

Java & OOP

11. Interface By Example

Principles

- Single responsibility Principle
 - Do one thing and do it right
- Dependency Inversion Principle
 - Abstraction should not depend on implementation should be otherway

interface

```
class <some-class> [extends <one-class>]
    [implements <interface1 [, interface2, ...]>] {

}
```

rules

- Though interface defines a type an object cannot be created using new keyword

```
SomeInterface s = new SomeInterface (); // Not allowed
```

- A variable of an interface type can be assigned a reference to an object of a class that implements the interface

```
SomeInterface s = new SomeClass ();  
//Where SomeClass implements SomeInterface
```

rules

- When an object is assigned to an interface variable only methods defined in the interface could be accessed. Other methods defined in the class are not accessible
- The class that implements an interface must implement all the methods defined in the interface
- If the class does not implement one or more methods then the class must be marked as abstract

rules

- An interface can extend one or more interfaces using `extends` keyword
- A class implementing an interface must implement all the methods defined in the interface and its super interfaces

variables

- variables are static and final
- used for defining constants a long time
- later considered as bad practice

Next

Using Interfaces in JDK