Scientech | Easy .com

Menu

12 Difference between Abstract Class and Interface in Java

In the previous tutorial, you may have noticed that abstract classes and interfaces have some similarities. For example, Both are used to achieving abstraction in Java.

Both define abstract methods that do not require an implementation. They are also similar in the sense that we cannot create an instance of both abstract class and interface.

But there are also many differences between them. So, let's discuss the difference between abstract class and interface in Java.

Abstract Class vs Interface in Java

1. Keyword(s) used:

a. Two keywords abstract and class are used to define an abstract class.
b. Only one keyword interface is used to define an interface.

2. Keyword used by implementing class:

a. To inherit the abstract class, we use the extends keyword.
b. To implement an interface, we can use the implements keyword.

3. Variables:

a. Abstract class can have final, non-final, static, and non-static variables.

b. Interface cannot have any instance variables. It can have only static variables.

---

4. Initialization:

a. The abstract class variable does not require performing initialization at the time of declaration.

For example:

```
public abstract class A {
 int x; //  No error.
}
```

b. Interface variable must be initialized at the time of declaration otherwise we will get compile-time error.

For example:

```
interface A {
int x; // Compile time error because the blank final field x may not have been initialized.
}
```

5. Method:

a. Every method present inside an interface is always public and abstract, whether we are declaring or not. That's why interface is also known as a pure (100%) abstract class.

b. An abstract class can have both abstract and non-abstract (concrete) methods.

---

6. Constructors:

a. Inside an interface, we cannot declare/define a constructor because the purpose of constructor is to perform initialization of instance variable but inside interface every variable is always static.

Therefore, inside the interface, the constructor concept is not applicable and does not require.

b. Since an abstract class can have instance variables. Therefore, we can define constructors within the abstract class to initialize instance variables.

7. Static and Instance blocks:

a. We cannot declare instance and static blocks inside an interface. If you declare them, you will get compile time error.

b. We can declare instance and static blocks inside abstract class.

---

8. Access modifiers:

a. We cannot define any private or protected members in an interface. All members are public by default.

b. There is no restriction in declaring private or protected members inside an abstract class.

9. Single vs Multiple inheritance:

a. A class can extend only one class (which can be either abstract or concrete class).

b. A class can implement any number of interfaces.

10. Default Implementation:

a. An abstract class can provide a default implementation of a method. So, subclasses of an abstract class can just use that definition but subclasses cannot define that method.

b. An interface can only declare a method. All classes implementing interface must define that method.

11. Difficulty in making changes:

a. It is easy to make changes to the implementation of the abstract class. For example, we can add a method with default implementation and the existing subclass cannot define it.

b. It is difficult to make changes in an interface if many classes already implementing that interface. For example, suppose you declare a new method in interface, all classes implementing that interface will stop compiling because they do not define that method.

12. Uses:

a. If you know nothing about the implementation. You have just requirement specification then you should go to use interface.

b. If you know about implementation but not completely (i.e. partial implementation) then you should go for using abstract class.

Java Abstract Class Example Program

Let's take an example program where we will deal with all the above points of abstract class in java.

Programs code 1:

```
package com.abstractProgram;
public abstract class AbstractClass { // Two keywords used: Abstract & class.

// Declaration of final, non-final, static, and instance variables.
   int a; // Not require initialization.
   final int b = 20; // Final variable.
   static int c = 30; // static variable.
```

```
// Declaration of abstract and non-abstract methods.
   abstract void m1();
   static void m2()
   {
      System.out.println("Static method in abstract class");
   }
// Default implementation of instance method.
   void m3() { // Concrete method.
     System.out.println("Instance method in abstract class");
    }
// Declaration of constructors to initialization of instance variable.
    AbstractClass()
    {
      int a = 10;
      System.out.println("Value of a; "+a);
    }
// Declaration of static and non-static blocks.
   static {
      System.out.println("Static block in abstract class");
   }
  {
    System.out.println("Instance block in abstract class");
  }
// Declaration of private & protected members.
   private void m4()
   {
      System.out.println("Private method");
    }
   protected void m5()
   {
     System.out.println("Protected method");
   }
```

```
}
public class A extends AbstractClass
{
  void m1()
  {
     System.out.println("Implementation of abstract method");
  }
}
public class AbstractTest
{
  public static void main(String[] args)
  {
    A a = new A();
    System.out.println("Value of b: " +a.b);
    System.out.println("Value of c: " +AbstractClass.c);
     a.m1();
    AbstractClass.m2();
     a.m3();
     a.m5();
   }
}
```

```
Output:
      Static block in abstract class
      Instance block in abstract class
      Value of a: 10
      Value of b: 20
      Value of c: 30
      Implementation of abstract method
      Static method in abstract class
```

```
    Instance method in abstract class
    Protected method
```

Java Interface Example Program

Let's take a simple example program to understand all the important points of the interface.

Program code 2:

```
package interfaceProgram;
public interface AA { // One keyword: interface.
  int x = 20; // Interface variable must be initialized at the time of declaration. By default, interface variable is public, static, and final.
  void m1(); // By default, interface method is public and static.

// Here, we cannot declare instance variables, instance methods, constructors, static, and non-static block.
  }
public interface BB
{
  int y = 20;
  void m2();
 }
public class CC implements AA, BB { // Multiple Inheritance.

 public void m1()
 {
   System.out.println("Value of x: " +x);
   System.out.println("m1 method");
  }
public void m2()
{
```

```
    System.out.println("Value of y: " +y);
    System.out.println("m2 method");
  }
}
public class MyClass
{
  public static void main(String [] args)
  {
    CC c = new CC();
     c.m1();
     c.m2();
  }
}
```

```
Output:
      Value of x: 20
      m1 method
      Value of y: 20
      m2 method
```

In this tutorial, you have learned about the **difference between abstract class and interface in java** with example programs. Hope that you will have understood the all basic points related to difference between abstract class and interface and practiced all programs.

In the next tutorial, we will learn about the difference between class and interface in Java. Thanks for reading!!!

> ***Next ⇒ Difference between Class and Interface in Java***

⇐ Prev Next ⇒

Please share your love

Core Java

Abstract Class vs Interface in Java

Java Tutorials

- Java Introduction +
- Basics of Java +
- Java Class and Object +
- Java Data types and Variables +
- Java Operators +
- Decision Making, Branching, Looping +
- Java Packages +
- Java Methods +
- Java Constructor +
- Java Modifiers +
- Blocks in Java +
- Java Static and Final Keywords +
- Inner Classes in Java +
- OOPs Concepts in Java +

ScientechEasy.com

ScientechEasy.com is optimized for learning various technologies step by step for beginners and professionals. You will get the best realtime coding examples that will simplify to improve reading and basic understanding. When using this site, you will agree to have read and accepted our terms of use and privacy policy.

Popular Tutorials

Learn Computer

Learn Java

Learn Selenium

Learn Python

Learn HTML

Learn JavaScript

Learn Computer GK

Follow us on:

Contact Us:

Whatsapp: 7979027618

Contact: contact@scientecheasy.com

Address: 4A, Shiv Shakti Complex, Joraphatak Road, Dhanbad.