



122040038 - Bàtxelor en Informàtica (presencial i virtual)

0038014 - Programació avançada

Dades generals

Codi: 0038014

Tipus: OB

Semestre: 3

Crèdits: 6.0

**Llengua
d'impartició:** Català

Requisits

Codi - Assignatura

0038003 - Fonaments de programació

Professorat

Professorat responsable

Pellicer Bullich, Montserrat (Titulació: Llicenciatura en Ciències (Informàtica); Àrea de coneixement: Ciències tecnològiques)

Informació

Presentació de l'assignatura:

Aquesta assignatura és el punt d'entrada al paradigma de la programació orientada a objecte. És necessari haver adquirit les bases algorísmiques i de programació explicades en l'assignatura de Fonaments de programació, en què s'ha introduït el paradigma clàssic o procedimental.

Es recomana haver cursat Estructures de dades i Tècniques de programació per tal de tenir un cert nivell i maduresa en programació.

Concretament, l'orientació a objecte ha influït en diferents àmbits de les tecnologies, com ara, el desenvolupament de software. Això ha constituït un paradigma en programació, la programació orientada a objecte, i una disciplina, l'enginyeria del software, que incorpora una metodologia O.O. (vista en les assignatures d'Enginyeria del software I i II).

En l'assignatura es veuran els coneixements necessaris per entendre què és la programació orientada objecte i la programació concurrent.

Objectius:

- Aprendre els conceptes d'orientació a objectes i els avantatges que aporta.
- Introduir els diagrames UML
- Conèixer els fonaments teòrics de la programació concurrent.
- Entendre els conceptes d'exclusió mútua i sincronització entre processos.
- Resoldre problemes senzills en cadascun dels diferents models.
- Materialitzar en un llenguatge orientat a objecte (en aquest cas Java) els conceptes teòrics.
- Desenvolupar treballs pràctics en Java.
- Utilitzar les funcionalitats pròpies del llenguatge de programació per a la implementació de solucions.
- Ús d'un entorn integrat de desenvolupament.
- Utilitzar una eina de modelat basada en UML per fer els diagrames.

Metodologia:

Per tal d'assolir les competències i els objectius de l'assignatura s'anirà combinant els continguts teòrics amb la resolució de problemes i casos pràctics.

Els continguts teòrics de l'assignatura ens serviran de base per a plantejar els problemes i altres treballs pràctics que es proposaran a l'estudiant.

En les hores lectives es revisaran els continguts teòrics i es solucionaran problemes que en facilitaran la comprensió. En aquestes sessions, s'impulsarà la participació de l'estudiant, que haurà d'aportar el seu plantejament i discutir la solució als problemes o treballs pràctics. També hi haurà sessions al laboratori on es farà ús del llenguatge Java per realitzar diferents activitats i orientar les pràctiques obligatòries de l'assignatura.

Es recomana estudiar un mòdul, desenvolupar treballs pràctics i implementar-los amb el llenguatge (Java). És important practicar de forma continuada al llarg del semestre per poder obtenir les capacitats requerides en l'assignatura.

Hi haurà treballs (TR) i pràctiques (PR) que s'han de lliurar al professor a través del campus de l'UdA (campus.uda.ad). A més, s'han de realitzar una defensa presencial de totes les pràctiques.

L'aula virtual (o campus) esdevé l'eina bàsica de comunicació entre els estudiants i el professor fora de les hores lectives.

Avaluació continuada:

L'avaluació continuada contempla 3 exàmens presencials (ExAC), 3 pràctiques (PR) i 3 treballs opcionals (TR).

Per aprovar l'avaluació continuada cal obtenir una qualificació global d'avaluació continuada (QAC) ≥ 5 .

La qualificació d'avaluació continuada està formada per:

- Una qualificació d' exàmens, la mitjana d'aquests exàmens (QExAC) ha de ser ≥ 4 .
- Una qualificació de pràctiques, la mitjana d'aquestes pràctiques (QPR) ha de ser ≥ 4 .
- Una qualificació de treballs, la mitjana d'aquests treballs (QTR) permet millorar la qualificació global d'avaluació continuada (QAC).

La qualificació global d'avaluació continuada (QAC) serà:

- Màxim entre $QExAC * 0,6 + QPR * 0,2 + QTR * 0,2$; i $QExAC * 0,8 + QPR * 0,2$; Si $QExAC \geq 4$ i $QPR \geq 4$

On QExAC és la mitjana dels ExAC, QPR és la mitjana de les PR i QTR és la mitjana dels TR.

Avaluació final:

Si no s'aprova l'assignatura per avaluació continuada o s'opta per realitzar únicament l'avaluació final, hi ha la possibilitat de fer l'examen presencial final (ExF) sempre i quan s'hagin defensat prèviament les pràctiques.

L'avaluació final contempla 1 examen presencial final (ExF), 3 pràctiques (PR) i 3 treballs opcionals (TR).

Per aprovar l'assignatura cal obtenir una qualificació global de l'avaluació final (QAF) ≥ 5 .

La qualificació d'avaluació final està formada per:

- Una qualificació de l' examen final (QExF) que ha de ser ≥ 4 .
- Una qualificació de pràctiques, la mitjana d'aquestes pràctiques (QPR) ha de ser ≥ 4 .
- Una qualificació de treballs, la mitjana d'aquests treballs (QTR) permet millorar la qualificació global d'avaluació continuada (QAC).

