



DATOS IDENTIFICATIVOS

Programación II

Materia	Programación II			
Código	V05G300V01302			
Titulación	Grao en Enxeñaría de Tecnoloxías de Telecomunicación			
Descritores	Creditos ECTS	Sinale	Curso	Cuadrimestre
	6	OB	2	1c
Lingua de impartición	Castelán			
Departamento	Enxeñaría telemática			
Coordinador/a	Fernández Masaguer, Francisco			
Profesorado	Blanco Fernández, Yolanda			
	Fernández Masaguer, Francisco			
	Sousa Vieira, Estrella			
Correo-e	francisco.fernandez@det.uvigo.es			
Web	http://www.faitic.es			

Descrición xeral O obxectivo xeral da materia é proporcionar ó alumno os fundamentos teóricos e as competencias prácticas que lle permitan analizar, deseñar, desenvolver e depurar aplicacións informáticas seguindo o paradigma orientado a obxectos. Esta é unha materia eminentemente práctica e neste sentido está orientada ó traballo dos alumnos na realización dun ou varios proxectos. Para facilitar o desenvolvemento dos proxectos, na materia tamén se fai unha introducción á Enxeñaría do Software. Neste sentido non se ocupa de todas as fases xeralmente recoñecidas nos procesos de desenvolvemento software, que van desde a captura e descrición de requisitos ata o despregamento dos sistemas, senón que se tratan principalmente as etapas de análise, deseño, implementación e depuración. En primeiro lugar presentaráse a Enxeñaría do Software como disciplina imprescindible para o desenvolvemento de grandes aplicacións informáticas, amosando os principais retos ós que se enfronta e os conceptos básicos que se empregarán. A continuación analizaráse os elementos do paradigma da programación orientada a obxectos (POO), utilizando elementos e diagramas UML que serán empregados polos alumnos nos seus desenvolvementos. Para acadar este obxectivo xeral os contidos que se verán na materia pódense resumir nos seguintes ítems:

□ O paradigma Orientado a Obxectos

- Conceptos básicos da orientación a obxectos: clases e obxectos
- Encapsulación. Principio de ocultación. Conceptos de desacoplamento e cohesión
- Herdanza, abstracción, polimorfismo e reutilización
- Relacións entre clases: xeneralización, asociación e dependencia
- Comunicación entre obxectos: métodos, eventos, mensaxes
- Persistencia. Almacenamento en ficheiros e en bases de datos
- Xeneración, captura e procesamento de excepcións

□ Introdución á Enxeñaría do Software

- Conceptos básicos da Enxeñaría do Software. Reseña histórica
- Introdución e concepto de Ciclo de Vida. Estándar ISO/IEC 12207
- Introdución ás metodoloxías de desenvolvemento de software. Clasificación
- Introdución ós procesos de desenvolvemento de software orientado a obxectos. Métrica v3 e o Proceso Unificado.
- Fases principais no desenvolvemento orientado a obxectos: análise, deseño, implementación e probas.
- Introdución á linguaxe de modelado UML: estrutura e interacción.

Competencias

Código

B6	CG6 Facilitade para o manexo de especificacións, regulamentos e normas de obrigado cumprimento.
B14	CG14 Capacidade para utilizar ferramentas informáticas de procura de recursos bibliográficos ou de información.
C50	(CE50/T18) Capacidade de desenvolver, interpretar e depurar programas utilizando os conceptos básicos da Programación Orientada a Obxectos (POO): clases e obxectos, encapsulación, relacións entre clases e obxectos, e herdanza.
C51	(CE51/T19) Capacidade de a aplicación básica das fases de análises, deseño, implantación e depuración de programas na POO.
C52	(CE52/T20) Capacidade de manexo de ferramentas CASE (editores, depuradores).
C53	(CE53/T21) Capacidade de desenvolvemento de programas atendendo aos principios básicos de calidade da enxeñaría do software, tendo en conta as principais fontes existentes en normas, estándares e especificacións.

