

Disseny de Software**2012/2013**

Codi: 102759

Crèdits ECTS: 6

Titulació	Pla	Tipus	Curs	Semestre
2502441 Graduat en Enginyeria Informàtica	958 Graduat en Enginyeria Informàtica	OT	0	0

Professor de contacte

Nom: Joan Serrat Gual

Correu electrònic: Joan.Serrat@uab.cat

Utilització d'idiomes

Llengua vehicular majoritària: català (cat)

Algun grup íntegre en anglès: No

Algun grup íntegre en català: Sí

Algun grup íntegre en espanyol: No

Prerequisits

No hi ha prerequisits oficials, però cal que l'estudiant domini la notació de diagrames de classe UML, els conceptes d'orientació a objecte i la seva programació en algun llenguatge, preferiblement Java.

Objectius

Un cop el propòsit d'un software s'ha determinat mitjançant l'anàlisi de requeriments, els desenvolupadors de software dissenyen una sol·lució. Aquesta va desde l'arquitectura dels seus components (classes, mòduls...) fins a la implementació en algun llenguatge de programació. Aquesta assignatura cobreix principalment dos aspectes claus del disseny: els patrons de disseny orientat a objecte i el disseny de la interfase (gràfica) d'usuari. A més d'aquests, tracta altres temes de disseny com són el disseny per contracte de classes i mètodes, l'estil de codificació i la refactorització.

Aquest assignatura doncs, exten una part del temari de Enginyeria del software de segon i es complementa amb la resta de les d'aquesta menció.

El mètode d'aprenentatge és basat en un projecte (la pràctica) o dit d'una altra manera, 'aprendre per fer': tots els conceptes tractats a teoria són necessaris per tal de dissenyar i implementar una aplicació de software, a partir d'un enunciat que fa el paper de document de requeriments. Es tracta de fer el disseny i la implementació (però no l'anàlisi de requeriments, la prova, el control de versions o el garantiment de la qualitat, que pertanyen a altres assignatures de la menció) d'una aplicació per a mòbil amb Android.

Competències

- Adquirir hàbits de pensament
- Adquirir hàbits de treball personal.
- Capacitat d'identificar i analitzar problemes i dissenyar, desenvolupar, implementar, verificar i documentar solucions de software sobre la base d'un coneixement adequat de les teories, models i tècniques actuals
- Capacitat per a desenvolupar, mantenir i avaluar serveis i sistemes de software que satisfacin tots els requisits de l'usuari i es comportin de manera fiable i eficient, siguin assequibles de desenvolupar i mantenir i compleixin les normes de qualitat, aplicant les teories, principis, mètodes i pràctiques de la Enginyeria del software

- Capacitat per concebre sistemes, aplicacions i serveis basats en tecnologies de xarxa, incloent Internet, Web, comerç electrònic, multimèdia, serveis interactius i computació mòbil
- Capacitat per concebre, desenvolupar i mantenir sistemes, serveis i aplicacions informàtiques emprant els mètodes de l'enginyeria del software com a instrument per a assegurar-ne la qualitat.
- Capacitat per dissenyar, desenvolupar, avaluar i assegurar l'accessibilitat, l'ergonomia, la usabilitat i la seguretat dels sistemes, serveis i aplicacions informàtiques, així com de la informació que gestionen
- Treballar en equip

Resultats d'aprenentatge

1. Assumir i respectar el rol dels diversos membres de l'equip, així com els diferents nivells de dependència de l'equip.
2. Concebre sistemes de software basats en tecnologies de xarxa, incloent Internet, Web, comerç electrònic, multimèdia, serveis interactius i computació mòbil.
3. Conèixer i aplicar els components de software específics en problemes computacionals basats en comunicacions.
4. Desenvolupar i mantenir sistemes de software que satisfacin tots els requisits de l'usuari i es comportin de forma fiable i eficient, siguin assequibles de desenvolupar i mantenir i compleixin les normes de qualitat, aplicant les teories, principis, mètodes i pràctiques de la Enginyeria del Software.
5. Desenvolupar la capacitat d'anàlisi, síntesi i prospectiva.
6. Dissenyar i documentar solucions de software.
7. Dissenyar una interface d'usuari partint de les especificacions del client.
8. Dividir problemes de disseny de programes en els seus components bàsics de funcionament.
9. Especificar les necessitats del client en un document d'especificació de software.
10. Identificar el paradigma que millor s'adapta al problema de disseny d'un sistema informàtic software específic tenint en compte els requeriments d'accessibilitat, ergonomia, usabilitat i seguretat.
11. Treballar cooperativament.
12. Treballar de manera autònoma.

