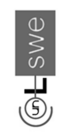
Objects Interaction

**Christian A. Rodríguez**

Object Oriented Programming 

Objects Interaction

How software works?

User interface triggers

Objects collaboration

Software layers

Objects Interaction

UI used to be a bridge between user and the “User perspective and the “black boObjects InteractionUsers do not know how software internally works 

Black box contains networks of related oObjects Interaction

Stud

Group

Professo

r

Course

UI Click

Objects collaborate and delegate tasObjects Interaction

Get grades for

each student

Group

Get this group

students list

Stud

Professo

Get my students

grades



r

Get this professor

group grades

Course

UI button

Click action

Get grades for

each student

Group

Get this group

students list

Stude

Delegation

Professo

Get my students

grades



r

Get this professor

group grades

UI button

Click

Professor object

**delegates** part of his

task to an Group

object

Objects Interaction

Delegation

Delegation is **invi**the object uDelegation amongobjects is exactly tas delegation bpeople in the rea

Objects Interaction



How to organize all those objects?

Objects Interaction

UI

• Software is generally structured

BusinLogi

– Business Logic Layer

• Each layer is responsible for a

Dat

set o related tasks

in layers:

– UI Layer

– Data Layer

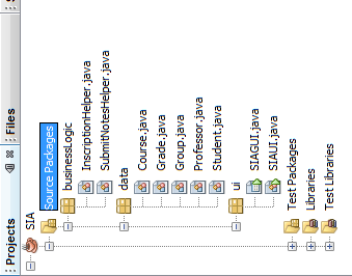
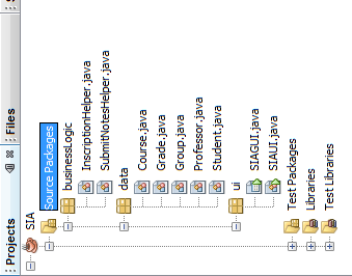
How to organize all those objects?

Classes can be

organized in pacEach layer can brepresented bypackage

Classes in differpackages mustimported in otheclasses

Objects Interaction











Layer responsibilities 

• Contains tforms, appages, etc 



• Do not cobusiness lUI Laye









• Contains the  

“functional code”. 

Business

Logic layer 

Objects Interaction

• Contains 

encapsulated classes  

definitions. 

• Contains data 

integrity validations. 

• Do not contain 

business logic code. 





Data layer 



Defining Methods

Declaring methods

Parameters and arguments

Method signatures & Methods overloading

Declaring methods - Knowing servic**Knowing services**

You need to know whichservices (methods) youto perform.

Fetch

Dance

Sit

Stay

Objects Interaction

Object A: You

Object B: Your pet



Declaring methods -Passing data

**Passing data**

Depending on the servicobject you may need to gsome additional informatiyour pet knows exactly h• Fetch beer

Fetch newspaper

Fetch stick

Objects Interaction

Object A: You

•

•

Object B: Your pet 

Declaring methods - Expecting someth**Expecting something?**

Your pet in turn needs to knwhether you expects your pback the outcome of what itasked to do.

Are you expecting your pthe beer?

Are you expecting your pthe stick

Are you expecting your pthe newspaper?

Objects Interaction

Object A: You

•

•

•

Object B: Your pet 

Declaring methods - Java perspectiObjects Interaction Passing data?

Knowing

services?

Expecting

something?

Declaring methods in Java terms 

Passing data

Objects Interaction

Knowing

services

Expecting

something?

Passing data

How it is the **passing** data process

Objects Interaction

Parameters and arguments

ArgPara

Parameter are

**variables**

Argument are

**values**

Objects Interaction

Parameters examples

Objects Interaction

Arguments examples

Objects Interaction

How it is the **returning** data procesReturning data

Objects Interaction

Methods can define zero or many returninObjects Interaction

Clients can use or not

returning data

Method header

Method header

Java definition for a method



Method’s Name 

This method header is:



Method’s return type 

nu

Course createCourse(String name, int



List of parameters 

with type and names 

Objects Interaction



Method signatures

This Method signature is:

createCourse(String , int)

The method **createCourse** declaresparameters of type **String** and **int**

respectively

**Method signature is unique**

Objects Interaction

Methods have

signatures

which indicates

Method’s  



Name 

Order, types  

and number  

of parameters 

Methods from the **same class** can be offea **unique name** but with **different sign**Methods overloading

