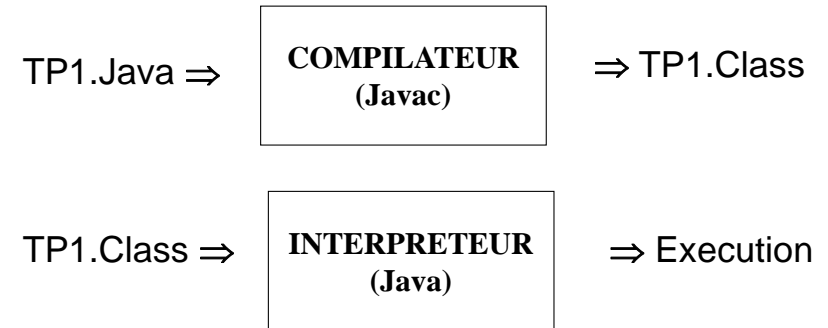

Elements de base du langage Java

Ludovic Liétard

Introduction

- La compilation génère du ByteCode (.Class)
- Le ByteCode est ensuite interprété



Le corps d'un programme Java

```
public class Essai
{
    public static void main (String[ ] args)
    {
        // ICI LE CORPS DU PROGRAMME
    }
}
```

Les Types Primitifs

- Les entiers : short, int, long
- Les réels : float, double
- Les booléens : boolean
- Les caractères : char

Les Tableaux

```
public class Essai
{
    public static void main (String[ ] args)
    { // Indice débute à 0
        // Déclaration de tableaux d'entiers
        int[ ] tableau1 = new int [10];
        int[ ] tableau2 = {1,2,3,4};
        int[ ] [ ] tableau3 = new int [10][10];
        int[ ] [ ] tableau4 = {{1,2,3,4},{5,6,7,8}};
    }
}
```

Les Opérateurs

- +, -, *, / (ceux de l'arithmétique)
- ++ (incrémentatation)
- -- (décrémentatation)
- >, <, <=, >=, ==, != (comparaison)
- && (et logique), || (ou logique)
- ! (négation)
- L'affectation se note =

Les Structures de Contrôle (1)

▪ Structures Conditionnelles

```
if (condition) {
    suite d'instructions;
}

if (condition) {
    suite d'instructions;
} else {
    suite d'instructions;
}
```

Les Structures de Contrôle (2)

▪ Choix multiple

```
switch (expr) {
    case val1 :
        suite d'instructions;
        break;
    case val2 :
        suite d'instructions;
        break;
    ...
    default:
        suite d'instructions;
}
```

Les Structures de Contrôle (3)

- **Structure Conditionnelle**

```
while (condition) {  
    suite d'instructions;  
}
```

- **Structure répétitive**

```
for( exp1; exp2; exp3){  
    suite d'instructions;  
}
```

Les Structures de Contrôle (4)

Pour la structure répétitive

- *exp1* est évaluée avant de débiter la boucle (classiquement, c'est l'initialisation)
- *exp2* est évaluée à chaque itération. La boucle se poursuit si l'évaluation délivre vraie (négation de la condition d'arrêt).
- *exp3* est évaluée en fin d'itération (classiquement, la progression).

Exemple

```
public class Essai  
{  
    public static void main (String[] args)  
    {  
        int indice;  
        char[] tab = new char[5];  
        // notation équivalente  
        //char tab[] = new char[5];  
  
        for (indice=0; indice<5; indice++){  
            tab[indice] = (char) ((int)'A'+indice);  
        }  
  
        for (indice=0; indice<5; indice++){  
            System.out.println(tab[indice]);  
        }  
    }  
}
```

Conclusion

- Syntaxe analogue au langage C
- Pas de procédures, ni de fonctions