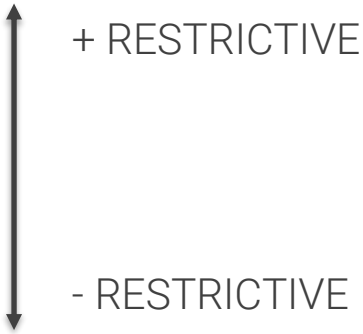
Protected access modifier

1. Attributes and methods access modifiers
2. Protected access modifier
3. Summary

Attributes and methods access modifiers

1. Attributes and methods access modifiers

- private
- no modifier (package private)
- protected
- public



1. Attributes and methods access modifiers

To define a well-encapsulated class:

1. Attributes and methods access modifiers

To define a well-encapsulated class:

✓ Define its **attributes** as **private** variables.

1. Attributes and methods access modifiers

To define a well-encapsulated class:

- ✓ Define its **attributes** as **private** variables. Allow access or manipulation to these variables using public methods (getters & setters).

1. Attributes and methods access modifiers

To define a well-encapsulated class:

- ✓ Define its **attributes** as **private** variables. Allow access or manipulation to these variables using public methods (getters & setters).
- ✓ Define **public constructors**.

1. Attributes and methods access modifiers

To define a well-encapsulated class:

- ✓ Define its **attributes** as **private** variables. Allow access or manipulation to these variables using public methods (getters & setters).
- ✓ Define **public constructors**.
- ✓ Define **public methods** to implement operations (accesible from other objects).

1. Attributes and methods access modifiers

To define a well-encapsulated class:

- ✓ Define its **attributes** as **private** variables. Allow access or manipulation to these variables using public methods (getters & setters).
- ✓ Define **public constructors**.
- ✓ Define **public methods** to implement operations (accesible from other objects).
- ✓ **Private methods** are **helper methods.**

1. Attributes and methods access modifiers

```
public class Clothing {
   String cod;
   double price;
   String size;
   String color;
   char genre; //W==Woman, M==Man

   public Clothing(String cod, double price, String size, String color, char genre) {
    this.cod = cod;
    this.price = price;
    this.size = size;
    this.color = color;
    this.genre = genre;
   }
   public String display() {
      return "cod='" + cod + '\'' +
            ", price=" + price +
            ", size='" + size + '\'' +
            ", color='" + color + '\'' +
            ", genre=" + genre;
   }
}
```

Clothing.java

1. Attributes and methods access modifiers

```
public class Clothing {
   String cod;
   double price;
   String size;
   String color;
   char genre; //W==Woman, M==Man

   public Clothing(String cod, double price, String size, String color, char genre) {
    this.cod = cod;
    this.price = price;
    this.size = size;
    this.color = color;
    this.genre = genre;
   }
   public String display() {
      return "cod='" + cod + '\'' +
            ", price=" + price +
            ", size='" + size + '\'' +
            ", color='" + color + '\'' +
            ", genre=" + genre;
   }
}
```

Clothing.java

1. Super

```
public class Jeans extends Clothing{

    String type; //slim, fit, ..

    public Jeans(String cod, double price, String size,
                 String color, char genre, String type) {
      super(cod, price, size, color, genre);
      this.type = type;
    }

    public String display() {
     return "Jeans{" + super.display() +
          ", type='" + type + '\'' +
          '}';
    }
}
```

Jeans.java

1. Attributes and methods access modifiers

To define a well-encapsulated class:

- ✓ Define its **attributes** as **private** variables. Allow access or manipulation to these variables using public methods (getters & setters).
- ✓ Define **public constructors**.
- ✓ Define **public methods** to implement operations (accesible from other objects).
- ✓ **Private methods** are **helper methods.**

1. Attributes and methods access modifiers

```
public class Clothing {
   String cod;
   double price;
   String size;
   String color;
   char genre; //W==Woman, M==Man

   public Clothing(String cod, double price, String size, String color, char genre) {
    this.cod = cod;
    this.price = price;
    this.size = size;
    this.color = color;
    this.genre = genre;
   }
   public String display() {
      return "cod='" + cod + '\'' +
            ", price=" + price +
            ", size='" + size + '\'' +
            ", color='" + color + '\'' +
            ", genre=" + genre;
   }
}
```

Clothing.java

1. Attributes and methods access modifiers

```
public class Clothing {
   private String cod;
   private double price;
   private String size;
   private String color;
   private char genre; //W==Woman, M==Man

   public Clothing(String cod, double price, String size, String color, char genre) {
    this.cod = cod;
    this.price = price;
    this.size = size;
    this.color = color;
    this.genre = genre;
   }
   public String display() {
      return "cod='" + cod + '\'' +
            ", price=" + price +
            ", size='" + size + '\'' +
            ", color='" + color + '\'' +
            ", genre=" + genre;
   }
}
```

Clothing.java

1. Attributes and methods access modifiers

```
public class Jeans extends Clothing{

    private String type; //slim, fit, ..

    public Jeans(String cod, double price, String size,
                 String color, char genre, String type) {
      super(cod, price, size, color, genre);
      this.type = type;
    }

    public String display() {
     return "Jeans{" + super.display() +
          ", type='" + type + '\'' +
          '}';
    }
}
```

