Abstract class

Wednesday, September 10, 2025 11:35 AM

Definition of an Abstract Class in Java

An **abstract class** in Java is a class declared with the abstract keyword. It **cannot be instantiated directly** (you can't create objects of it). It is meant to serve as a **base or blueprint** for other classes.
It may contain:

- **Abstract methods** (methods without a body) — these must be implemented by subclasses.
- **Concrete methods** (methods with a body) — these can be inherited as-is or overridden by subclasses.
- **Fields, constructors, and static methods** just like a normal class.
-

```java
// abstract class
abstract class Human{  2 usages  2 inheritors  new *

    // variables
    String name;  2 usages
    int age ;  2 usages
    String gender ;  3 usages

    // constructor
    public Human(String name , int age){  2 usages  new *
        this.name = name ;
        this.age = age ;
    }

    // abstract method
    abstract public String favTask() ;  1 usage  2 implementations  new *

    // concrete method
    @Override  new *
    public String toString() {
        return name + " is a " + gender + " of " + age + " loves to " + favTask() ;
    }

}
```

- Sub class with implementation of the abstract class

```java
// concrete class
class Male extends Human{  2 usages  new *

    // initializer
    {
        gender = "Male" ;
    }

    // constructor
    public Male(String name , int age){  1 usage  new *
        super(name , age);
    }

    // implementation of abstract class
    public String favTask(){  1 usage  new *
        return "play basketball";
    }

}
```

Notes :

1. Abstract method must be defined inside an abstract class itself
2. Abstract may contain both abstract and concrete methods
3. Variables cannot be an abstract