

Disseny de Software**2015/2016**

Codi: 102759

Crèdits: 6

Titulació	Tipus	Curs	Semestre
2502441 Enginyeria Informàtica	OB	3	1
2502441 Enginyeria Informàtica	OT	4	1

Professor de contacte

Nom: Joan Serrat Gual

Correu electrònic: Joan.Serrat@uab.cat

Equip docent

David Vazquez Bermudez

Utilització de llengües

Llengua vehicular majoritària: català (cat)

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 solució. Aquesta va des de 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 interfície (gràfica) d'usuari. A més d'aquests, tracta altres temes com són el disseny per contracte de classes i mètodes i l'estil de codificació.

Aquesta assignatura doncs, extén una part del temari d'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**Enginyeria Informàtica**

- 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, 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 concebre sistemes, aplicacions i serveis basats en tecnologies de xarxa, incloent Internet, Web, comerç electrònic, multimèdia, serveis interactius i computació mòbil
- 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. Orientació a objecte
 1. patrons GRASP
 2. principis de disseny orientat a objecte
2. Patrons de disseny
 1. creacionals
 2. estructurals
 3. de comportament
3. Disseny per contracte: OCL, pre, post-condicions, invariants
4. Estil de codificació: un estil per Java, comentaris, l'eina javadoc
5. Disseny de la interfície d'usuari
 1. usabilitat
 2. procés: prototipat, user testing
 3. patrons interfícies mòbil
 4. internacionalització
6. Programació d'interfícies Android

Metodologia

Tal com hem dit als objectius, l'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 dóna 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 prendre 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 llibres d'accés obert i altres, disponibles a la pàgina web de l'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, és 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). 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 l'assignatura juntament amb la seva planificació detallada es trobaran a la pàgina web <http://www.cvc.uab.es/shared/teach/a21291/web> . La majoria d'aquests materials són en anglès.

Les dates d'avaluació continuada i lliurament de treballs es publicaran en aquesta pàgina i poden estar subjectes a canvis de programació per motius d'adaptació a possibles incidències. S'informarà a classe o a través d'aquesta pàgina dels possibles canvis.

Activitats formatives

Títol	Hores	ECTS	Resultats d'aprenentatge
Tipus: Dirigides			
classes de pràctiques	12	0,48	1, 2, 3, 4, 5, 6, 7, 11
classes de problemes	12	0,48	2, 3, 5, 6, 7, 8, 12
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 (PR) s'obté fent promig de les notes de les fites de pràctiques (M1, M2, M3). La segona (EX) del promig de dues proves escrites durant el trimestre (EX1, EX2), o bé d'un examen final de recuperació (EXF).

L'algoritme pel càlcul de la nota final (FM) és el següent :

1. float EX = max((EX1 + EX2)/2., EXF)
2. float PR =(M1 + M2 + M3)/3.
3. if (EX >=5.) and (PR >=5.) and (M3 >0.) {
5. FM =(EX + PR)/2.
6. } else if (M3>0.) { // EX<5. or PR<5.
7. FM = min(EX, PR)
8. } else { // M3==0. whatever EX and PR
9. FM =0.
10. }

La nota final serà 'no presentat' si no s'ha entregat cap examen, o sigui, si no hi ha nota de EX1, EX2 ni EXF. La raó de la condició M3 > 0. és forçar a treballar en la tercera fita tot i que s'hagi obtingut una nota suficient per aprovar les pràctiques a les dues primeres fites.

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 l'avaluarem, i/o dividirem el grup.

Les dates d'avaluació continuada i lliurament de treballs es publicaran a la pàgina web de l'assignatura i poden estar subjectes a 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.

Sense perjudici d'altres mesures disciplinàries que s'estimin oportunes, i d'acord amb la normativa acadèmica vigent, es qualificaran amb un zero les irregularitats comeses per l'estudiant que puguin conduir a una variació de la qualificació d'un acte d'avaluació. Per tant, plagiar, copiar o deixar copiar una pràctica o qualsevol altra activitat d'avaluació implicarà suspendre-la amb un zero i no es podrà recuperar en el mateix curs acadèmic. Si aquesta activitat té una nota mínima associada, aleshores l'assignatura quedarà suspesa.

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

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.

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

Clean code: a handbook of agile software craftsmanship. R.C. Martin. Prentice Hall, 2008.

Paper prototyping: the fast and easy way to refine user interfaces. C. Snyder, Morgan Kaufmann, 2003.

Rocket surgery made easy: the do-it-yourself guide to finding and fixing usability problems. S. Krug. New Riders, 2010.

Disseny de Software 2015 - 2016

Mobile design pattern gallery: UI patterns for mobile applications. Theresa Neil.O'Reilly, 2012.

Android design patterns: interaction design solutions for developers. G. Nudelman. Wiley, 2013.

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.

Beginning Android 4 application development. Wei-Meng Lee. Wiley, 2012.

Android user interface design: turning ideas and sketches into beautifully designed apps. I.G. Clifton. Addison Wesley, 2013