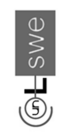
Relationships between objects II

**Christian Rodríguez Bustos**

Object Oriented Programming 

Relationships between objects

Reusupebeh

Agenda

Overriding

methods

Relationships between objects

Inheritance

and Access

Modifiers

Inheritance and accessibility

Inheritance and accessibility

Java Access Modifiers

Relationships between objects

Java Access Modifiers

**Access Levels**

Subclass PackageClass

☺

☺

☺

☺

☺

☺

☺ ☺

☺

Access level modifiers determine **whether other classes can usefield or invoke a particular method**

Relationships between objects

**Modifier**

public

protected

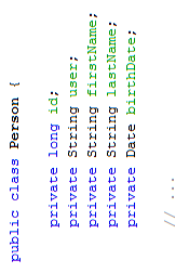
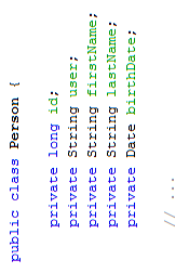
Default

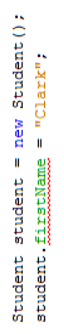
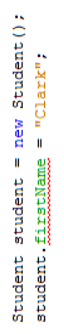
(no modifier)

private

Inheritance and Access Modifiers

Encapsulationthat attributedefined as **pri**And private attcannot be inhSo….

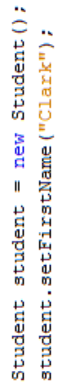
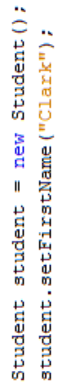
Relationships between objects

Inheritance and Access Modifiers 

Relationships between objects

How caaccessuperclattributessubcla

Inheritance and Access Modifiers

We can access to private attributes throug**superclass methods **

Relationships between objects

Inheritance and Access Modifiers 













**Public or protected**

**Person** Methods are  

inherited by the

**Student** subclass 

Inherited Person met

Relationships between objects

Inheritance and Access Modifiers 













**Public or protected**

****

****

****

****

**Person** Methods are

inherited by the



**Student** subclass 

Student methods

Relationships between objects

Inheritance and Access Modifiers

**Publi**

**protectedand St**Methodinherited**Graduate**subcl

Relationships between objects



Inheritance and Access Modifiers

**Public or protPerson and S**Methods are inby the 

**GraduateStu**subclass

Relationships between objects

Overriding methods

Relationships between objects

Overriding

Overriding involves “rewriting” how a m

works internally, **without changing the si**of that method.

Relationships between objects

In the real life

Animals talk inways

Dogs say: GUCats say: MELions say: GSpiders say: 

Relationships between objects

In UML life

Relationships between objects

I am the superclass

Relationships between objects

Respectioverrid

Overriding example 

sayHello mis called

Relationships between objects

Overriding example 

If no method is found, the JVM search forsuperclass

Relationships between objects

Overriding example (another methoRelationships between objects

This is a **valid**

**override**, because

both methods have

the same signature

**walk ( int )**

Overriding example (another methoRelationships between objects

This is a **invalid**

**override**, because

both methods have

different signature

**walk ( int )**

**walk ( long )**

More speciMore gene

Execution order

2

When we override

methods, first is called

the more specific

1

method

Relationships between objects

Reusing superclass behaviors

Relationships between objects

Reusing superclass methods

Relationships between objects

Bilingual Dog

can say

GUAU!!!

**Bilingual Dog**

**can say**

**REGUAUSS!!!**

Normal Dog 

Normal Dog only say GUAU!!!

**Bilingual Dog** speaks normal **Dog LanguAncient Dog Language**

Relationships between objects

We can reuse the superclass methods witkeyword

Relationships between objects

I am rsupersayHellBilinguspeak asusing tme

Reusing superclass methods 

Relationships between objects

Using UML design a **class hierarchy** (at least 3 levelsinheritance) for a **pet store with** at least 6 different kind

Create the **Java classes definitions** (encapsulated) foavailable on the pet store, each pet must have at least(not inherited).

Create a **test class** for your pet store, this class must smenu with the available pets (previously created).

User can select a pet and the **system must show all iavailable for this pet (including ancestor informatio**

Program finish when user select the option finish in the

**You have to use the keyword super and override an**Time to play in the pet store Relationships between objects

1.

2.

3.

4.

5.

**6.**

[Barker] J. Barker, Beginning Java Objects: From ConceptsSecond Edition, Apress, 2005.

http://download.oracle.com/javase/6/docs/api/java/lang/ObjeReferences

[Oracle] Understanding Instance and Class Members, Availhttp://download.oracle.com/javase/tutorial/java/javaOO/clas[Oracle] Java API documentation, Class Object, Available:

Relationships between objects