

Differences between class, abstract class, interface

Class

We use classes as object factories. A class defines a blueprint of what an object should look like and act like and then implements that blueprint by initialising class properties and defining methods. Therefore, when we create an instance of the class, we get an object that has actionable functions and defined properties. Let's look at an example of defining a class named **PizzaMaker**:

```
class PizzaMaker {  
  create(event: { name: string; toppings: string[] }) {  
    return { name: event.name, toppings: event.toppings };  
  }  
}  
  
const pizzaMaker = new PizzaMaker();  
  
const pizza = pizzaMaker.create({  
  name: 'Inferno',  
  toppings: ['cheese', 'peppers'],  
});  
  
console.log(pizza);  
// Output: { name: 'Inferno', toppings: [ 'cheese', 'peppers' ] }
```

Abstract classes

A class that has the abstract keyword in its declaration is called an abstract class. Abstract classes should have zero or more abstract methods. i.e., methods without a body. It can have multiple concrete methods.

Abstract classes allow you to create blueprints for concrete classes. But the inheriting class should implement the abstract method.

Abstract classes cannot be instantiated.

```
abstract class sunstar {  
    abstract void printInfo();  
}  
  
class employee extends sunstar {  
    void printInfo() {  
        String name = "Ohoud";  
        int age = 25;  
        float salary = 222.2F;  
  
        System.out.println(name);  
        System.out.println(age);  
        System.out.println(salary);  
    }  
}  
  
class base {  
    public static void main(String args[]) {  
        sunstar s = new employee();  
        s.printInfo();  
    }  
}
```

Output

Ohoud

25

222.2

Interface:

The interface is a blueprint that can be used to implement a class. The interface does not contain any concrete methods (methods that have code). All the methods of an interface are abstract methods.

An interface cannot be instantiated. However, classes that implement interfaces can be instantiated. Interfaces never contain instance variables but, they can contain public static final variables (i.e., constant class variables).

```
interface student
{
    void printInfo();
}
class avi implements student
{
    public void printInfo()
    {
        String name= "Ohoud";
        int age=25;
        System.out.println(name);
        System.out.println(age);
    }
}
class interfacesss
{
    public static void main (String args[])
    {
        avi s = new avi();
        s.printInfo();
    }
}
```

Output

Ohoud

25

References:

<https://www.geeksforgeeks.org/difference-between-abstract-class-and-interface-in-java/>

<https://www.guru99.com/interface-vs-abstract-class-java.html>