Continguts

1. Principis de disseny orientat a objecte
 1. què és el disseny de software
 2. principis: encapsular el que varia, programar per a una interfície per comptes d'una implementació, obert a extensió i tancat a modificació, principi de substitució de Liskov, principi de responsabilitat única, composició millor que herència, evitar codi duplicat.
2. Patrons de disseny
 1. concepte de patró
 2. creacionals : singleton, factory method
 3. estructurals : composite, bridge, adapter, decorator
 4. de comportament : strategy, visitor, observer, mediator, command, template method, chain of responsibility
 5. patrons específics d'un domini
3. Disseny per contracte
 1. Object constraint language (OCL)
 2. especificació de mètodes: precondicions i postcondicions
 3. especificació de classes : invariants
4. Refactorització
 1. què és i per què cal
 2. exemple guia
 3. catàleg de refactoritzacions més freqüents
5. Estil de codificació
 1. què és i perquè cal
 2. regles d'estil
 3. documentació javadoc
6. Disseny de la interfase d'usuari
 1. importància de la interfase i tipus

2. principis de disseny i factors de qualitat
3. el sistema visual humà i el disseny d'interfases
4. proces de disseny de la interfase
7. Programació de la interfase d'usuari en Android
 1. conceptes bàsics de programació : activity, intent, service, broadcast receiver, views
 2. guia d'estil

Metodologia

Tal com hem dit als objectius, la assignatura segueix una metodologia d'aprenentatge basat en el desenvolupament d'un projecte. La seva implementació concreta pel que fa a l'ús de les hores de classe és la següent:

Classes de teoria. En elles el professor introdueix els continguts teòrics i dona referències útils (llibres, articles, pàgines web) per que l'estudiant després en pugui continuar el seu aprenentatge, així com desenvolupar l'aplicació software proposada. Les transparències utilitzades en aquestes classes doncs no s'han de pendre com a font d'estudi, sinó que serveixen per que el professor pugui exposar de manera més àgil. En canvi, cal estudiar els llibres recomanats, i els articles, capítols de llibre d'accés obert i altres, disponibles a la pàgina web de la assignatura.

Classes de problemes. En elles els estudiants presenten les respostes a una llista de qüestions, que hauran resolt prèviament. Seran tant problemes de disseny (diagrames de classe, codi, prototips d'interfase d'usuari etc.) com preguntes sobre articles que caldrà, es clar, haver llegit i analitzat, i que complementen els continguts de les classes de teoria o obliguen al seu estudi.

Sessions de pràctiques. Els estudiants hauran format a principi de curs grups petits (2 o 3 per grup, si és possible). Cada grup assisteix a una sessió de pràctiques d'una hora cada setmana. A les classes de pràctiques els estudiants venen a 1) 'passar comptes' amb el professor de la feina feta cada setmana i resoldre dubtes, i 2) donat que la pràctica estarà dividida en fites, les setmanes que corresponen a una fita s'avalua la part corresponent in situ, segons un barem que figurarà a l'enunciat de la pràctica. Per tant, les sessions de pràctiques *no* són per fer la pràctica.

Tots els materials de la assignatura juntament amb la seva planificació detallada es trobaran a la pàgina web <http://www.cvc.uab.es/shared/teach/a21291/>.

