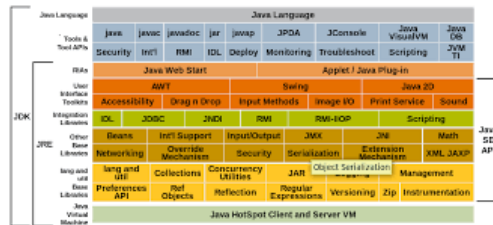


La Base du Langage de Programmation Java



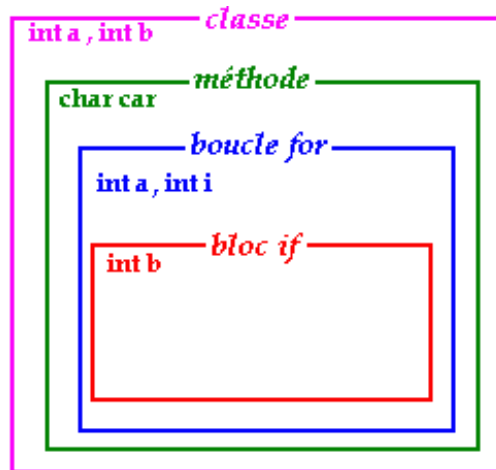
Java est un langage de programmation orienté objet, largement utilisé pour le développement d'applications mobiles, web et d'entreprise. Voici les bases essentielles de Java :

1. Installation de Java



- **JDK (Java Development Kit)** : Comprend le JRE (Java Runtime Environment) et des outils de développement comme le compilateur Java.
- **IDE (Integrated Development Environment)** : Des outils comme IntelliJ IDEA, Eclipse, ou NetBeans facilitent le développement.

2. Structure de base d'un programme Java



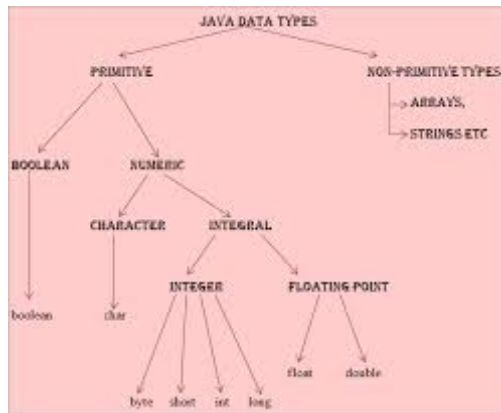
Un programme Java typique se compose de classes et de méthodes. Voici un exemple de base :

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

2. Structure de base d'un programme Java

- `public class Main` : Déclare une classe nommée `Main`.
- `public static void main(String[] args)` : Point d'entrée du programme. Chaque programme Java commence par la méthode `main`.
- `System.out.println("Bonjour, monde !")` : Imprime le texte "Bonjour, monde !" à la console.

3. Syntaxe de base



- **Variables** : Déclaration et initialisation.

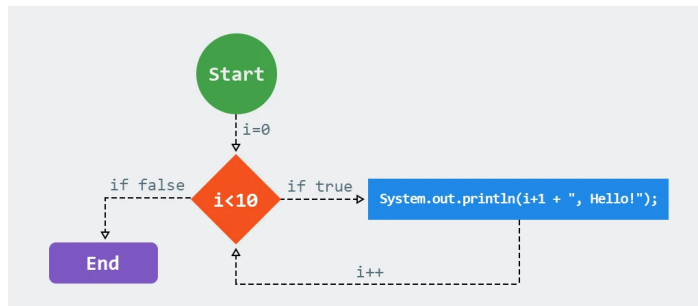
```
int nombre = 10;
String texte = "Bonjour";
```

- **Types de données** :

- **Primitifs** : int, float, double, char, boolean, etc.
- **Objets** : String, ArrayList, etc.

- **Opérateurs** : +, -, *, /, %, ++, --, ==, !=, etc.

4. Contrôle de flux



- **Conditions** : if, else if, else

```
if (nombre > 0) {
    System.out.println("Nombre positif");
} else {
```

```

        System.out.println("Nombre négatif ou zéro");
    }

```

- **Boucles :** for, while, do-while

```

for (int i = 0; i < 5; i++) {
    System.out.println(i);
}

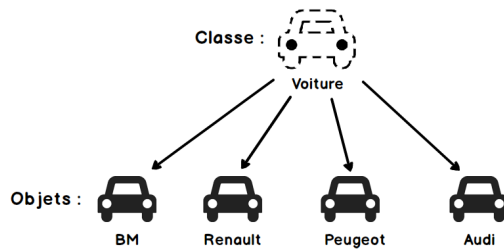
```

```

int j = 0;
while (j < 5) {
    System.out.println(j);
    j++;
}

