

Programació Orientada als Objectes

Codi: 104389
Crèdits: 6

Titulació	Tipus	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

Altres indicacions sobre les llengües

Els materials d'estudi (transparències, enunciats etc) són en anglès.

Equip docent extern a la UAB

Abel Serrat Castella

Prerequisits

No hi ha prerequisits oficials, però cal saber programar en Python.

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). 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 serie 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 son 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. Pel context de la assignatura, el projecte està relacionat amb 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 i 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 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 asumendo y respetando 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 Python 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, *logging*, comentaris i documentació

Metodologia

- 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 petits. 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. Tots aquests materials són en anglès.

Nota: es reservaran 15 minuts d'una classe, dins del calendari establert pel centre/titulació, per a la complementació per part de l'alumnat de les enquestes d'avaluació de l'actuació del professorat i d'avaluació de l'assignatura/mòdul.

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 a la pràctica. La segona (EX) del promig de dues proves escrites durant el trimestre (EX1, EX2), o bé de l'examen de recuperació (EXR).

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

$EX = \max(0.5 * (EX1 + EX2), EXR)$

if $(EX \geq 4.0)$ and $(PR \geq 4.0)$:

$FM = 0.5 * (EX + PR)$

else :

$FM = \min(EX, PR)$

La pràctica

La realització de la pràctica és essencial per assimilar els conceptes de teoria. La primera cosa doncs que volem assegurar és que la pràctica ha estat realment feta pel grup que la presenta, i que cada membre del grup hi ha contribuït en la mateixa mesura que els altres. La segona és que la pràctica es faci de manera progressiva durant el curs, en paral·lel a la teoria. Per això farem el següent:

- La nota de la pràctica s'obté combinant la nota de les parts en les que aquesta està dividida. Cada part serà avaluada in situ en una de les sessions de pràctiques que anunciarem a principi de curs (les fites). Els punts que es poden obtenir de cada part s'especificarà a l'enunciat. Aquestes avaluacions no tenen recuperació després.

- Els dies d'avaluació de la pràctica (fites), si de resultes de les respostes a les preguntes que faci el professor durant l'avaluació, tenim dubtes sobre l'autoria o la contribució d'algun dels membres del grup, llavors la nota de pràctica passarà de ser de grup a individual i en funció de quanta i quina part de la pràctica hagi fet cada membre.

Els exercicis

Proposarem alguns exercicis que són opcionals. Aquests exercicis es poden fer individualment o en petit grup, el mateix o diferent que el de pràctiques. Els exercicis poden sumar punts a la nota de teoria EX, amb un topall de 10. Diem 'poden' perquè la obtenció d'aquests punts extra estarà condicionada a una prova de validació per tal de comprovar que l'estudiant ha realitzat efectivament l'exercici que ha lliurat. Aquestes proves de validació les farem només un cop, als examens EX1 i EX2, i no tenen recuperació després.

Altres

No convalidem pràctiques, exercicis ni exàmens de cursos anteriors. Els estudiants repetidors no reben cap tractament diferenciat.

Si no s'entrega un examen o una part de la pràctica, la nota d'aquesta part és zero. La nota final serà 'no avaluable' si no s'ha entregat res avaluable, ja sigui examen, pràctica o exercici.

Les fites de pràctiques són avaluacions equivalents al examen escrits. Si un estudiant no s'hi presenta, la nota és zero.

Les notes MH les concedirem de manera discrecional als estudiants amb nota final superior 9.0 i tenint en compte tota mena de treballs fets (exàmens, pràctica i exercicis).

Les dates d'avaluació continuada i lliurament de treballs es publicaran al campus virtual de l'assignatura i poden estar subjectes a canvis en resposta a possibles incidències. Sempre s'informarà mitjançant el campus virtual, que serà el mecanisme de comunicació entre docents i estudiants.

Per a les activitats d'avaluació de tipus examen escrit, s'indicarà un lloc, data i hora de revisió en la que l'estudiant podrà revisar l'activitat amb el professor. Si l'estudiant no es presenta a aquesta revisió, no es revisarà posteriorment aquesta activitat.

Sense perjudici d'altres mesures disciplinàries que s'estimin oportunes, 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, la còpia, el plagi, l'engany, deixar copiar, etc. en qualsevol de les activitats d'avaluació implicarà suspendre-la amb un zero. Les activitats d'avaluació qualificades d'aquesta forma i per aquest procediment no seran recuperables. Si és necessari superar qualsevol d'aquestes activitats d'avaluació per aprovar l'assignatura, aquesta assignatura quedarà suspesa directament, sense oportunitat de recuperar-la en el mateix curs, amb nota zero.

Activitats d'avaluació

Títol	Pes	Hores	ECTS	Resultats d'aprenentatge
Exàmens	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

Bàsica

Les classes de teoria estan basades en un joc de transparències, que són un resum, una síntesi de conceptes que es troben més ben explicats en aquests llibres, que val molt la pena llegir o consultar per comprendre'ls millor i per ampliar coneixements.

Patrones de diseño. E. Gamma, R. Helm, R. Johnson, J. Vlissides. Pearson Educacion 2003.

Es un clàssic del patrons de disseny orientat a objecte, pels autors que van proposar aquest concepte. La versió original en anglès és de 1995 i per això els seus exemples i diagrames són ja antics. Però explica molt be els conceptes.

Design patterns explained simply. Alexander Shvets. Llibre on-line,
<https://sourcemaking.com/design-patterns-ebook>. També del mateix autor un altre llibre online a <https://refactoring.guru/design-patterns>

Ben explicat i il·lustrat amb exemples, que a més trobem programats en Java, Python, C++ i més.

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

Un llibre molt visual sobre patrons de disseny orientat a objecte. Molt didactic, els explica pas a pas i amb exemples en Java. Recentment n'ha aparegut una segona edició (2020).

Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow : concepts, tools, and techniques to build intelligent systems. Aurélien Géron. O'Reilly, 2019.

La pràctica consisteix en el disseny i programació orientada a objecte d'un mètode de classificació i regressió conegut com random forest. Tres capítols d'aquest llibre expliquen molt be els conceptes al voltant d'aquest mètode i l'algoritme mateix que cal implementar.

Complementària

Introduction to Programming Using Java, David J. Eck. 2020.

Llibre online, <http://math.hws.edu/javanotes/>. Alguns exercicis són en llenguatge Java i aquest és un llibre online i actualitzat que explica el llenguatge desde zero. No entra en els detalls més avançats però és més que suficient pel que necessitem.

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

Un llibre molt extens (700pp) sobre patrons de disseny en Python. Un aspecte molt positiu és que mostra com programar-los en Python en detall, un a un.

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

Python avançat, o més aviat, entrant als detalls del llenguatge. "*Grasp the intricacies of object-oriented programming in Python in order to efficiently build powerful real-world applications*". D'interès el capítol de *logging* i algun altre.

Programari

- Python en la versió Anaconda, que ja inclou les llibreries numpy, matplotlib.
- Com a entorn de desenvolupament farem servir Pycharm (millor que Spyder) en la versio "community edition". Li instalarem el plugin PlantUML.
- Per la programació en Java, IntelliJ IDEA també "community edition"