

Un programme Java est structuré selon les trois principes fondamentaux suivants :

1. Un programme contient :
  - une fonction principale, appelée fonction `main` ;
  - un ensemble de fonctions définies par le programmeur ;
  - des instructions de déclaration de variables.
2. Les fonctions contiennent :
  - des instructions de déclaration de variables ;
  - des instructions élémentaires (affectation, test, répétition, etc.) ;
  - des appels à des fonctions, prédéfinies ou non.
3. Chaque fonction est comparable à une boîte noire, dont le contenu n'est pas visible en dehors de la fonction.

De ces trois propriétés découlent les notions suivantes :

- **Visibilité** : toute variable déclarée à l'intérieur d'une fonction n'est visible que dans cette fonction et ne peut être utilisée dans une autre fonction.
- **Variable locale** : toute variable déclarée à l'intérieur d'une fonction est une variable locale à cette fonction. Ces variables n'existent que le temps de l'exécution de la fonction, et elles ne sont pas modifiables depuis une autre fonction.
- **Variable de classe** : les variables déclarées en dehors de toute fonction sont appelées des variables de classe. Ces variables sont définies pour l'ensemble du programme, et elles sont visibles et modifiables par toutes les fonctions de la classe.

Lorsqu'une variable de classe et une variable locale portant le même nom coexistent à l'intérieur d'une fonction, la règle veut que ce soit la variable locale qui soit prise en compte et non la variable de classe.

Les fonctions sont des blocs d'instructions distinctes. Pour communiquer le contenu d'une case mémoire (variable) locale de l'une à une case mémoire locale de l'autre fonction, il est nécessaire d'utiliser les techniques suivantes :

- **Les paramètres d'entrée des fonctions** : lorsqu'une fonction communique le contenu d'une variable à une autre fonction par l'intermédiaire d'un paramètre, on dit que le paramètre est passé **par valeur**. Ce type de transmission de données ne permet pas de modifier, dans la fonction appelante, le contenu de la variable passée en paramètre.
- **Le retour de résultat en sortie** : pour garder en résultat la modification du contenu d'une variable en sortie de fonction, une technique consiste à retourner la valeur calculée par l'intermédiaire de l'instruction `return`.

Ces deux modes de communication ne permettent pas de récupérer plusieurs données modifiées à l'intérieur d'une fonction. Seul le concept d'objet, étudié au chapitre suivant, permet de réaliser cette opération.

