

# Java

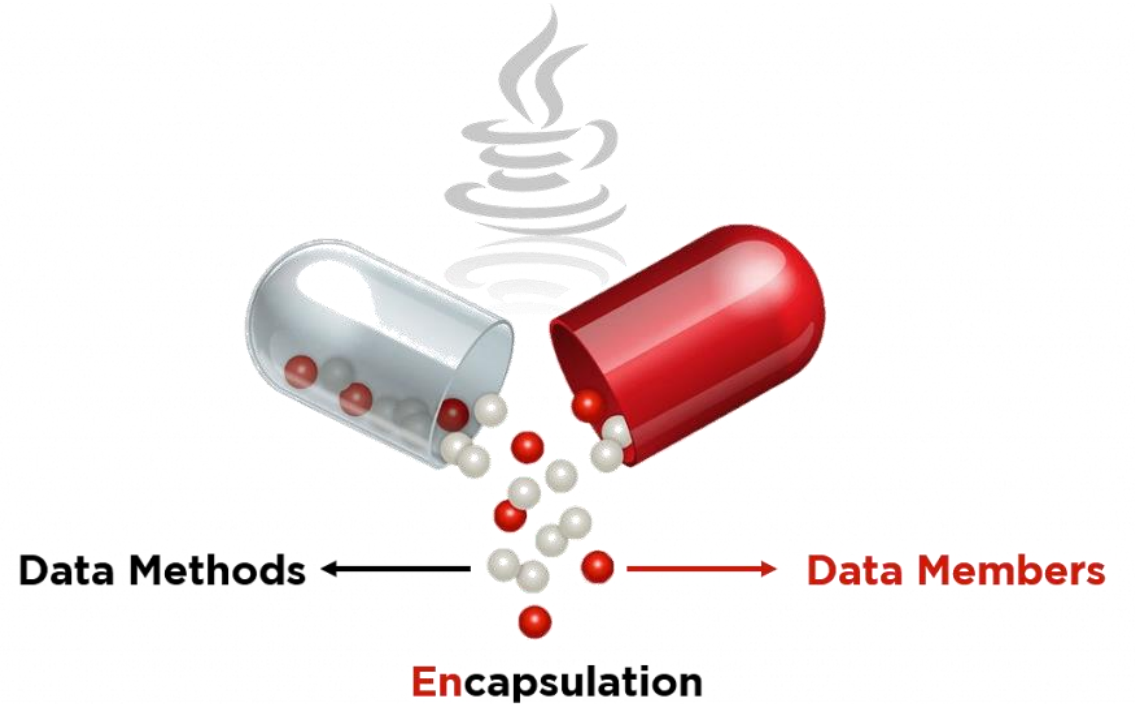
## <<Interface>>

## Based Polymorphism



# Encapsulation

```
public class Person {  
    private String name;  
  
    // Getter  
    public String getName() {  
        return name;  
    }  
  
    // Setter  
    public void setName(String newName) {  
        this.name = newName;  
    }  
}
```



# Encapsulation

---



Flexibility



Maintainability



Security

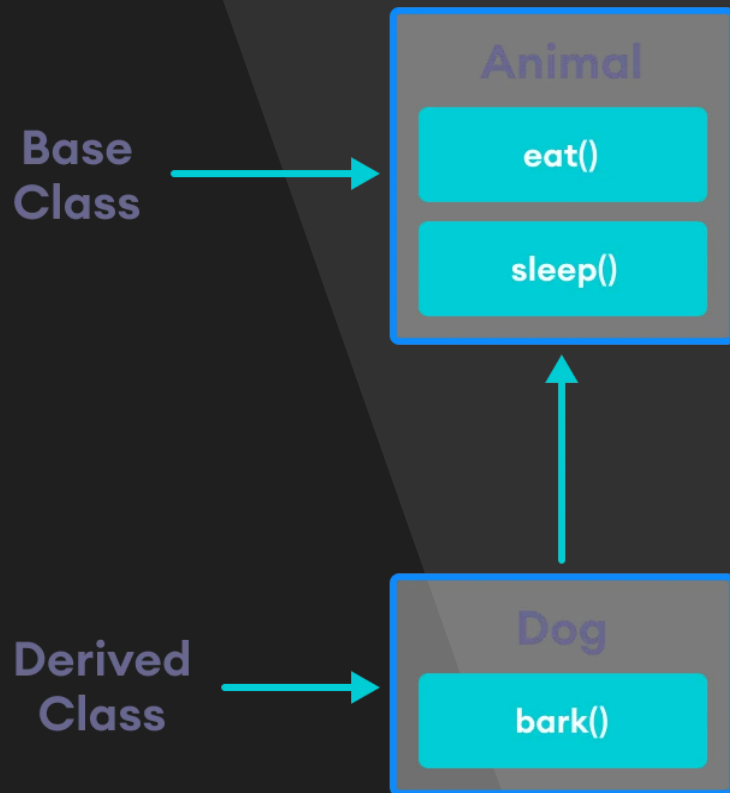


Protect data



Data hiding

# Inheritance



**subclass** (child)

**superclass** (parent)

To inherit from a class, use the **extends** keyword.

