



JAVA CLASS

Object Oriented Programming II

JavaBean

- JavaBean. Often shortened to "bean".
- JavaBean is a reusable component in accordance with JAVA standards.

Properties

A bean is configurable.

A bean's non-public fields are called “properties”

A property can be either of type object or be a primitive type.

You can parameterize the bean by storing data in these fields.

Serialization



Serialization:

A bean is designed to be persistent.

Serialization is a process that allows you to save the state of a bean, giving you the option to restore it later to the saved state.

JavaBean is reusable

A bean is a component designed specifically to be reusable.

A JavaBean is not a EJB

The Bean or JavaBean

A bean :

- Must be a public class ;
- Must have at least one constructor by default, public and without parameters ;
- Can implement the Serializable interface, so it becomes persistent and its state can be saved. This is not required ;
- Cannot have any public fields (attributes);
- Can set properties (non-public fields), which must be accessible via public getter and setter methods, according to the naming convention rules (get / set).

Bean Example

```
public class Student implements Serializable{
//member data
private int id;
private String lastName;
private String firstName;
//constructor without parameters(not necessary, as java will create one if it
    is missing)
public Student(){
// getters setters
public int getId() {return id;}
public void setId(int id) {this.id = id;}
public String getLastName() {return lastName;}
public void setLastName(String lastName) {this.lastName = lastName;}
public String getFirstName() {return firstName;}
public void setFirstName(String firstName) {this.firstName = firstName;}
public int getAge() {return age;}
public void setAge(int age) {this.age = age;}
}
```

Class POJO



POJO is an acronym meaning
Plain Old Java Object.

POJO cont.

POJO is not bound to any restriction other than the Java language. In other words, it is imperative that a POJO:

- Does not implement pre-specified classes, such as:

```
public class Foo extends  
    javax.servlet.http.HttpServlet{...
```

- Does not implement pre-specified interfaces, such as:

```
public class Bar implements javax.ejb.EntityBean { ...
```

- Does not contain pre-specified annotations, such as:

```
@javax.persistence.Entity public class Baz { ...
```