## Exercices

### Repérer les variables locales et variables de classe

1.1. En observant le programme suivant :

```
import java.util.*;
public class Calculette {
    public static double resultat ;

    public static void main( String []
        argument) {
        int a, b;
        Scanner Lire = new Scanner(System.in);
        menu();
        System.out.println("Entrer la première
            valeur ");
        a = Lire.nextInt();
        System.out.println("Entrer la seconde
            valeur ");
        b = Lire.nextInt();
        calculer();
        afficher();
    }

    public static void calculer() {
        char operation ;

        switch (operation) {
            case '+' : resultat = a + b ;
                break ;
            case '-' : resultat = a - b ;
                break ;
            case '/' : resultat = a / b ;
                break ;
            case '*' : resultat = a * b ;
                break ;
        }

        public static void afficher() {
            char operation ;
            System.out.print(a + " ") ;
            System.out.print(operation) ;
            System.out.println(" + b + " = " +
                resultat);
        }
    }
}
```

```

public static void menu() {
    Scanner Lire = new Scanner(System.in);
    char operation ;
    System.out.println("Je sais compter, entrez
        l'operation choisie") ;
    System.out.println(" + pour additionner ") ;
    System.out.println(" - pour soustraire ") ;
}

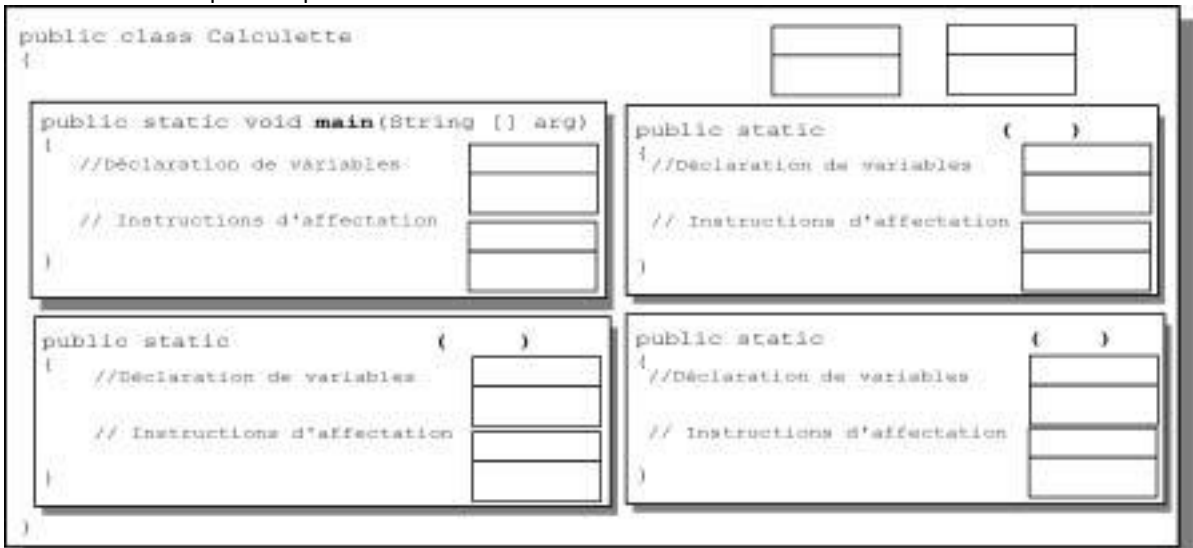
```

```

System.out.println(" * pour multiplier ") ;
System.out.println(" / pour diviser ") ;
System.out.println(" (+, -, *, /) ? : ") ;
operation = Lire.next().charAt(0);
}
}

```

- Rechercher les différentes fonctions définies dans la classe `Calcullette`.
- À l'aide du schéma ci-dessous, placer les variables dans les blocs où elles sont déclarées.
- À l'aide du schéma, déterminer les variables locales à chacune des fonctions, ainsi que les variables de classe.
- Après exécution de la fonction `menu` et lecture des deux valeurs numériques `a` et `b`, la fonction `calculer` peut-elle réaliser l'opération demandée ? Pourquoi ?
- Même question pour la fonction `afficher`.



### Communiquer des valeurs à l'appel d'une fonction

- Afin de corriger le programme `Calcullette`, nous supposons que les variables `résultat` et `opération` sont déclarées en tant que variables de classe et non plus localement aux fonctions `afficher` et `menu`.
  - Modifier le schéma réalisé en 1.1.b, en tenant compte de ces nouvelles déclarations.
  - Quelle technique doit on utiliser pour que les fonctions `calculer` et `afficher` connaissent le contenu des variables `a` et `b`, afin d'effectuer ensuite les instructions qui les composent ?
  - Ecrire les fonctions en utilisant cette technique.

### Transmettre le résultat à la fonction appelante

- Nous supposons que le programme `Calcullette` ne contient plus de variables de classe. Les variables `résultat` et `opération` sont maintenant déclarées localement aux fonctions qui l'utilisent.
  - Quelles sont les conséquences de cette nouvelle hypothèse, sur le résultat du programme ?
  - Comment la fonction `calculer` peut-elle connaître l'opérateur choisi par l'utilisateur, dans la fonction `menu` ?
  - Transformer la fonction `menu` de façon à ce que l'opérateur soit transmis à la fonction `main`.
  - Modifier la fonction `calculer` de façon à lui transmettre l'opérateur fourni par la fonction `menu`.
  - Comment la fonction `afficher` peut-elle connaître le résultat de la fonction `calculer` ?
  - Transformer la fonction `calculer` de façon à ce que le résultat soit transmis à la fonction `main`.
  - Modifier la fonction `afficher` de façon à lui transmettre le résultat fourni par la fonction `calculer`.