

Programació Orientada als Objectes

Codi: 104389
Crèdits: 6

Titulació	Típus	Curs	Semestre
2503740 Matemàtica Computacional i Analítica de Dades	FB	1	2

Professor/a de contacte

Nom: Joan Serrat Gual

Correu electrònic: Joan.Serrat@uab.cat

Utilització d'idiomes a l'assignatura

Llengua vehicular majoritària: català (cat)

Grup íntegre en anglès: No

Grup íntegre en català: Sí

Grup íntegre en espanyol: No

Prerequisits

No hi ha prerequisits oficials, però cal saber programar, i saber-ho fer en Python. Conèixer Java és un valor afegit.

Objectius

Una de les metodologies més exteses de disseny i programació és la orientada a objectes, segons la qual un software s'organitza en classes que contenen mètodes (procediments) i atributs (dades). Les instàncies o objectes d'aquestes classes s'envien missatges les unes a les altres (fan "trucades" a mètodes d'altres objectes) aconseguint així la funcionalitat desitjada. A part del concepte de classe, tres altres elements fonamentals són l'herència, la composició i la separació de les parts pública i privada de les classes. Aquests elements són doncs necessaris per a la programació OO, però no suficients: cal saber com utilitzar-los per arribar a solucions que siguin fàcilment extensibles en el futur, ja que una llei del software és que els canvis són inevitables. Existeixen una sèrie de principis de disseny o heurístiques que ens diuen com s'han d'emprar aquests elements per tal d'aconseguir-ho. I sobre ells encara, es construeixen els patrons de disseny, que són solucions a nivell de disseny OO a problemes recurrents, que apareixen sovint en resoldre problemes aparentment independents. En aquesta assignatura s'introdueixen i es practiquen els conceptes OO així com els principis i patrons de disseny.

L'aprenentatge gira al voltant d'un projecte ---la pràctica--- en que és necessari aplicar tots els temes explicats per tal de dissenyar i implementar una aplicació de software a partir d'un enunciat que fa el paper de document de requeriments. Pel context de la assignatura, el projecte cau en el camp de l'aprenentatge computacional (machine learning). Els llenguatges de programació utilitzats a la assignatura seran Python principalment per que es en el que es farà el projecte, però també Java que utilitzarem en alguns exercicis i suporta millor els conceptes d'orientació a objecte. Així, podrem comparar com un altre els implementa i en quin grau.

Competències

- Aplicar coneixements bàsics sobre l'estructura, l'ús i la programació d'ordinadors, sistemes operatius i programes informàtics per solucionar problemes de diferents àmbits.
- Avaluar de manera crítica i amb criteris de qualitat el treball realitzat.
- Dissenyar, desenvolupar i avaluar solucions algorísmiques eficients per a problemes computacionals d'acord amb els requisits establerts.

- Que els estudiants hagin demostrat que comprenen i tenen coneixements en una àrea d'estudi que parteix de la base de l'educació secundària general, i se sol trobar a un nivell que, si bé es basa en llibres de text avançats, inclou també alguns aspectes que impliquen coneixements procedents de l'avantguarda d'aquell camp d'estudi.
- Que els estudiants puguin transmetre informació, idees, problemes i solucions a un públic tant especialitzat com no especialitzat.
- Que els estudiants sàpiguen aplicar els coneixements propis a la seva feina o vocació d'una manera professional i tinguin les competències que se solen demostrar per mitjà de l'elaboració i la defensa d'arguments i la resolució de problemes dins de la seva àrea d'estudi.
- Treballar cooperativament en un context multidisciplinar asumint i respectant el rol de los diferentes miembros del equipo.
- Utilitzar eficaçment la bibliografia i els recursos electrònics per obtenir informació.