Resultados de aprendizaxe

Resultados previstos na materia	Resultados de Formación e Aprendizaxe	
Comprender os aspectos básicos da Programación Orientada a Obxectos (POO).	B14	C50
Coñecer os principais diagramas UML para a documentación nas fases de análise e deseño de programas de acordo á POO.	B6	C52
	B14	C53
Desenvolver habilidades no proceso de análise, deseño, implementación e depuración de aplicacións de acordo á POO, tendo en conta os estándares principais e normas de calidade.	B6	C51
	B14	C53

Contidos

Tema	
1. Introducción ó paradigma orientado a obxectos	a. Breve introdución á materia e a súa organización b. Nacemento do paradigma c. Bases: clases e obxectos d. Conceptos de encapsulación, herdanza (xeneralización), e polimorfismo e. Breve introdución a UML
2. Encapsulación	a. Clases, interfaces e paquetes b. Métodos e variables membro. Visibilidade. Resolución de ámbito. c. Método constructor d. Paso de parámetros: punteiros e referencias e. Punteiros a obxectos
3. Herdanza	a. Clases derivadas e tipos de herdanza b. Clases abstractas c. Herdanza múltiple d. Clase object
4. Deseño orientado a obxectos	a. Fundamentos de deseño b. Conceptos básicos da Enxeñaría do Software c. Utilización de diagramas UML
5. Polimorfismo	a. Sobrecarga e sobreescritura b. Clases abstractas e interfaces c. Clases xenéricas
6. Xestión de excepcións	a. Fundamentos de excepcións b. Manipulación de excepcións en Java

Planificación

	Horas na aula	Horas fóra da aula	Horas totais
Sesión maxistral	28	42	70
Resolución de problemas e/ou exercicios	9	9	18
Resolución de problemas e/ou exercicios de forma autónoma	4	10	14
Estudo de casos/análises de situacións	1	1	2
Proxectos	9	31	40
Estudo de casos/análise de situacións	0	1	1
Resolución de problemas e/ou exercicios	3	0	3
Probas prácticas, de execución de tarefas reais e/ou simuladas.	2	0	2

*Os datos que aparecen na táboa de planificación son de carácter orientador, considerando a heteroxeneidade do alumnado.

Metodoloxía docente

	Descrición
Sesión maxistral	Clases que combinarán a exposición dos conceptos a tratar na materia coa realización de pequenos exercicios. Éstos poderán ser resoltos polo docente ou polos propios alumnos individualmente e/ou en grupo. O obxectivo é fomentar o debate na clase e reforzar a adquisición de destrezas. Esta metodoloxía está orientada á adquisición das competencias CE50, CE51 e CE53.
Resolución de problemas e/ou exercicios	No laboratorio, o profesor plantexará pequenos retos que serán resoltos colectivamente para que se poidan debater os conceptos subxacentes, as diferentes opcións de resolución e que os alumnos adquiren as destrezas obxectivo da materia. Esta metodoloxía está orientada á adquisición das competencias CE50, CE51 e CE53.
Resolución de problemas e/ou exercicios de forma autónoma	Os alumnos resolverán de forma autónoma os problemas que o profesor lles plantexe no laboratorio. As solucións e as dúbidas que xurdan ó abordar ditos problemas serán postas en común para consensuar a mellor forma de resolución. Esta metodoloxía está orientada á adquisición das competencias CE50, CE51, CE53, CG6 e CG14.
Estudo de casos/análises de situacións	Posta en común dos deseños propostos polos alumnos para solucionar o proxecto que teñen que levar a cabo durante a segunda parte do curso. A comparación das diferentes propostas servirá para seleccionar as mellores opcións e como realimentación para, se é oportuno, mellorar os deseños realizados. Esta metodoloxía está orientada á adquisición das competencias CE51 e CE52.