Jeans.java

Accesibility matrix

Modifier	Class	Package	Subclass	Other Classes
Private	Yes	No	No	No
No modifier	Yes	Yes	No	No
Protected	Yes	Yes	Yes	No
Public	Yes	Yes	Yes	Yes

1. Attributes and methods access modifiers

```
public class Jeans extends Clothing{

    private String type; //slim, fit, ..

    public Jeans(String cod, double price, String size,
                 String color, char genre, String type) {
      super(cod, price, size, color, genre);
      this.type = type;
    }

    public String display() {
     return "Jeans{" + super.display() +
          ", type='" + type + '\'' +
          '}';
    }
}
```

Subclasse Jeans doesn't need access to Clothing private attributes

Jeans.java

1. Attributes and methods access modifiers

```
public class Jeans extends Clothing{

    private String type; //slim, fit, ..

    public Jeans(String cod, double price, String size,
                 String color, char genre, String type) {
      super(cod, price, size, color, genre);
      this.type = type;
    }

    public String display() {
     return "Jeans{" + super.display() +
          ", type='" + type + '\'' +
          '}';
    }
    public double finalPrice(){
      double profitMargin=0.20;
      return price / (1-profitMargin);

    }
}
```

Jeans.java

1. Attributes and methods access modifiers

```
public class Jeans extends Clothing{

    private String type; //slim, fit, ..

    public Jeans(String cod, double price, String size,
                 String color, char genre, String type) {
      super(cod, price, size, color, genre);
      this.type = type;
    }

    public String display() {
     return "Jeans{" + super.display() +
          ", type='" + type + '\'' +
          '}';
    }
    public double finalPrice(){
      double profitMargin=0.20;
      return price / (1-profitMargin);

    }
}
```

Subclasse Jeans needs access to Clothing private attribute price

```
public class Clothing{
 private String cod;
 private double price;
 ....
}
```

Jeans.java

1. Attributes and methods access modifiers

Main.java | Clothing.java | Jeans.java | Socks.java | Scarf.java

```java
package com.company;

public class Jeans extends Clothing{
    private String type; //slim, fit, etc

    public Jeans(String cod, double price, String size, String color, char genre, String type) {
        super(cod, price, size, color, genre);
        this.type = type;
    }

    public String display() {
        return "Jeans{" +
                super.display() +
                ", type='" + type + '\'' +
                "}";
    }

    public double finalPrice(){
        double profitMargin=0.20;
        return price / (1-profitMargin);

'price' has private access in 'com.company.Clothing'
}
```

Protected access modifier

2. Protected access modifier

Accesibility matrix

Modifier	Class	Package	Subclass	Other Classes
Private	Yes	No	No	No
No modifier	Yes	Yes	No	No
Protected	Yes	Yes	Yes	No
Public	Yes	Yes	Yes	Yes

2. Protected access modifier

```
public class Clothing {
   private String cod;
   private double price;
   private String size;
   private String color;
   private char genre; //W==Woman, M==Man

   public Clothing(String cod, double price, String size, String color, char genre) {
    this.cod = cod;
    this.price = price;
    this.size = size;
    this.color = color;
    this.genre = genre;
   }
   public String display() {
      return "cod='" + cod + '\'' +
            ", price=" + price +
            ", size='" + size + '\'' +
            ", color='" + color + '\'' +
            ", genre=" + genre;
   }
}
```

Clothing.java

2. Protected access modifier

```
public class Clothing {
   private String cod;
   protected double price;
   private String size;
   private String color;
   private char genre; //W==Woman, M==Man

   public Clothing(String cod, double price, String size, String color, char genre) {
    this.cod = cod;
    this.price = price;
    this.size = size;
    this.color = color;
    this.genre = genre;
   }
   public String display() {
      return "cod='" + cod + '\'' +
            ", price=" + price +
            ", size='" + size + '\'' +
            ", color='" + color + '\'' +
            ", genre=" + genre;
   }
}
```

Clothing.java

2. Protected access modifier

```
public class Jeans extends Clothing{

   private String type; //slim, fit, ..

   public Jeans(String cod, double price, String size,
                String color, char genre, String type) {
     super(cod, price, size, color, genre);
     this.type = type;
   }

   public String display() {
    return "Jeans{" + super.display() +
          ", type='" + type + '\'' +
          '}';
   }
   public double finalPrice(){
     double profitMargin=0.20;
     return price / (1-profitMargin);

   }
}
```

```
public class Clothing{
 private String cod;
 protected double price;
 ....
}
```

Jeans.java

Summary

3. Summary

To define a well-encapsulated class:

- ✓ Define its **attributes** as **private** variables. Allow access or manipulation to these variables using public methods (getters & setters).
- ✓ Define **public constructors**.
- ✓ Define **public methods** to implement operations (accesible from other objects).
- ✓ **Private methods** are **helper methods.**

3. Summary

To define a well-encapsulated class:

- ✓ Define its **attributes** as **private** variables. Allow access or manipulation to these variables using public methods (getters & setters).
- ✓ Define **public constructors**.
- ✓ Define **public methods** to implement operations (accesible from other objects).
- ✓ **Private methods** are **helper methods.**
- ✓ **Protected** attributes and methods if subclasses need access

“La carrera se hace en público, el talento en privado“

Marilyn Monroe, actriz de cine estadounidense