

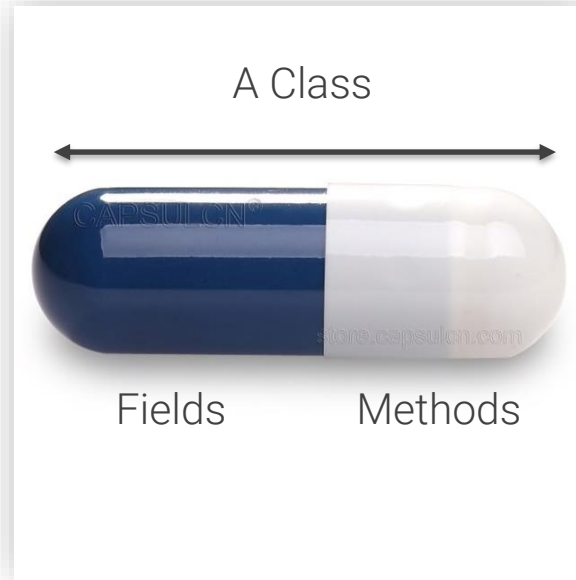


1. Encapsulation
2. Class access
modifiers

Encapsulation Introduction

Encapsulation

1. Encapsulation



A Class



INFORMATION HIDING

- Each object keeps its state private, inside a class: **attributes inaccessible from the outside**
- Other objects don't have direct access to this state, unless explicitly allowed by providing **getter and/or setter methods for attributes** that shall be readable or updatable by other classes
- Other objects can only call a list of its **public** methods.

JAVA ACCESS MODIFIERS

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



+ RESTRICTIVE

- RESTRICTIVE

SCHOOL LEARNING PLATFORM



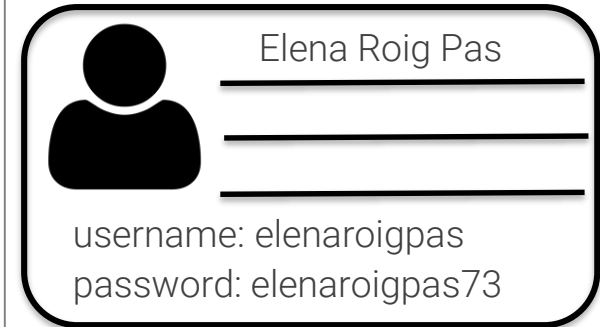
	Elena Roig Pas

username: elenaroigpas	
password: elenaroigpas73	


SCHOOL LEARNING PLATFORM

User.java

```
public class User{  
    String name;  
    //more attributes like address, phone, etc  
    String userName;  
    String userPassword;  
  
}
```



A user profile card with a black silhouette of a person on the left. To the right of the silhouette, the name "Elena Roig Pas" is displayed above three horizontal lines representing input fields. Below these fields, the text "username: elenaroigpas" and "password: elenaroigpas73" is shown.

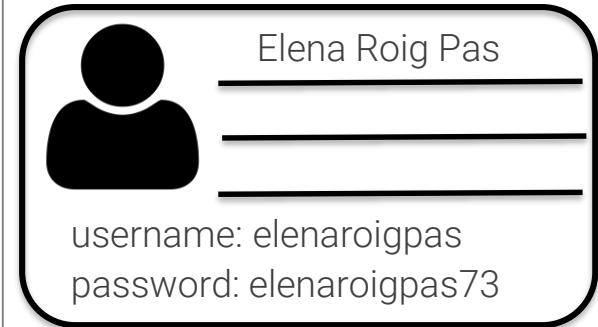
	Elena Roig Pas
username: elenaroigpas	
password: elenaroigpas73	

SCHOOL LEARNING PLATFORM


User.java

```
public class User{  
    String name;  
    String userName;  
    String userPassword;  
  
    User(String name) {  
        this.name = name;  
        this.userName = createUserName();  
        this.userPassword = createUserPassword();  
    }  
  
    String createUserName(){ //block of code }  
  
    String createUserPassword(){ //block of code }  
  
    String display(){ //block of code }  
}
```

Constructor



A user profile card with a black silhouette of a person on the left. To the right of the silhouette, the name "Elena Roig Pas" is displayed above three horizontal lines representing input fields. Below these fields, the text "username: elenaroigpas" and "password: elenaroigpas73" is shown.

	Elena Roig Pas
username: elenaroigpas	
password: elenaroigpas73	

SCHOOL LEARNING PLATFORM

User.java


```
public class User{
    String name;
    String userName;
    String userPassword;

    User(String name) {
        this.name = name;
        this.userName = createUserName();
        this.userPassword = createUserPassword();
    }

    String createUserName(){
        return name.replaceAll("\\s", "").toLowerCase();
    }

    String createUserPassword(){
        int num1 = (int) (Math.random() * 100) + 1;
        int num2 = (int) (Math.random() * 100) + 1;
        return this.userName+num1+num2;
    }

    String display(){ //block of code }
}
```



A user profile card with a black silhouette of a person on the left. To the right of the silhouette, the name "Elena Roig Pas" is displayed above three horizontal lines representing input fields. Below these fields, the text "username: elenaroigpas" and "password: elenaroigpas73" is shown, with "elenaroigpas" highlighted in yellow.

