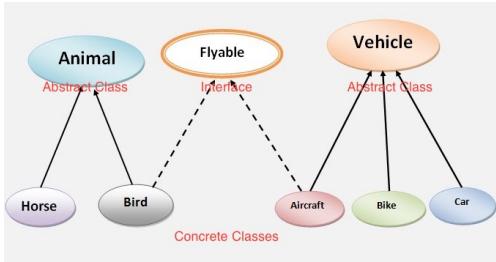
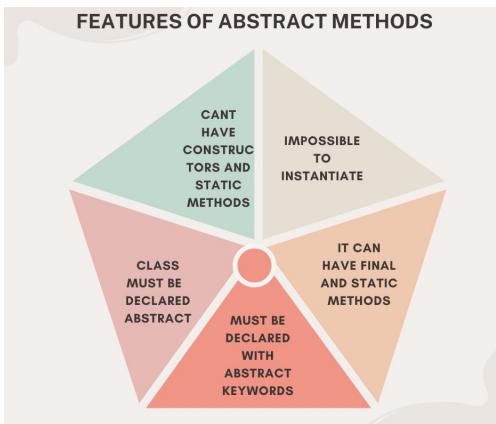


La Classe Abstraite



En Java, une **classe abstraite** est une classe qui ne peut pas être instanciée directement. Elle sert de modèle pour d'autres classes et peut contenir des méthodes abstraites, qui sont déclarées sans implémentation, ainsi que des méthodes concrètes, qui ont une implémentation. Les classes qui héritent de cette classe abstraite doivent fournir une implémentation pour toutes les méthodes abstraites qu'elle contient, sauf si elles sont elles-mêmes abstraites.

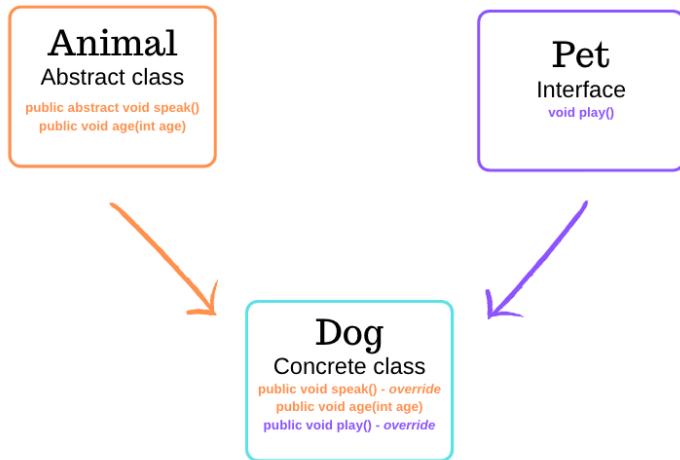
Caractéristiques principales d'une classe abstraite :



- Déclaration** : On utilise le mot-clé `abstract` pour déclarer une classe abstraite.
- Méthodes abstraites** : Une méthode abstraite n'a pas de corps, c'est-à-dire qu'elle ne contient pas de code implémenté.
- Héritage** : Une classe abstraite peut avoir des méthodes concrètes (méthodes avec une implémentation). Une classe qui hérite d'une classe abstraite doit implémenter toutes les méthodes abstraites, sauf si elle est elle-même abstraite.

4. **Instance** : On ne peut pas créer une instance d'une classe abstraite directement.

Exemple de code :



```

abstract class Animal {
    // Méthode abstraite (sans implémentation)
    abstract void makeSound();

    // Méthode concrète (avec implémentation)
    void sleep() {
        System.out.println("The animal is sleeping");
    }
}

class Dog extends Animal {
    // Implémentation de la méthode abstraite
    @Override
    void makeSound() {
        System.out.println("Woof Woof");
    }
}

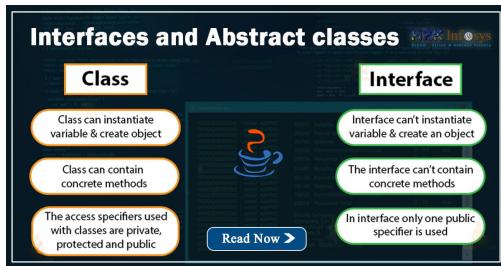
public class Main {
    public static void main(String[] args) {
        // Animal animal = new Animal(); // Erreur: impossible de créer une instance
    }
}
  
```

```

        Dog dog = new Dog();
        dog.makeSound(); // Affiche "Woof Woof"
        dog.sleep();    // Affiche "The animal is sleeping"
    }
}

```

Explications :



- **Animal** est une classe abstraite qui contient une méthode abstraite `makeSound()` et une méthode concrète `sleep()`.
- **Dog** est une classe qui hérite de la classe abstraite **Animal** et fournit une implémentation pour la méthode abstraite `makeSound()`.
- Dans la méthode `main`, une instance de **Dog** est créée, et les méthodes `makeSound()` et `sleep()` sont appelées.

Conclusion

Les classes abstraites sont utilisées pour fournir un modèle de base que d'autres classes peuvent partager, tout en laissant les détails spécifiques à chaque sous-classe.