Activitats formatives

Títol	Hores	ECTS	Resultats d'aprenentatge
Tipus: Dirigides			
classes de problemes	12	0,48	2, 3, 5, 6, 7, 8, 12
classes de pràctiques	12	0,48	1, 2, 3, 4, 5, 6, 7, 11
classes de teoria	26	1,04	2, 3, 4, 6, 7, 8, 10, 12
Tipus: Autònomes			
estudi individual	29	1,16	3, 5, 6, 7, 8, 9, 10, 12
preparacio practiques	52	2,08	1, 2, 3, 4, 5, 6, 7, 8, 10, 11
resolució de problemes	12	0,48	2, 3, 5, 6, 7, 8, 9, 10, 12

Avaluació

La forma d'avaluació és continuada. Aquesta consta d'una part d'avaluació de grup i una altra d'individual. La nota de la primera s'obté fent promig de les notes de les fites de pràctiques (PR). La segona del promig de dues proves escrites sobre teoria, problemes i també la pràctica (EX), o bé d'un examen final de recuperació (ER). Les dues parts, la individual i la de grup, s'han d'aprovar per separat.

La nota final serà 'no presentat' si no s'ha entregat res avaluable: ni cap fita ni cap prova escrita. Altrament, cal entregar totes les fites de pràctiques, sinó la nota serà suspes. En cas d'haver entregat totes les fites, la nota final serà :

$0.5 * PR + 0.5 * EX$, si $PR \geq 5.0$ i $EX \geq 5.0$

$0.5 * PR + 0.5 * ER$, si $PR \geq 5.0$ però $EX < 5.0$

Una cosa que volem assegurar mitjançant les diferents avaluacions és que cada membre del grup realment ha contribuït a la pràctica en la mateixa mesura que els altres. Així doncs, si de resultes de les preguntes que el professor fa durant la fita o per les notes de les proves escrites, tenim indicis que això no passa, llavors la nota de fita passarà de ser de grup a individual : esbrinarem quina part de la fita actual ha fet cada membre i la avaluarem, i per les següents fites dividirem la feina a fer i seguirem fent avaluació individual de la part de cadascu, si el grup es manté amb tots els seus membres.

Les dates d'avaluació continuada i lliurament de treballs es publicaran a la pàgina web de l'assignatura i poden estar subjectes a possibles canvis de programació per motius d'adaptació a possibles incidències; sempre s'informarà a la pàgina web sobre aquests canvis ja que s'entén que aquest pàgina és el mecanisme habitual d'intercanvi d'informació entre professor i estudiants.

Plagiar, copiar o deixar copiar una activitat d'avaluació, implicarà suspendre aquesta amb un zero. A més, no serà recuperable encara que impliqui suspendre la assignatura.

Activitats d'avaluació

Títol	Pes	Hores	ECTS	Resultats d'aprenentatge
avaluació de grup	50%	3	0,12	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
avaluació individual	50%	4	0,16	2, 3, 5, 6, 7, 8, 9, 10, 12

Bibliografia

Head first object-oriented analysis and design. Brett D. McLaughlin, Gary Pollice, David West. O'Reilly, 2007.

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.

Head first design patterns. E. Freeman, E. Freeman, K. Sierra, B. Bates. O'Reilly, 2004.

Thinking in patterns (draft 0.9). B. Eckel. Versió web a <http://mindview.net/Books/TIPatterns/>

Construcción de software orientado a objetos (2a edició). B. Meyer. Prentice Hall 1997.

Refactoring: improving the design of existing code. M. Fowler, K. Beck, J. Brant, W. Opdyke, D. Roberts. Addison-Wesley, 2002.

The elements of Java style. Al Vermeulen, Scott W. Ambler, Greg Bumgardner et al. Cambridge University Press, 2000.

Designing with the Mind in Mind. Jeff Johnson. Morgan Kaufmann 2010.

User interface design and evaluation. Debbie Stone, Caroline Jarrett, Mark Woodroffe, Shailey Minocha. Morgan Kaufmann, 2005. Set capítols de mostra a <http://www.elsevierdirect.com/v2/companion.jsp?ISBN=9780120884360>

The essential guide to user interface design: an introduction to GUI design principles and techniques. Wilbert O. Galitz. Wiley, 2002.

Específics per les pràctiques :

Thinking in Java, 3rd edition. B. Eckel. Versió pdf a <http://mindview.net/Books/TIJ/>. Versió en castellà de Pearson Educación, 2002.

Professional Android 4: application development. Reto Meier. Wiley, 2012.

Android UI fundamentals: develop and design. J. Ostrander. Peachpit, 2012.