Elena Roig Pas

username: elenaroigpas

password: elenaroigpas73

SCHOOL LEARNING PLATFORM

User.java


```
public class User{
    String name;
    String userName;
    String userPassword;

    User(String name) {
        this.name = name;
        this.userName = createUserName();
        this.userPassword = createUserPassword();
    }

    String createUserName(){
        return name.replaceAll("\\s", "").toLowerCase();
    }

    String createUserPassword(){
        int num1 = (int) (Math.random() * 100) + 1;
        int num2 = (int) (Math.random() * 100) + 1;
        return this.userName+num1+num2;
    }

    String display(){ //block of code }
}
```



A user profile card with a black silhouette of a person on the left. To the right of the silhouette, the name "Elena Roig Pas" is displayed above three horizontal lines representing input fields. Below these fields, the text "username: elenaroigpas" and "password: elenaroigpas73" is shown, with the password field containing a yellow highlight.

Class access modifiers


2. Class access modifiers

SCHOOL LEARNING PLATFORM

```
public class User{  
    String name;  
    String userName;  
    String userPassword;  
  
    User(String name) {  
        this.name = name;  
        this.userName = createUserName();  
        this.userPassword = createUserPassword();  
    }  
  
    String createUserName(){  
        return name.replaceAll("\\s", "").toLowerCase();  
    }  
    String createUserPassword(){  
        int num1 = (int) (Math.random() * 100) + 1;  
        int num2 = (int) (Math.random() * 100) + 1;  
        return this.userName+num1+num2;  
    }  
    String display(){ //block of code }  
}
```

User.java

public:
The class is visible to all
classes everywhere



Elena Roig Pas

username: elenaroigpas
password: elenaroigpas73

SCHOOL LEARNING PLATFORM

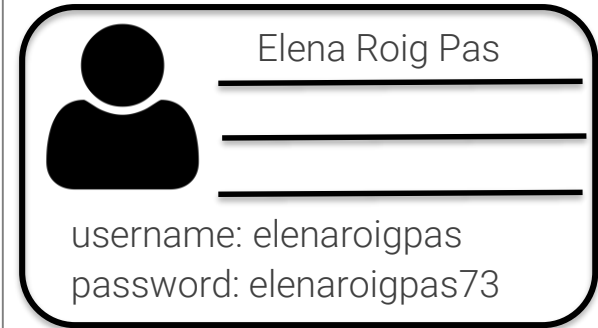
User.java

```
class User{
    String name;
    String userName;
    String userPassword;


    User(String name) {
        this.name = name;
        this.userName = createUserName();
        this.userPassword = createUserPassword();
    }

    String createUserName(){
        return name.replaceAll("\\s", "").toLowerCase();
    }
    String createUserPassword(){
        int num1 = (int) (Math.random() * 100) + 1;
        int num2 = (int) (Math.random() * 100) + 1;
        return this.userName+num1+num2;
    }
    String display(){ //block of code }
}
```

No modifier:
The class is visible only
withing its own package

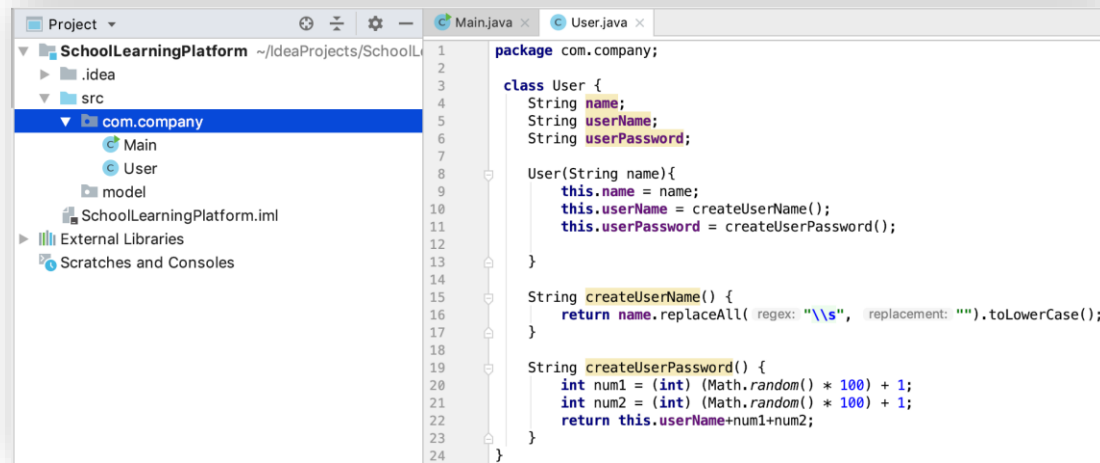


A user profile card with a black silhouette of a person on the left. To the right of the silhouette, the name "Elena Roig Pas" is displayed above three horizontal lines representing input fields. Below these fields, the text "username: elenaroigpas" and "password: elenaroigpas73" is shown.