```

5. Classes et objets



- **Déclaration de classe :**

```

public class Personne {
    String nom;
    int age;

    // Constructeur
    public Personne(String nom, int age) {
        this.nom = nom;
        this.age = age;
    }

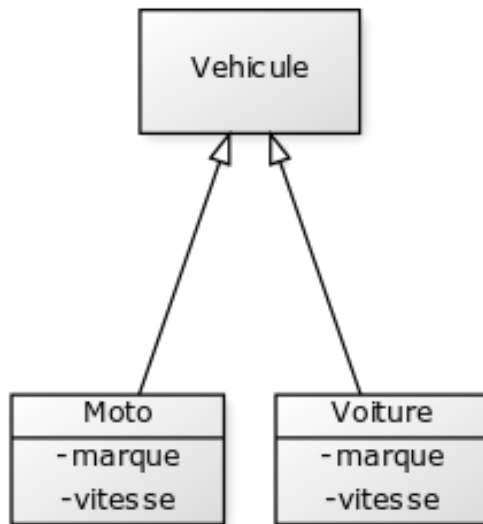
    // Méthode
    public void afficherInfo() {
        System.out.println("Nom:" + nom + ",Age:" + age);
    }
}

```

- Création d'objets :

```
Personne personne = new Personne("Alice", 30);
personne.afficherInfo();
```

6. Héritage



Java permet la création de nouvelles classes basées sur des classes existantes via l'héritage.

```
public class Animal {
    void manger() {
        System.out.println("L'animal mange");
    }
}

public class Chien extends Animal {
    void aboyer() {
        System.out.println("Le chien aboie");
    }
}

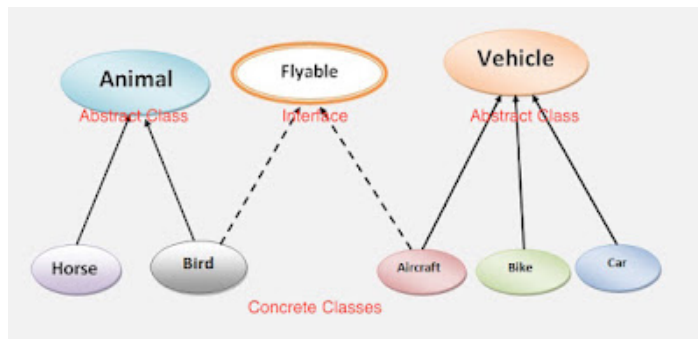
public class Test {
    public static void main(String[] args) {
        Chien chien = new Chien();
    }
}
```

```

        chien.manger();
        chien.aboyer();
    }
}

```

7. Interfaces et classes abstraites



- **Interface** : Une collection de méthodes abstraites.

```

interface Voler {
    void voler();
}

class Oiseau implements Voler {
    public void voler() {
        System.out.println("L'oiseau vole");
    }
}

```

- **Classe abstraite** : Peut avoir des méthodes abstraites et concrètes.

```

abstract class Animal {
    abstract void faireDuBruit();

    void dormir() {
        System.out.println("L'animal dort");
    }
}

class Chat extends Animal {
    void faireDuBruit() {
        System.out.println("Le chat miaule");
    }
}

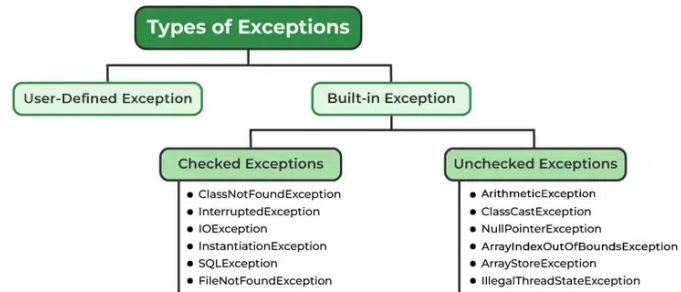
```

```

    }
}

```

8. Gestion des exceptions



Les exceptions sont des événements qui perturbent le flux normal de l'exécution des programmes.

```

try {
    int division = 10 / 0;
} catch (ArithmeticException e) {
    System.out.println("Erreur : Division par zéro");
} finally {
    System.out.println("Bloc finally exécuté");
}

```

Conclusion

Ces concepts de base constituent le fondement de la programmation en Java. En les maîtrisant, vous serez en mesure de comprendre et de développer des programmes Java simples. Au fur et à mesure de votre apprentissage, vous pourrez explorer des concepts plus avancés tels que les collections, les flux, la programmation multithread, et les API Java.