

Lire les entrées clavier

Après la lecture de ce chapitre, vous pourrez saisir des informations et les stocker dans des variables afin de pouvoir les utiliser a posteriori.

En fait, jusqu'à ce que nous voyions les interfaces graphiques, nous travaillerons en mode console.

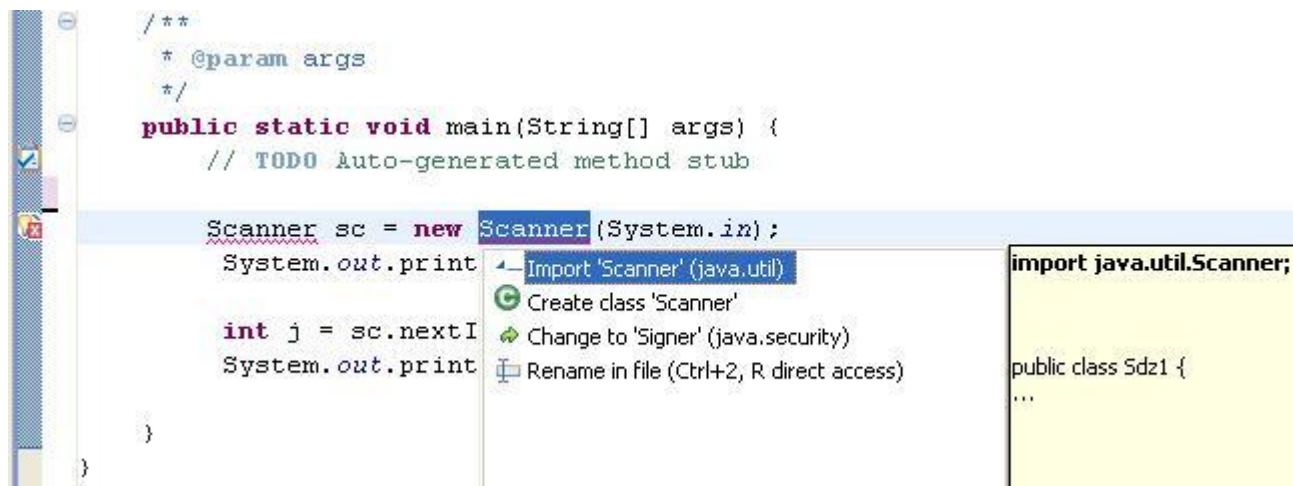
Donc, afin de rendre nos programmes plus ludiques, il est de bon ton de pouvoir interagir avec ceux-ci. Par contre, ceci peut engendrer des erreurs (on parlera d'exceptions, mais ce sera traité plus loin).

La classe Scanner

Je me doute qu'il vous tardait de pouvoir communiquer avec votre application... Le moment est enfin venu ! Mais je vous préviens, la méthode que je vais vous donner présente des failles. Je vous fais confiance pour ne pas rentrer n'importe quoi n'importe quand...

Je vous ai dit que vos variables de type String sont en réalité des objets de type String. Pour que Java puisse lire ce que vous tapez au clavier, vous allez devoir utiliser un objet de type Scanner. Cet objet peut prendre différents paramètres, mais ici nous n'en utiliserons qu'un : celui qui correspond à l'entrée standard en Java. Lorsque vous faites `System.out.println()`; je vous rappelle que vous appliquez la méthode `println()` sur la sortie standard; ici, nous allons utiliser l'entrée standard `System.in`. Donc, avant d'indiquer à Java qu'il faut lire ce que nous allons taper au clavier, nous devons instancier un objet Scanner. Avant de vous expliquer ceci, créez une nouvelle classe et tapez cette ligne de code dans votre méthode main :

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



Importer la classe Scanner

Maintenant, regardez au-dessus de la déclaration de votre classe, vous devriez voir cette ligne :

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

Voilà ce que nous avons fait. Je vous ai dit qu'il fallait indiquer à Java où se trouve la classe Scanner.