	Elena Roig Pas

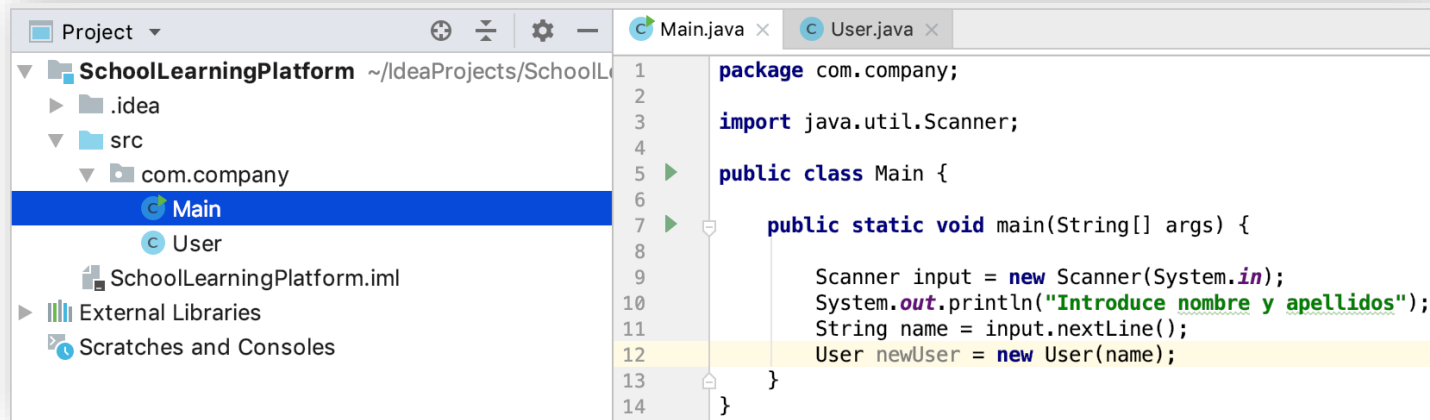
username: elenaroigpas	
password: elenaroigpas73	

SCHOOL LEARNING PLATFORM



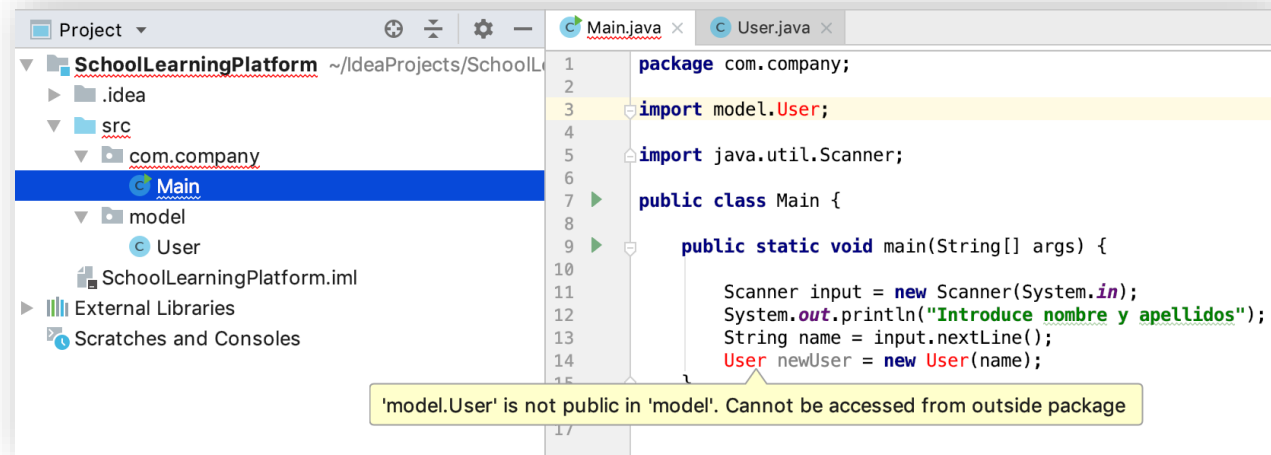
- class User: no access modifier
- class Main and class User are in the same package
- class User is visible from the class Main

SCHOOL LEARNING PLATFORM




- class User: no access modifier
- class Main and class User are in the same package
- class User is visible from the class Main

SCHOOL LEARNING PLATFORM



- class User: no access modifier
- class Main and class User **NOT** are in the same package
- class User is **NO** visible from the class Main

2. Class access modifiers

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

- **no modifier (package private):** the class is only visible within its own package
- **public:** the class is visible everywhere

“Cuando todo parezca ir en tu contra, recuerda que el avión despegó contra el viento. ”

Henry Ford, fundador de la compañía Ford Motor Company y padre de las cadenas de producción

