Difference Between Class and Interface

www.differencebetween.com

Key Difference - Class vs Interface

Object-Oriented Programming(OOP) is a common paradigm in software development. It helps to bring real-world scenarios to programming using classes and objects. A class is a blueprint to create an object. The programmer can create a class with properties and methods. A student, teacher, are objects. Creating an object is known as instantiation. OOP also uses interfaces. Interfaces and class may seem similar, but they have a difference. This article discusses the difference between a class and an interface. The **key difference** between a class and an interface is that **a class is a reference type that is a blueprint to instantiate an object while the interface is a reference type that cannot be used to insatiate an object.**

What is Class?

In OOP, everything is considered as an object. It is not possible to create an object without a class. A class is a blueprint to create an object. When building a house, the architect draws the plan. The plan is similar to a class. The house is similar to the object. The class is the plan to build an object. An object is what is created using the class.

The class contains the properties and methods. A student can have properties such as name, grade, index number. A student can have methods such as reading, walking, studying. A class is created with the necessary properties and methods.

The syntax for creating a class in many programming languages is as follows. It is created using the keyword class.

```
class class_name {

// properties

//methods

}
```

Programming languages such as C#, Java follow a similar syntax to create an object using a class. Assume that the class name is Student.

```
Student s1= new Student ();
```

This s1 is the object. The "new" keyword is used to allocate memory for the properties. A class also has a constructor to initialize properties or variables.

Class members such as properties and methods have access modifiers. Access specifiers describe the accessibility and visibility of those members to other classes. Members of the class can have access specifiers such as public, private and protected. Public members are accessible by other classes. Private members are only accessible to the class. Protected members are accessible within the class and relevant subclasses.

What is Interface?

Abstraction is a pillar of Object Oriented programming. It is to hide the implementation details and to display the functionality to the user. Abstraction is achieved using abstract classes and interfaces. An abstract method does not have an implementation. A class that contains at least one abstract method is called an abstract class.

When there are two abstract classes, the methods declared in those classes should be implemented. A new class is used to implement those methods. If both classes had the same method, it might cause an ambiguity problem. Therefore, programming languages such as Java, C# has an interface.

Interfaces contain only the declaration of methods. There is no method implementation. Also, interfaces cannot be used to create objects. They are used to support multiple inheritances and to secure the code.

The syntax of Interface is as follows. Interface use the keyword "interface".

```
interface interface_name{

type method1(parameter_list);

type method2(parameter_list);

}
```

According to above, interfaces only have the declaration. There is no definition. So, interfaces cannot insatiate objects. It only gives an abstract view of what the interface is. Methods declared in the interface, can be implemented by one or many classes. A class use the keyword "implement" to implement an interface. Refer below example written using Java.

```
public class HelloWorld
{
  public static void main(String[] args)
  {
    C obj = new C();
    obj.sum();
    obj.sub();
  }
}

interface A{

  void sum();
}

interface B{

  void sub();

}

class C implements A,B{

  public void sum(){
    System.out.println("summation");
  }

  public void sub(){
    System.out.println("subtraction");
  }

}
```

Figure 01: Program using Interfaces

According to the above program, A and B are interfaces. Interface A has a method declaration which is the sum(). Interface B has a method declaration sub(). Class C is implementing both interfaces which are A and B. Therefore; class C define both sum() and sub() methods. After creating the object of type C, it is possible to call both methods sum() and sub().

Methods declared inside the interface must always be public because the implementing classes define them. An interface can also inherit from another interface.

What are the Similarities Between Class and Interface?

- Both are reference types.
- Both relate to Object-Oriented Programming.

What is the Difference Between Class and Interface?

Class vs Interface	
A class is a reference type that is a blueprint to create an object.	An interface is a reference type that cannot be instantiated.
Object Instantiation	
A class is used to insatiate an object.	An interface cannot be instantiated because the methods are unable to perform any action.

Constructor	
A class contain a constructor, to initialize the variables.	An interface does not contain a constructor because they are hardly any variables to initialize.
Keyword	
A class uses the keyword "class".	An interface uses the keyword "interface".
Access Specifier	
Members of the class can be private, public and protected.	Members of the interface should be always public because the implementing classes define them.

Summary - Class vs Interface

Classes and Interfaces are widely used in Object Oriented Programming. The difference between a class and an interface is that a class is a reference type which is a blueprint to instantiate an object and interface is a reference type which cannot be used to insatiate an object. A class can implement many interfaces. But it can only extend one superclass. In interface can inherit many interfaces but there cannot be an implementation. Both have their importance. The programmer can use them according to developing software.

Reference:

1.tutorialspoint.com. "Java Object and Classes.", The Point. Available here
2.navinreddy20. The interface in java with an example, Java By Navin Reddy, 11 Dec. 2012. Available here

How to Cite this Article?

APA: Difference Between Class and Interface.(2018 January 15). Retrieved (date), from http://differencebetween.com/difference-between-class-and-vs-interface/

MLA: "Difference Between Class and Interface" Difference Between.Com. 15 January 2018. Web.

Chicago: "Difference Between Class and Interface." Difference Between.Com. http://differencebetween.com/difference-between-class-and-vs-interface/ accessed (accessed [date]).