Pour faire ceci, nous devons importer la classe Scanner grâce à l'instruction **import**. La classe que nous voulons se trouve dans le package java.util. Un package est un ensemble de classes. En fait, c'est un ensemble de dossiers et de sous-dossiers contenant une ou plusieurs classes, mais nous verrons ceci plus en détail lorsque nous ferons nos propres packages. Les classes qui se trouvent dans les packages autres que java.lang (package automatiquement importé par Java, on y trouve entre autres la classe System) sont à importer à la main dans vos classes Java pour pouvoir vous en servir. La façon dont nous avons importé la classe java.util.Scanner dans Eclipse est très commode. Vous pouvez aussi le faire manuellement :

```
1 //Ceci importe la classe Scanner du package java.util
```

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

```
3 //Ceci importe toutes les classes du package java.util
```

```
4 import java.util.*;
```

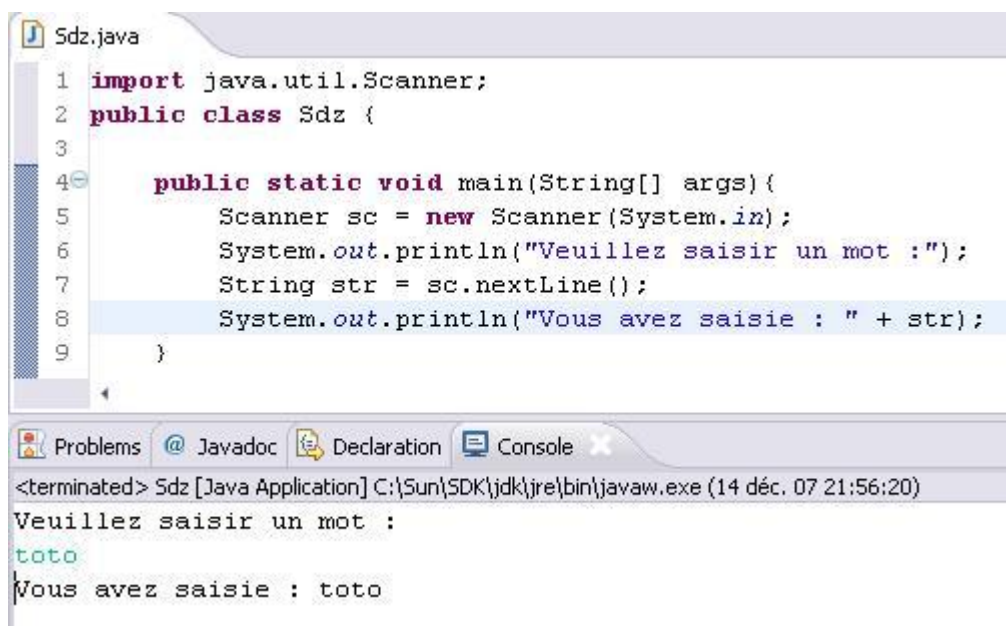
Récupérer ce que vous tapez

Voici l'instruction pour permettre à Java de récupérer ce que vous avez saisi pour ensuite l'afficher :

```
1 Scanner sc = new Scanner(System.in);
2 System.out.println("Veuillez saisir un mot :");
3 String str = sc.nextLine();
4 System.out.println("Vous avez saisi : " + str);
```

Une fois l'application lancée, le message que vous avez écrit auparavant s'affiche dans la console, en bas d'Eclipse.

Pensez à cliquer dans la console afin que ce que vous saisissez y soit écrit et que Java puisse récupérer ce que vous avez inscrit (figure suivante) !



The screenshot shows the Eclipse IDE with a Java file named 'Sdz.java' open. The code in the editor is as follows:

```
1 import java.util.Scanner;
2 public class Sdz {
3
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         System.out.println("Veuillez saisir un mot :");
7         String str = sc.nextLine();
8         System.out.println("Vous avez saisi : " + str);
9     }
```

Below the editor, the 'Console' window is visible, showing the output of the program:

```
<terminated> Sdz [Java Application] C:\Sun\SDK\jdk\jre\bin\javaw.exe (14 déc. 07 21:56:20)
Veuillez saisir un mot :
toto
Vous avez saisi : toto
```

The word 'Saisie' is written to the right of the console output.