Resultats d'aprenentatge

1. Avaluar de manera crítica i amb criteris de qualitat el treball desenvolupat.
2. Avaluar i analitzar la complexitat computacional de les solucions algorítmiques per poder desenvolupar i implementar aquella que garanteixi el millor rendiment.
3. Comprendre els principis bàsics de la lògica dels computadors.
4. Conèixer els conceptes bàsics de l'estructura i la programació dels computadors.
5. Descriure el funcionament bàsic dels sistemes de còmput.
6. Que els estudiants hagin demostrat que comprenen i tenen coneixements en una àrea d'estudi que parteix de la base de l'educació secundària general, i se sol trobar a un nivell que, si bé es basa en llibres de text avançats, inclou també alguns aspectes que impliquen coneixements procedents de l'avantguarda d'aquell camp d'estudi.
7. Que els estudiants puguin transmetre informació, idees, problemes i solucions a un públic tant especialitzat com no especialitzat.
8. Que els estudiants sàpiguen aplicar els coneixements propis a la seva feina o vocació d'una manera professional i tinguin les competències que se solen demostrar per mitjà de l'elaboració i la defensa d'arguments i la resolució de problemes dins de la seva àrea d'estudi.
9. Reconèixer i identificar els mètodes, els sistemes i les tecnologies propis de la computació.
10. Seleccionar i utilitzar les estructures algorítmiques i de representació de les dades apropiades per a la resolució d'un problema.
11. Treballar cooperativament en un context multidisciplinari assumint i respectant el rol dels diferents membres de l'equip.
12. Utilitzar eficaçment la bibliografia i els recursos electrònics per obtenir informació.
13. Utilitzar els sistemes operatius i els programes d'ús comú en diversos camps.
14. Verificar i assegurar el funcionament correcte d'una solució algorítmica d'acord amb els requisits del problema que cal resoldre.

Continguts

1. Conceptes d'orientació a objecte : encapsulament, herència, composició, interfície. Diagrames de classe UML
2. Programació orientada a objecte en Python3 i Java
3. Assignació de responsabilitats : patrons GRASP
4. Principis generals de disseny orientat a objecte
5. Patrons de disseny : creacionals, estructurals, de comportament
6. Estil de codificació, assercions i missatges de logging

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 dona referències útils (llibres, articles, pàgines web) per que l'estudiant després en pugui continuar el seu aprenentatge, així com desenvolupar el projecte proposat. Les transparències utilitzades en aquestes classes doncs no s'han de prendre com a única font d'estudi. Cal estudiar els llibres recomanats, i els articles, capítols de llibres d'accés obert i altres, disponibles a la pàgina web de l'assignatura. En aquestes classes també resoldrem petits problemes, de disseny i/o programació OO en Java i Python.

- Sessions de pràctiques. Els estudiants hauran format a principi de curs grups de 2-3 estudiants. En aquestes sessions els estudiants venen a 1) 'passar comptes' amb el professor de la feina feta cada setmana del projecte i resoldre dubtes, i 2) a ser avaluats de manera contínua, d'acord amb el barem que figura a l'enunciat de la pràctica. Per tant, la pràctica *no* es fa només durant aquestes sessions sino que demana un treball previ.

Tots els materials de l'assignatura juntament amb la seva planificació detallada es trobaran al campus virtual o bé a una la pàgina web que ja s'anunciarà. La majoria d'aquests materials són en anglès.

Activitats formatives

Títol	Hores	ECTS	Resultats d'aprenentatge
Tipus: Dirigides			
Classes de teoria	75	3	2, 4, 5, 9, 10
classes de pràctiques	75	3	7, 11, 13

Avaluació

Hi ha una part d'avaluació de grup i una altra d'individual. La nota de la primera (PR) correspon al projecte. La segona (EX) del promig de dues proves escrites durant el trimestre (EX1, EX2), o bé d'un examen final de recuperació (EXR).

L'algoritme pel càlcul de la nota final (FM) és el següent (totes les notes són sobre 10) :

```
EX = max((EX1 + EX2)/2.0, EXR )
if (EX >=4.0) and (PR >=4.0) :
FM = (EX + PR)/2.0
else :
FM = min(EX, PR)
```

Si no s'entrega un examen o la practica, la nota d'aquesta part és zero. No hi ha recuperació de la pràctica. La nota final serà 'no avaluable' si no s'ha entregat res avaluable, ja sigui examen o pràctica. Els estudiants repetidors no reben tractament diferenciat.

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 les sessions de projecte o per les notes de les proves escrites tenim indicis que això no passa, llavors la nota de projecte passarà de ser de grup a individual: avaluarem separatament la part del projecte que hagi fet cada membre, podent a més a més dividir el grup a partir d'aquell moment.

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 la 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
Examens	50%	0	0	2, 4, 5, 6, 7, 8, 9, 10
Pràctica	50%	0	0	1, 3, 4, 5, 7, 9, 11, 12, 13, 14

Bibliografia

Python: master the art of design patterns. D. Phillips, C. Giridhar, S. Kasampalis. Packt Publishing, 2016.

Mastering object-oriented Python. Steven F. Lott. Packt publishing, 2014.

Head first object oriented analysis and design. B.D. McLaughlin, G. Pollice, D. West. O'Reilly, 2007.

Design patterns explained simply. Alexander Shvets. <https://sourcemaking.com/design-patterns-ebook>.

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

Clean code: a handbook of agile software craftsmanship. R.C. Martin. Prentice Hall, 2008. O la versió en castellà, Código limpio, d'Anaya Multimedia 2012.