

## Nom de l'assignatura : Enginyeria del Software 2

| Codi  | Tipus                                                                         | Curs/semestre          | Crèdits ECTS |
|-------|-------------------------------------------------------------------------------|------------------------|--------------|
| 21291 | <ul style="list-style-type: none"><li>• Troncal</li><li>• Semestral</li></ul> | 3er curs / 2n semestre | 5            |

### Professors

| Nom              | Dpt/Unitat                | Despatx | Direcció e-mail    | Telèfon   |
|------------------|---------------------------|---------|--------------------|-----------|
| Joan Serrat Gual | Ciències de la Computació | S/247   | Joan.serrat@uab.es | 935811828 |
|                  |                           |         |                    |           |
|                  |                           |         |                    |           |

## Objectius

### Objectiu genèric de l'assignatura

Mitjançant el desenvolupament d'una aplicació petita però realista l'estudiant assoleix coneixements bàsics de l'enginyeria del software. En particular sobre el disseny orientat a objecte, tant pel que fa a la notació (UML) com a aspectes avançants com ara els patrons de disseny i el disseny per contracte. Igualment, s'introdueixen mitjançant la pràctica aspectes de la gestió de la configuració i algunes tècniques de prova.

D'acord amb aquest objectiu, que genèricament en podriem dir *aprendre fent*, les classes magistrals (teoria) s'han reduït a la mínima expressió, i les de problemes i pràctiques reconvertit en reunions tutoritzades pel professor i avaluacions intermitges de la aplicació a desenvolupar.

### Coneixements

1. Disseny orientat a objecte :
  - UML i OCL
  - patrons de disseny
  - disseny per contracte
2. Disseny de la interfície gràfica d'usuari
3. Codificació : Java, estil de programació
4. Prova d'unitat (JUnit) i de validació (a partir de casos d'us)
5. Gestió de la configuració
6. Garantiment de la qualitat del software

### Competències

Capacitat de treball en equip, concretada en

- preparació prèvia i assistència a les reunions de grup
- planificació de les tasques a dur a terme
- acompliment de les tasques assignades
- tenir un rol actiu : aportar coneixements i participar en les discussions

### Habilitats

Donada la especificació potser incompleta dels requeriments d'una aplicació a desenvolupar,

- fer-ne el disseny orientat a objecte en UML, emprant els patrons de disseny adequats per a resoldre els problemes que apareixen en el disseny de classes

## Curs acadèmic 2010-11

- implementar el disseny en Java seguint un estil de codificació
- aplicar els principis del disseny per contracte a aquests classes
- dissenyar i implementar les corresponents proves d'unitat amb JUnit
- dissenyar i implementar una interfície gràfica d'usuari d'acord amb uns principis generals de qualitat
- durant els desenvolupament, que serà en equip, efectuar gestió de versions

## Capacitats prèvies

Disseny orientat a objecte.

Especificació de l'arquitectura del software amb UML.

Aconsellable, però no imprescindible, coneixements de programació en Java.

Encara que no calgui dir-ho, anglès a nivell de lectura de documents tècnics.

Haver treballat realment en equip en les pràctiques d'altres assignatures.

## Continguts

(T:teoria, S:seminaris=reunions tutoritzades i fites, PS:preparació de seminaris, L:laboratoris, PP:preparació pràctiques, E:estudi, AA:altres activitats)

| 1. Presentació de la assignatura | T   | S | PS | L | E | PP | AA | Total |
|----------------------------------|-----|---|----|---|---|----|----|-------|
|                                  | 0,5 |   |    |   |   |    |    | 0,5   |

Explicació del funcionament de la assignatura: planificació, avaluació, temari, aplicació a desenvolupar. Llista de tasques a fer per la primera reunió tutoritzada.

| 2. Patrons de disseny | T   | S   | PS   | L | E   | PP | AA | Total |
|-----------------------|-----|-----|------|---|-----|----|----|-------|
|                       | 2,5 | 5,0 | 11,0 |   | 5,0 |    |    | 23,5  |

Són solucions a problemes recurrents en disseny orientat a objecte, que cal conèixer per poder fer dissenys reusables i fàcilment mantenibles. Donada la gran quantitat de patrons proposats, il·lustrarem aquest concepte amb alguns dels més freqüents: *observer*, *composite*, *singleton*, *façade*, *factory*, *bridge*. Composició de patrons: *visitor*, *decorator*.