utilisateur dans la console

Si vous remplacez la ligne de code qui récupère une chaîne de caractères comme suit :

```
1 Scanner sc = new Scanner(System.in);
2 System.out.println("Veuillez saisir un nombre :");
3 int str = sc.nextInt();
4 System.out.println("Vous avez saisi le nombre : " + str);
```

... vous devriez constater que lorsque vous introduisez votre variable de type Scanner et que vous introduisez le point permettant d'appeler des méthodes de l'objet, Eclipse vous propose

une liste de méthodes associées à cet objet (ceci s'appelle l'autocomplétion) ; de plus, lorsque vous commencez à taper le début de la méthode `nextInt()`, le choix se restreint jusqu'à ne laisser que cette seule méthode.

Exécutez et testez ce programme : vous verrez qu'il fonctionne à la perfection. Sauf... si vous saisissez autre chose qu'un nombre entier !

Vous savez maintenant que pour lire un `int`, vous devez utiliser `nextInt()`. De façon générale, dites-vous que pour récupérer un type de variable, il vous suffit d'appeler `next<Type de variable commençant par une majuscule>` (rappelez-vous de la convention de nommage Java).

```
1 Scanner sc = new Scanner(System.in);
2 int i = sc.nextInt();
3 double d = sc.nextDouble();
4 long l = sc.nextLong();
5 byte b = sc.nextByte();
6 //Etc.
```

Attention : il y a un type de variables primitives qui n'est pas pris en compte par la classe `Scanner` : il s'agit du type `char`. Voici comment on pourrait récupérer un caractère :

```
1 System.out.println("Saisissez une lettre :");
2 Scanner sc = new Scanner(System.in);
3 String str = sc.nextLine();
4 char caractere = str.charAt(0);
5 System.out.println("Vous avez saisi le caractère : " + caractere);
```

Qu'avons-nous fait ici ?

Nous avons récupéré une chaîne de caractères, puis utilisé une méthode de l'objet `String` (ici, `charAt(0)`) afin de récupérer le premier caractère saisi. Même si vous tapez une longue chaîne de caractères, l'instruction `charAt(0)` ne renverra que le premier caractère. Vous devez vous demander pourquoi `charAt(0)` et non `charAt(1)` : nous

aborderons ce point lorsque nous verrons les tableaux ... Jusqu'à ce qu'on aborde les exceptions, je vous demanderai d'être rigoureux et de faire attention à ce que vous attendez comme type de données afin d'utiliser la méthode correspondante.

Une précision s'impose, toutefois : la méthode `nextLine()` récupère le contenu de toute la ligne saisie et replace la « tête de lecture » au début d'une autre ligne. Par contre, si vous avez invoqué une méthode comme `nextInt()`, `nextDouble()` et que vous invoquez directement après la méthode `nextLine()`, celle-ci ne vous invitera pas à saisir une chaîne de caractères : elle videra la ligne commencée par les autres instructions. En effet, celles-ci ne repositionnent pas la tête de lecture, l'instruction `nextLine()` le fait à leur place. Pour faire simple, ceci :

```
1 import java.util.Scanner;
2
3 public class Main {
4     public static void main(String[] args){
5         Scanner sc = new Scanner(System.in);
6         System.out.println("Saisissez un entier : ");
7         int i = sc.nextInt();
8         System.out.println("Saisissez une chaîne : ");
9         String str = sc.nextLine();
10        System.out.println("FIN ! ");
11    }
12 }
```

... ne vous demandera pas de saisir une chaîne et affichera directement « Fin ». Pour pallier ce problème, il suffit de vider la ligne après les instructions ne le faisant pas automatiquement :

```
1 import java.util.Scanner;
2
3 public class Main {
```

```
4 public static void main(String[] args){
5     Scanner sc = new Scanner(System.in);
6     System.out.println("Saisissez un entier : ");
7     int i = sc.nextInt();
8     System.out.println("Saisissez une chaîne : ");
9     //On vide la ligne avant d'en lire une autre
10    sc.nextLine();
11    String str = sc.nextLine();
12    System.out.println("FIN ! ");
13 }
14 }
```