nombres entre 1 et 9 et demande à l'utilisateur le résultat de leur multiplication, si la réponse est mauvaise le programme repose la question.

Exemple d'exécution :

```
> java TableMultiplication
Quel est le resultat de 2 X 6 ? 10
Faux !
Recommencez:2 X 6 = ? 18
Faux !
Recommencez:2 X 6 = ? 12
Bravo !!!
```

Aides :

- La méthode

```
public static double Math.random()
```

retourne une valeur pseudo-aléatoire supérieure à 0 et inférieure strictement à 1.
- Pour obtenir la partie entière d'un flottant, on utilisera le cast : `(int)unflottant`.

Exercice 3 [switch case] Écrivez un programme qui **affiche** au hasard une séquence d'ADN, la longueur de la séquence est donnée par l'utilisateur. On suppose que la probabilité de chaque lettre est identique.

Pour cela, vous écrirez d'abord une méthode

```
public static char baseADN()
```

qui tirera une lettre au hasard de la manière suivante : on engendrera un nombre au hasard entre 1 et 4, et à l'aide d'un `switch case`, on en déduira la lettre.

Exercice 4 [String] Modifiez le programme de l'exercice 3 de telle sorte que l'on ait une méthode

```
String sequenceADN(int longueur)
```

qui engendre une séquence aléatoire de longueur `longueur`. Cette méthode sera récursive.

On rappelle que l'on peut concaténer 2 chaînes de caractères, ou une chaîne et un caractère grâce à l'opérateur `+`.

lecture de la ligne de commande Lorsqu'on lance un programme java de la manière suivante :

```
>java Prog Il fait beau
```

Le tableau de `String` donné comme argument du `main` reçoit le tableau `{"Il", "fait", "beau"}`. (Ce tableau s'appelle `args` dans le programme de l'exercice 1.)

Exercice 5 [String] Écrivez un programme qui compte le nombre de voyelles ('a', 'e', 'i', 'o' ou 'u') contenues dans les chaînes de caractères données en argument de la ligne de commande.

Exemple d'exécution :

```
>java ComptVoyelle Il fait beau
Il y a 6 voyelles.
```

Aide : la longueur d'une chaîne `s` est donnée par `s.length()`. Les indices des caractères d'une chaîne commencent par 0. Pour obtenir le *i*ème caractère de la chaîne `s`, on écrira `s.charAt(i)`.

On peut utiliser un `switch case`.

Exercice 6 [String] Écrivez un programme qui réécrit les chaînes de caractères données en argument de la ligne de commande en les tronquant de leur première et dernière lettre.

Exemple d'exécution :

```
>java Tronque Il fait beau aujourd'hui
ai ea ujourd'hu
```

Aide : La méthode `s.substring(i,j)` retournera la sous-chaîne constituée des caractères de *i* à *j*-1.

Exercice 7 [String] (plus difficile) Reprenez la méthode `sequenceADN` de l'exercice 4. Créez une méthode

`String modification(String sequence)`

qui à partir d'une séquence d'ADN applique aléatoirement une des trois modifications suivantes :

1. suppression d'une base (probabilité 20%);
2. insertion d'une base (probabilité 30%);
3. remplacement d'une base (probabilité 50%);

L'indice de la base supprimée, insérée ou remplacée est choisi aléatoirement avec une égale probabilité et il en est de même pour la nouvelle base éventuelle.

Dans la méthode `main`, on demandera à l'utilisateur la longueur de la chaîne produite aléatoirement et le nombre de modifications aléatoires, on affichera la chaîne à chaque étape.