Objects Interaction

println overloading 

Objects Interaction



println overloading

Compiler choose the

correct method checkingthe list of argumentspassed to parameters

Objects Interaction



Constructors

Objects Interaction

Do you remember how to instantiatObjects Interaction

This is the Student

class constructor

Constructors

Objects Interaction

object

Invoking a constructor

serves as a request to the

JVM to construct

(instantiate) a brand-new

Constructors

Constructors are special type of procedure**are responsible to ask the JVM to inflathelium balloon**

JV

M



Objects Interaction

Default constructor



If there is no defined constructors, the JVM

use the by **default constructor**

Objects Interaction

Default constructors setting all attributo their zero-equivalent default valuDefault constructor

Objects Interaction

Explicit constructors



We use **explicit constructors**

something more “interesting” to inwhen it is

Objects Interaction

Explicit constructors rules

Constructor’s name **must be e**as the name of the class for whithe constructor

Constructors works like anothedefine a **list of parameters **

We **cannot specify a return type** for a constructor; by

definition, a constructor returns a **reference to a newly**

**created object of the class type**

Objects Interaction

Passing parameters to constructor 

Test Class

Objects Interaction

Class definition

Be careful

If there is at least one constructor defin**cannot** use the default constructor

Objects Interaction

Replacing the Default Parameterless ConObjects Interaction

It is possible to overload Constructors like any other method

Overloading Constructors

Objects Interaction

Constructor 1 signatur

Student ( )

Constructor 2 signatur

) Student ( String

Constructor 3 signatur

Student ( String ,

Constructors reuse

Objects Interaction

**Code duplicati**

****

Constructors reuse 

It is possible to reuse Constructors using the keyword **this**

e

Reusing constructors can avoid duplication of cod

Objects Interaction

Review: Tic Tac Toe

Layers and packages

Constructors and access modificators

Relation between objects

Data Layer

Which classes

belong to the

data layer?



Objects Interaction

Data layer

Objects Interaction

Board class

Objects Interaction

Overriding default

constructor

Printing user defined objects metho 

Objects Interaction

toString method

toString method allow us **override the**

**default way how the object is printed**

Important, do

not forget it

The return type

is String

Objects Interaction

Overriding toString method 

Objects Interaction

Overloading 

constructors

Objects Interaction

Overriding toString method 

String.valueOf( ) is used to cast

variables to String

Objects Interaction

Player class

Objects Interaction

Overriding toString method 

Objects Interaction

How do you know if two user defined objeequal?

Objects Interaction

Overriding equals method

equals method allow us to **override the default way hodefined objects are eq**

Important, do

not forget it

If the two

objects have the

same player

(Symbol) are

equals

Objects Interaction

Which classes

business logic

belong to the

layer?

Business logic Layer

Objects Interaction



Business logic layer

Objects Interaction

GameEngine Class

This is the game starting point

Objects Interaction

Methods can be

private

**Function** 

GameEngine Class

Objects Interaction

**Method signature**

Initialize objects anCall the method plstartGame ( )

Iterate until win or fcondition is reache

Change the currenthe end of each tur getCurrentPlayer ( Player

**Class**

play ( )

GameEngine

)

TurnController Class

Objects Interaction

TurnController Class 

**Function**

Handle movemeCall movement vCall board modifi

Modify board aftplay

Chek if there aresquares to play

Generate randomovement

**Method signatureClass**

playTurn ( Board , Player )

markBoard ( Board , int ,

char )

TurnController

existFreeSquares ( Board ,

Player , Player)

generateRobotMovement ( )

Objects Interaction

MovementHandler Class

Objects Interaction

MovementHandler Class 

**Function**

Objects Interaction

**Method signature**

Check if a squaravailable to be isValid ( Board , int )

Check if the lastcauses the playisWinningMovement ( Board

**Class**

MovementHandl

, Player ) er

UML Class diagram

Objects Interaction

UML Class diagram 

Objects Interaction







**BUSINESS **

**LAYER **

**UI **

**LAYER**

****

– Business logi

Each layers is represented as a packObjects Interaction

• Three layers

– Data

– UI

Tic Tac Toe

Check the code here

Objects Interaction

References

[Barker] J. Barker, Beginning Java Objects: From ConceCode, Second Edition, Apress, 2005. •

Objects Interaction