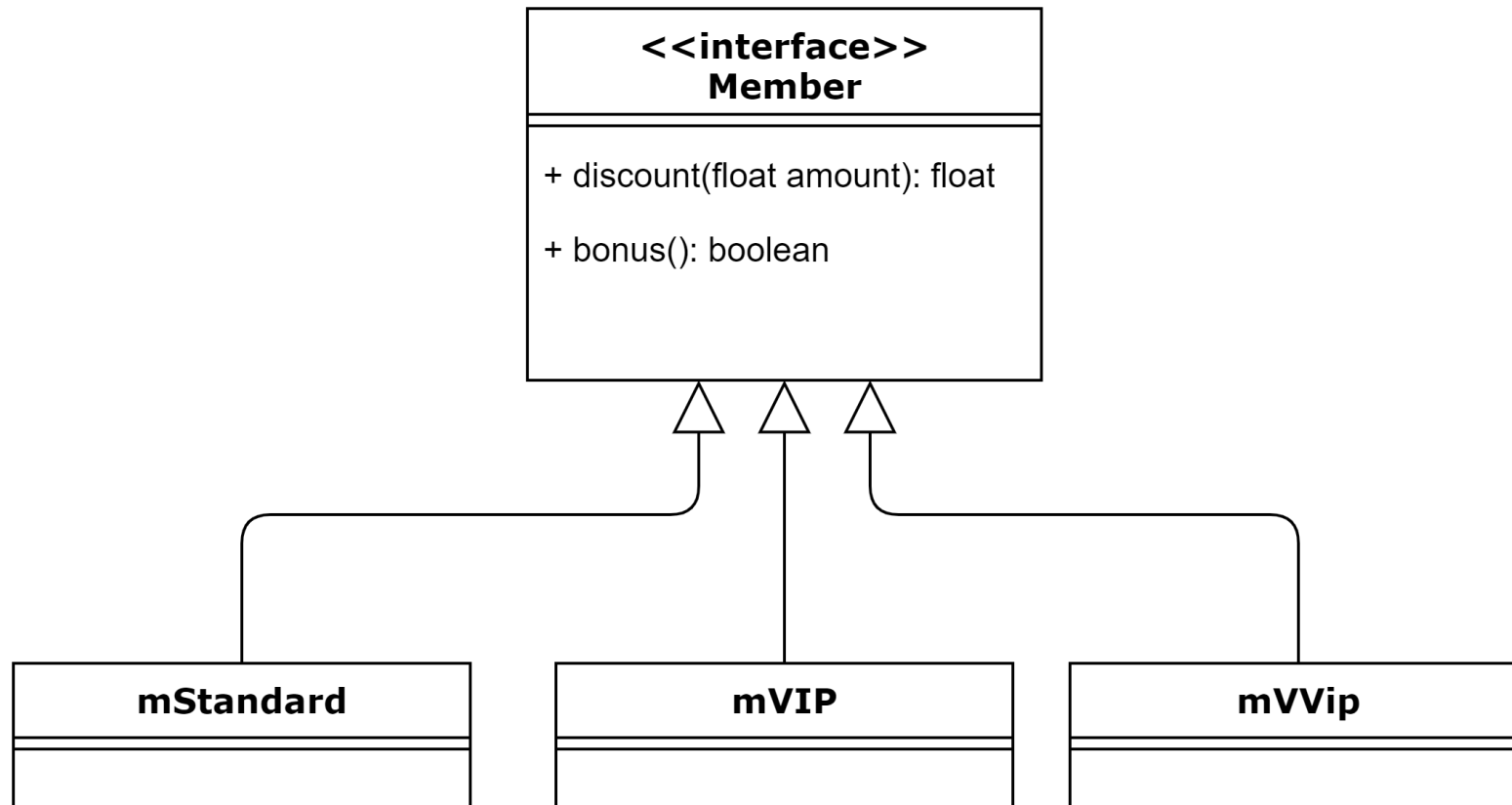
```
class Vehicle {  
    protected String brand = "Ford";  
    public void honk() {  
        System.out.println("Tuut, tuut!");  
    }  
}
```

```
class Car extends Vehicle {  
    private String modelName = "Mustang";  
    public static void main(String[] args) {  
        Car myCar = new Car();  
        myCar.honk();  
        System.out.println(myCar.brand + " " + myCar.modelName);  
    }  
}
```

# Polymorphism

---

# <<Interface>> Based Polymorphism



## Interface Based Polymorphism

To access the interface methods, the interface must be "**implemented**"

### Member.java

```
public interface Member {  
    float discount(float amount);  
    boolean bonus();  
}
```

### mStandard.java

```
public class mStandard implements Member{  
    @Override  
    public float discount(float amount) {  
        return amount * 0.02f;  
    }  
  
    @Override  
    public boolean bonus() {  
        return false;  
    }  
}
```

## Interface Based Polymorphism

To access the interface methods, the interface must be **"implemented"**

### Main.java

```
public class Main {  
    public static void main(String[] args) {  
        intface();  
    }  
}
```

```
public static void passMember(Member m, float amount) {  
    System.out.println(m.getClass().getName());  
    System.out.printf("amount = %10.2f\n", amount);  
    System.out.printf("discount = %10.2f\n", m.discount(amount));  
    System.out.printf("net = %10.2f\n", amount -  
        m.discount(amount));  
    System.out.println("*****");  
}
```

```
public static void intface() {  
    Member m1 = new mStandard();  
    passMember(m1, 100f);  
  
    Member m2 = new mVIP();  
    passMember(m2, 100f);  
  
    Member m3 = new mVVIP();  
    passMember(m3, 100f);  
}
```



# Lab

---