| 3. Disseny per contracte | T   | S   | PS  | L | E   | PP | AA | Total |
|--------------------------|-----|-----|-----|---|-----|----|----|-------|
|                          | 1,0 | 1,0 | 4,0 |   | 2,0 |    |    | 8,0   |

Assercions: invariants de classe, pre-condicions i post-condicions de les operacions (funcions públiques) d'una classe. Disseny per contracte envers programació defensiva. Implementació amb assert de Java.

| 4. Prova d'unitat | T   | S   | PS  | L | E   | PP | AA | Total |
|-------------------|-----|-----|-----|---|-----|----|----|-------|
|                   | 2,0 | 2,0 | 4,0 |   | 4,0 |    |    | 12,0  |

Concepte de prova d'unitat orientada a objecte: prova de classes. Framework JUnit per a Java.

| 5. Gestió de la configuració | T   | S   | PS  | L | E   | PP | AA | Total |
|------------------------------|-----|-----|-----|---|-----|----|----|-------|
|                              | 1,5 | 2,0 | 4,0 |   | 3,0 |    |    | 10,5  |

Concepte de gestió de la configuració. Control de versions del software: concepte de versió i baseline. Model de check-in, check-out. Branques de versions. Una eina pel control de versions: CVS.

## Curs acadèmic 2010-11

|                       |     |     |     |   |     |    |    |       |
|-----------------------|-----|-----|-----|---|-----|----|----|-------|
| 6. Prova de validació | T   | S   | PS  | L | E   | PP | AA | Total |
|                       | 1,5 | 2,0 | 4,0 |   | 3,0 |    |    | 10,5  |

Introducció a la prova del software. Tipus de proves. Una tècnica de prova de validació: disseny de casos de prova a partir de casos d'us.

|                                |     |     |     |   |     |    |    |       |
|--------------------------------|-----|-----|-----|---|-----|----|----|-------|
| 7. Interfície gràfica d'usuari | T   | S   | PS  | L | E   | PP | AA | Total |
|                                | 1,0 | 1,0 | 6,0 |   | 2,0 |    |    | 10,0  |

Introducció al disseny d'interfases gràfiques d'usuari: tipus d'interfases, principis genèrics de disseny, la utilització del color, confecció de missatges d'error i anàlisi d'errors de disseny comuns en interfases gràfiques d'aplicacions conegudes.

|                          |     |     |     |   |     |    |    |       |
|--------------------------|-----|-----|-----|---|-----|----|----|-------|
| 8. Qualitat del software | T   | S   | PS  | L | E   | PP | AA | Total |
|                          | 1,0 | 1,0 | 6,0 |   | 2,0 |    |    | 10,0  |

Concepte de qualitat del software. Les revisions tècniques formals com a mecanisme de garantiment de la qualitat.

## Metodologia docent

Durant l'horari de classes, els estudiants faran alguna de les següents activitats :

1. Classe de teoria: exposicions magistrals on s'introdueixen els conceptes bàsics de cada tema de teoria, però sense entrar en els detalls. Tots aquests temes són indispensables per al desenvolupament de la aplicació proposada.
2. Pràctiques o reunions tutoritzades: per efectuar el seguiment del desenvolupament d'una petita aplicació de la qual se'n donen unes breus especificacions. En elles, els estudiants del grup consensuen el que creuen els punts claus del problema, fan una llista de tasques i les assigna als seus membres, comparteixen coneixements, analitzen resultats, treuen conclusions, fan una *acta* de la reunió, i en lliuren una còpia al professor. I el professor, avalua l'estat del treball, remarca els punts febles o pendants, facilita l'aprenentatge fent preguntes clau i suggereix recursos útils.
3. Avaluació : 1) "fites", veure més avall, a instruments d'avaluació. 2) depenent del calendari acadèmic del curs (distribució dels dies festius), "qüestionaris breus" escrits. En algun cas, examen final.

Fora de l'horari de classes, cada grup haurà de fer reunions no tutoritzades per tirar endavant la resolució del problema en curs, a part del treball individual de cada membre. Cal comptar com a mínim una hora per setmana per a aquestes reunions.

Tot el material de la assignatura, lliuraments, qualificacions i la comunicació amb el professor, es fa mitjançant el **campus virtual**.