Proxectos	Os alumnos implementarán o sistema software plantexado polo profesor. Disporán para elo da segunda parte do curso, combinando traballo presencial no laboratorio supervisado polo profesor con traballo non presencial. Esta metodoloxía está orientada á adquisición das competencias CE50, CE53, CG6 e CG14.
-----------	---

Atención personalizada

Metodoloxías	Descrición
Resolución de problemas e/ou exercicios	A atención individualizada articularáse co seguemento do traballo de cada alumno, supervisando as solucións que propón para cada problema plantexado nas prácticas de laboratorio, e o seguemento do proxecto software que debe implementar.
Proxectos	A atención individualizada articularáse co seguemento do traballo de cada alumno, supervisando as solucións que propón para cada problema plantexado nas prácticas de laboratorio, e o seguemento do proxecto software que debe implementar.
Resolución de problemas e/ou exercicios de forma autónoma	A atención individualizada articularáse co seguemento do traballo de cada alumno, supervisando as solucións que propón para cada problema plantexado nas prácticas de laboratorio, e o seguemento do proxecto software que debe implementar.
Estudo de casos/análises de situacións	A atención individualizada articularáse co seguemento do traballo de cada alumno, supervisando as solucións que propón para cada problema plantexado nas prácticas de laboratorio, e o seguemento do proxecto software que debe implementar.

Avaliación

	Descrición	Cualificación	Resultados de Formación e Aprendizaxe
Proxectos	Os alumnos, organizados en grupos de dous, entregarán o proxecto software proposto como máximo o día 5 de Xaneiro. Constará do seu deseño final (diagramas UML), o código e a documentación xerada explicativa da implementación. Que o código entregado poida ser compilado e executado nos equipos dos laboratorios é condición indispensable para superar esta proba de avaliación. Durante a última semana do curso, os alumnos terán unha entrevista co profesor no horario de laboratorio, adicada a demostrar a autoría do proxecto e realizar diversas probas de funcionalidade. Os dous membros de cada grupo deberán estar obrigatoriamente presentes en dita entrevista. As cuestións plantexadas na mesma deberán ser respondidas individualmente para poder constatar o grado de entendemento e implicación do alumno no proxecto desenrolado. En caso de que un alumno non acredite adecuadamente a autoría, a avaliación do proxecto farase mediante un exame práctico de programación individual no laboratorio docente, na data aprobada pola Xunta de Escola a tal fin. Se o alumno non se presenta a este exame práctico perderá o 30% da nota da materia. Para os alumnos que acrediten adecuadamente a autoría, a avaliación do proxecto terá en conta tanto a correcta funcionalidade como a calidade do código e o emprego das técnicas da programación orientada a obxectos.	30	B6 C50 B14 C53
Estudo de casos/análise de situacións	Ó remate da 9ª semana do curso académico os alumnos, organizados en grupos de dous, entregarán o deseño dun proxecto software.	10	C51 C52
Resolución de problemas e/ou exercicios	Exame escrito e individual, realizado na data aprobada pola Xunta de Escola para elo, que constará da combinación dos seguintes tipos de preguntas: resolución de problemas, cuestións breves para resolver aplicando os conceptos teóricos explicados en clase, xustificar razoadamente se unha ou varias afirmacións son verdadeiras ou falsas, pequenos tests sobre aspectos teóricos e de aplicación. Non se permite o emprego de apuntamentos, libros ou coleccións de problemas. O número e a combinación de ditas preguntas fixarase para cada exame en particular.	50	C50 C51 C53
Probas prácticas, de execución de tarefas reais e/ou simuladas.	Ó remate da 7ª semana do curso académico os alumnos, organizados en grupos de dous, entregarán as prácticas de iniciación en Java propostas no laboratorio.	10	C50 C51 C52 C53

Outros comentarios sobre a Avaliación

Existen dúas modalidades de avaliación da asignatura: avaliación continua (AC) y avaliación tradicional (AT). Os alumnos deberán elixir unha das dúas modalidades tendo en conta as seguintes condicións:

- A AC inclúe as 4 probas descritas no apartado avaliación.
- Tanto se optan pola AC coma se optan pola AT os alumnos deberán realizar un proxecto de laboratorio. Para facilitar a elección de AC ou AT os alumnos disporán en Faitic do proxecto a realizar a partir da 4ª semana do curso académico.
- Na AT o proxecto realizarase de forma individual.
- Os alumnos que opten pola AC deberán entregar ó remate da 9ª semana do curso académico o deseño UML do proxecto a realizar (correspondente á 2ª proba descrita no apartado de avaliación). Mediante dita entrega os alumnos comprométense a seguir a AC e renuncian á AT. Dende ese intre os alumnos non poderán figurar como ``Non presentados".
- Os alumnos que no entreguen o deseño UML do proxecto na data estipulada renuncian á AC, de modo que serán avaliados mediante a modalidade de AT. Non existe a posibilidade de sumarse á AC nas seguintes probas intermedias.
- As probas de AC non son en ningún caso recuperables, non podendo repetirse fóra das datas estipuladas polos profesores.
- Non se gardan cualificacións (de probas de AC nen de proxectos prácticos ou exames) dun curso a outro.

