

Abstract keyword in java

In Java, abstract is a non-access modifier in java applicable for classes, and methods but not variables.

Characteristics of Java abstract Keyword

In Java, the abstract keyword is used to define abstract classes and methods. Here are some of its key characteristics:

- **Abstract classes cannot be instantiated:** An abstract class is a class that cannot be instantiated directly. Instead, it is meant to be extended by other classes, which can provide concrete implementations of its abstract methods.
- **Abstract methods do not have a body:** An abstract method is a method that does not have an implementation. It is declared using the abstract keyword and ends with a semicolon instead of a method body. Subclasses of an abstract class must provide a concrete implementation of all abstract methods defined in the parent class.
- **Abstract classes can have both abstract and concrete methods:** Abstract classes can contain both abstract and concrete methods. Concrete methods are implemented in the abstract class itself and can be used by both the abstract class and its subclasses.
- **Abstract classes can have constructors:** Abstract classes can have constructors, which are used to initialize instance variables and perform other initialization tasks. However, because abstract classes cannot be instantiated directly, their constructors are typically called constructors in concrete subclasses.
- **Abstract classes can contain instance variables:** Abstract classes can contain instance variables, which can be used by both the abstract class and its subclasses. Subclasses can access these variables directly, just like any other instance variables.
- **Abstract classes can implement interfaces:** Abstract classes can implement interfaces, which define a set of methods that must be implemented by any class that implements the interface. In this case, the abstract class must provide concrete implementations of all methods defined in the interface.

Example program

```
// abstract class

abstract class Example {

    public String fname = "John";

    public int age = 24;

    public abstract void study(); // abstract method

}

class Student extends Example {

    public int graduationYear = 2018;

    public void study() { // the body of the abstract method is provided here

        System.out.println("Studying all day long");

    }

}

class Second {

    public static void main(String[] args) {

        // create an object of the Student class (which inherits attributes and methods from Example)

        Student myObj = new Student();

        System.out.println("Name: " + myObj.fname);

        System.out.println("Age: " + myObj.age);

        System.out.println("Graduation Year: " + myObj.graduationYear);

        myObj.study(); // call abstract method

    }

}
```