

Interfaces

(OOP Core Concept 4 - Abstraction)

Learning objectives

- ▶ How to create and use Interfaces
- ▶ Interfaces vs Abstract methods
- ▶ Loose coupling

Interfaces

- ▶ Interfaces contain only abstract methods
- ▶ Interfaces are implemented by a class, not extended
- ▶ By implementing an interface, a class must implement the methods of the interface
- ▶ It's like signing a contract promising to implement certain methods

Interfaces vs Abstract methods

- ▶ Interfaces are almost like abstract classes with only abstract methods
- ▶ Abstract classes can have implementations of methods, but interfaces have only abstract methods (with some exceptions after Java 8)
- ▶ A class can implement many interfaces, but a class can only extend one superclass

The implements keyword

```
public interface Animal {  
    void eat();  
}
```



Implements

```
public class Dog implements Animal {  
    @Override  
    void eat() {  
        System.out.println("Dog eats");  
    }  
}
```

Changes since Java 8

- ▶ From Java 8 interfaces can have default methods with implementations
- ▶ This was introduced to be able to add methods in standard Java interfaces without breaking existing code
- ▶ Now interfaces can even have static methods and private methods

Loose coupling

- ▶ Interfaces facilitates loose coupling
- ▶ Loose coupling means that changes to one object should not need to result in changes in other objects, ie objects are more independent
- ▶ Tight coupling reduces flexibility and reusability
- ▶ By using interfaces as reference types code doesn't need to know what classes they are dealing with

Demo 1 - Interfaces

- ▶ Create an interface
- ▶ Implement an interface from a class
- ▶ Differences between abstract classes and interfaces

Exercise 1 for Interfaces

- ▶ Continue on the exercise from the Polymorphism module
- ▶ Change the abstract Product class into an interface
- ▶ What needs to change? Do you have to move functionality out from the Product into other classes? Maybe implement the description method in some classes that previously inherited it from the superclass?

Learning objectives

- ▶ How to create and use Interfaces
- ▶ Interfaces vs Abstract methods
- ▶ Loose coupling