Primera convocatoria. Alumnos que opten pola AC

. Serán avaliados como segue:

☐ Parte teórica:

- Exame escrito (50%). Exame individual. Correspóndese coa 3ª proba descrita no apartado avaliación. A nota deste examen no se gardara en ningún caso.

☐ Parte práctica:

- Prácticas de iniciación en Java (10%). A realizar en grupos de dous. Correspóndese coa 4ª proba descrita no apartado avaliación.

- Proxecto (40%). A realizar en grupos de dous. Divídese en dúas partes:

1. Deseño (10%). Correspóndese coa 2ª proba descrita no apartado avaliación.

2. Implementación (30%). Correspóndese coa 1ª proba descrita no apartado avaliación.

☐ Os requisitos para aprobar serán:

- Un mínimo de 1/3 sobre o total na parte teórica.
- Un mínimo de 1/3 sobre o total na parte de implementación do proxecto (ou 1/3 sobre o total do examen práctico no seu caso).
- Unha nota total (suma das 4 probas) igual ou superior a 5.
- Se a nota total é igual ou superior a 5 pero non se acadou a nota mínima nalgũa parte, a nota final será 4.5 puntos (suspenso).

Primeira convocatoria. Alumnos que opten pola AT.

Serán avaliados como segue:

☐ Parte teórica:

- Exame escrito (50%). Exame individual. Correspóndese coa 3ª proba descrita no apartado avaliación. A nota deste examen no se gardara en ningún caso.

☐ Parte práctica:

- Realización individual dun proxecto software que suporá o restante 50% da nota final. Este proxecto constará do deseño (diagramas UML), o código Java e a documentación xerada explicativa da implementación. A avaliación terá en conta correcto deseño, correcta funcionalidade, calidade do código e emprego de técnicas de POO. Deberá ser entregado como máximo o día 5 de Xaneiro.

- Realización dunha entrevista co profesor adicada a demostrar a autoría do proxecto. Dita entrevista terá lugar no laboratorio durante a derradeira semana do curso. Se o alumno non acredita adecuadamente a autoría, a avaliación da

parte práctica farase por medio dun exame práctico.

□ Os requisitos para aprobar serán:

- Un mínimo de 1/3 sobre o total na parte teórica.
- Un mínimo de 1/3 sobre o total no proxecto (ou 1/3 sobre o total do exame práctico no seu caso).
- Unha nota total (suma das 2 probas) igual ou superior a 5.
- Se a nota total é igual ou superior a 5 pero non se acadou a nota mínima nalguna parte, a nota final será 4.5 puntos (suspense).

Segunda convocatoria.

Os alumnos serán avaliados como segue:

□ Parte teórica:

- Exame escrito (50%). Exame individual. Correspóndese coa 3ª proba descrita no apartado avaliación. A nota deste examen no se gardara en ningún caso.

□ Parte práctica:

Dependerá de se o alumno entregou ou non o proxecto na primeira convocatoria. Para os alumnos que seguiron a AC na primeira convocatoria, considerarase que un alumno entregou o proxecto se como mínimo entregou un deseño UML no que obtivo unha nota igual ou superior a 0.6 sobre 1.

- Os alumnos que no entreguen o proxecto na primeira convocatoria serán avaliados mediante un exame de programación individual que terá lugar no laboratorio na data fixada pola Xunta de Escola a tal fin. A avaliación deste exame suporá un 50% da nota final.

- A parte práctica a realizar polos alumnos que entreguen o proxecto na primeira convocatoria dependerá da nota obtida no proxecto en dita convocatoria, como segue:

Nota ≥ 1.5 por AC ou Nota ≥ 2.5 por AT. Manteráselles a nota, no tendo que presentarse ó exame práctico na segunda convocatoria. Poderán, nembargantes, mellorar a puntuación do proxecto entregando unha nova versión do da primeira convocatoria xunto coas novas funcionalidades a realizar, que se publicarán no seu momento en Faitic. Do mesmo xeito, deberán entregar un documento que recolla os cambios e actualizacións realizados no proxecto sobre a versión entregada na primeira convocatoria.