La qualificació global de l'avaluació final (QAF) serà:

- NP; Si no es presenta a l'examen presencial final
- QExF; Si $QExF < 4$
- Màxim entre $QExF * 0,6 + QPR * 0,2 + QTR * 0,2$; i $QExF * 0,8 + QPR * 0,2$; Si $QExF$

≥ 4 i QPR ≥ 4

On QExF és la qualificació de l'ExF, QPR és la mitjana de les PR i QTR és la mitjana dels TR.

Bibliografia bàsica:

Material del professor

Apunts format diapositiva dels continguts de l'assignatura (Mòdul 1, 3).

Apunts de l'assignatura (Mòdul 2).

Recull d'exercicis per solucionar.

Llibre de referència 1:

Programació orientada a objectes.

JOAN ARNEDO MORENO

DANIEL RIERA I TERRÉN

JORDI BRÍNQUEZ JIMÉNEZ

ELENA GARCÍA BARRIOCANAL

MIQUEL ÀNGEL PIERA I EROLES

JUAN JOSÉ RAMOS GONZÁLEZ

Editorial UOC, 2007 ISBN 978-84-9788-582-9

Llibre de referència 2:

Programación concurrente.

JOSÉ TOMÁS PALMA MÉNDEZ,

M. C. GARRIDO CARRERA,

F. SANCHEZ FIGUEROA,

A. QUESADA ARENCIBIA

Editorial Thomson- Paraninfo, 2008 ISBN 84-9732-184-7

Bibliografia complementària:

[Laza & Baltasar, 2008]

R. Laza, J. Baltasar

Metodología y tecnología de la programación

Prentice-Hall, 2008

[Meyer, 1999]

Bertrand Meyer

Construcción de software orientado a objetos

Prentice-Hall, 1999

[Stevens & Pooley, 2007]

P. Stevens, R. Pooley

Utilización de UML

Addison Wesley, 2007

[Lea, 2001]

Doug Lea

Programación Concurrente en Java. Principios y patrones de diseño

Addison Wesley, 2001

Competències específiques (1)

Codi - Competència específica

BINFO02 - Disseny i construcció d'aplicacions informàtiques mitjançant tècniques de desenvolupament de software

Competències transversals (2)

Codi - Competència transversal

Àrea

Uda02 - Orientació cap a la qualitat

Responsabilitat personal

Uda10 - Comunicació i expressió oral i escrita

Comunicació

Continguts (54)

1. Programació orientada a objecte

1.1. Introducció

1.1.1. L'evolució. Del paradigma de la programació clàssica al paradigma O.O.

1.1.2. Influència de l'O.O.

1.1.3. Llenguatges de programació orientada a objecte

1.2. Conceptes bàsics d'O.O.

1.2.1. Classes i objectes. La classificació

1.2.2. Atributs i mètodes d'una classe

1.2.3. Abstracció. Encapsulament i ocultació de la informació

1.2.4. Responsabilitats de classe i d'objecte

1.2.5. Relació de dependència entre classes

1.2.6. Visió genèrica d'un sistema O.O.

1.3. Herència i polimorfisme

1.3.1. Conceptes d'herència. La jerarquia

1.3.2. Sobrecàrrega i polimorfisme

1.3.3. Enllaç estàtic i enllaç dinàmic

1.3.4. Tipus d'herència

1.3.5. Tipus de classes

1.3.6. Interfícies

1.3.7. Delegació

1.3.8. Compatibilitat d'objectes

1.3.9. Conversions de tipus/classe

1.4. Modelització de classes en UML

1.4.1. L'estàndard U.M.L.

1.4.2. Representació d'una classe

1.4.3. Relacions entre classes. Associacions

- 1.4.4. La relació de generalització/especialització
- 1.4.5. La relació d'agregació/composició
- 1.4.6. El model conceptual de classes
- 1.5. Un procés simplificat
 - 1.5.1. Etapes
 - 1.5.2. Principis de disseny
 - 1.5.3. Un exemple
- 2. Programació concurrent
 - 2.1. Concepte generals sobre programació concurrent
 - 2.1.1. Què s'entén per programació concurrent?. Models
 - 2.1.2. Abstraccions realitzades en programació concurrent
 - 2.1.2. Justificació
 - 2.1.4. Problemes associats a la concurrència: deadlock, indeterminisme i equitativitat
 - 2.1.5. Representació del paral·lelisme
 - 2.1.6. Llenguatges de programació concurrent
 - 2.2. Solucions per l'exclusió mútua
 - 2.3. Mecanismes de comunicació i sincronització de processos
 - 2.3.1. Semàfors
 - 2.3.2. Regions crítiques
 - 2.3.3. Monitors
 - 2.3.4. Bústies
 - 2.3.5. Problemes clàssics de sincronització de processos: problemes de competència i cooperació
- 3. El llenguatge Java
 - 3.1. Introducció: tipus bàsics i estructures de control
 - 3.2. Aspectes d'O.O : objectes, classes, herència, polimorfisme, conversions de tipus ...
 - 3.3. Col·leccions d'objectes (Vector, List ...)
 - 3.4. Interfície gràfica d'usuari i gestió d'esdeveniments
 - 3.5. Programació concurrent en Java. Control de threads

Activitats (3)

Codi - Activitat	Descripció	Competències	Percentatge
1 - Ex AC	Exàmens presencials	BINFO02, UdA02	60,00%
2 - TR	Treballs	BINFO02, UdA02	20,00%
3 - PR	Pràctiques	BINFO02, UdA02	20,00%