## Avaluació

### Fites.

El desenvolupament de la aplicació es divideix en quatre parts. Per a cada una, hi ha una fita, que és un punt d'avaluació. Els dies concrets són a l'enunciat i a les transparències de la

**Curs acadèmic 2010-11**

presentació de la assignatura. A les fites el grup presenta el treball fet fins aleshores i el professor el puntua segons el barem que figura a l'enunciat. Aquest barem té cinc qualificacions possibles : A=10, B=7.5, C=5, D=3, E=0.

La qualificació en una fita és la mateixa per a tots els membres del grup. Però si hi ha indicis que algun membre no ha treballat suficientment (de resultes de les reunions tutoritzades o de la nota dels qüestionaris), serà individual, en funció de les aportacions de cada un.

**Qüestionaris.**

Breu prova escrita per comprovar que tots els membres del grup han treballat, així com que han assimilat els coneixements de teoria. Els qüestionaris es faran depenent de la disponibilitat d'hores de classe, que varien cada curs segons les festes del calendari acadèmic.

**Sistema d'avaluació.**

1) Qui hagi faltat fins a tres reunions tutoritzades i no hagi faltat a cap fita,

- 70 % promig de les qualificacions de les fites
- 30 % promig dels qüestionaris. Si pel calendari acadèmic del curs no es poguessin fer, aquest percentatge passaria al promig de les qualificacions de les fites.

No cal aprovar per separat les dues parts.

2) Qui hagi faltat a quatre o més reunions tutoritzades o hagi faltat a alguna fita (per la raó que sigui), però les hagi lliurat al campus virtual dins del termini fixat,

- 50% promig de les qualificacions de les fites
- 50% exàmen el dia de la primera convocatòria. Conté problemes i preguntes sobre teoria i també sobre la aplicació desenvolupada, per verificar que realment s'ha fet.

Cal aprovar totes dues parts per separat.

3) Qui no hagi lliurat en el termini fixat alguna fita, "No presentat"

**No hi ha segona convocatòria.**

## Bibliografia bàsica

---

Design patterns: elements of reusable object oriented software. E. Gamma, R. Helm, R. Johnson, J. Vlissides. Addison Wesley, 1994. Versió en castellà de Pearson Educación, 2003. *Un gran clàssic. Imprescindible.*

Head first design patterns. E. Freeman, E. Freeman, K. Sierra, B. Bates O'Reilly, 2004. *Una alternativa a Gamma et al. il·lustrat amb nombrosos exemples, pas a pas.*

Object-oriented analysis and design: understanding system development with UML 2.0. Mike O'Docherty. Wiley, 2005. *Un llibre que cobreix quasi tots els temes del curs. I a més a més, els fonaments del disseny orientat a objecte i UML.*

Object oriented software engineering using UML, patterns and Java. 2a edició. Bernd Bruegge, Allen H. Dutoit. Prentice Hall, 2004. *Un bon llibre de text, com l'anterior. Tracta també el control dels canvis i la gestió de projectes (800pp.)*

A guide to software configuration management. A. Leon, Artech House, 2001.

Pragmatic unit testing in Java with JUnit. A. Hunt, D. Thomas. The Pragmatic Programmers, 2003.

A practitioner's guide to software testing. Lee Copeland. Artech House, 2004.

## Bibliografia complementària

---

Construcción de software orientado a objetos (2a edició). B.Meyer, Prentice Hall 1997.  
*Només els capítols 11 (disseny per contracte) i 12 (gestió d'excepcions), però un llibre molt interessant sobre els fonaments de la orientació a objecte.*

JUnit recipes. Practical methods for programmer testing. J.B.Rainsberger, S.Stirling. Manning Publications, 2004.

Software Configuration Management Patterns: Effective Teamwork, Practical Integration. S.P. Berczuk, B. Appleton. Addison-Wesley, 2002.

Software Verification and Validation for Practitioners and Managers, 2nd Edition. S. Rakitin. Artech House, 2001.

Eclipse in action. D. Gallardo, E. Brunette, R. McGovern. Manning Publications, 2003.  
*Una molt bona referència sobre Eclipse, amb la que ja en tindrem prou, si bé ja antiga.*

SWT/JFace in action: GUI design with Eclipse 3.0. M.Scarpino, S.Holder, S.Ng, L.Mihalkovic. Manning Publications, 2004.

Thinking i Java, 3rd edition. B. Eckel. Versió electrònica a <http://www.planetpdf.com/> . Versió impresa en castellà de Pearson Educación, 2002.

## Enllaços web

---

Inclosos en cada tema al campus virtual.