Nota entre 1 e 1.5 por AC ou Nota entre 5/3 e 2.5 por AT. Poderán optar entre realizar o exame práctico ou o proxecto ampliado da segunda convocatoria. Non se lles manterá a nota do proxecto da primeira convocatoria, pero si a das partes de iniciación en Java e deseño UML, se optaron pola AC na primeira convocatoria.

Nota < 1 por AC ou Nota $< 5/3$ por AT. Poderán optar entre realizar o exame práctico ou o proxecto ampliado da segunda convocatoria. En calquera caso pérdense as notas das partes de iniciación en Java e deseño UML se optaron pola AC na primeira convocatoria, é dicir, serán avaliados sobre 5.

□ Os requisitos para aprobar serán:

- Un mínimo de 1/3 sobre o total na parte teórica.
- Un mínimo de 1/3 sobre o total no proxecto sen ter en conta a nota de iniciación en Java e deseño UML se optaron pola AC na primeira convocatoria (ou 1/3 sobre o total do examen práctico no seu caso).
- Unha nota total (suma de todas as probas) igual ou superior a 5.
- Se a nota total é igual ou superior a 5 pero no se acadou a nota mínima nalgunha parte, a nota final será 4.5 puntos (suspense).

Bibliografía. Fontes de información

Propónse a seguinte bibliografía organizada en dous grandes grupos: manuais básicos e referencias adicionais.

Manuais básicos:

[1] □Absolute Java□. W. Savitch, 4ª edición. 2010, Pearson.

[2] [Introduction to Java programming]. Y. D. Liang, 8ª edición. 2010, Pearson.

[3] [Java: How to program]. P. Deitel, H. Deitel, 9ª edición. 2011, Pearson.

Referencias adicionales:

[1] [Programación orientada a objetos con Java: Una introducción práctica usando BlueJ]. D. J. Barnes, M. Kölling, 3ª edición. 2007, Pearson.

[2] [The Java Tutorial. A short course on the basics]. S. Zakhour, S. Hommel, J. Royal, I. Rabinovitch, T. Risser, M. Hoeber, 4ª edición. 2006, Prentice-Hall.

[3] [Data Structures & Algorithms in Java]. M. T. Goodrich, R. Tamassia, 5ª edición. 2010, Willey.

[4] [Java Tools]. A. Eberhart, S. Fischer. 2002, Wiley.

[5] [Java in a Nutshell]. D. Flanagan, 5ª edición. 2005, O'Reilly.

[6] [Thinking in Java]. B. Eckel, 4ª edición. 2006, Prentice-Hall.

[7] [Learning Java]. P. Niemeyer, D. Leuck, 4ª edición. 2013, O'Reilly.

[8] [How to Think Like a Computer Scientist. JavaTM Version]. 4ª versión. Online:
<http://www.greenteapress.com/thinkapjava/>

[9] [Java notes]. F. Swartz. Online: <http://www.leepoint.net/notes-java/index.html>

[10] [Java SE. Oracle]. Online: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

[11] [Java 2 Platform Standard Edition 5.0. API Specification]. Online: <http://download.oracle.com/javase/1.5.0/docs/api/>

[12] [The Java Tutorials]. Oracle. Online: <http://download.oracle.com/javase/tutorial/>

[13] [Ingeniería del Software orientada a objetos con UML, Java e Internet]. A. Weitzenfeld. 2005, Thomson.

[14] [Open-oriented Analysis and Design with Applications]. G. Booch, R. Maksimchuk, M. Engel, B. Young, J. Conallen, K. Houston, 3ª edición. 2007, Addison-Wesley.

[15] [The Unified Modeling Language User Guide]. G. Booch, J. Rumbaugh, I. Jacobson, 2ª edición. 2005. Addison-Wesley.

[16] [UML Distilled: A Brief Guide to the Standard Object Modeling Language]. M. Fowler, 3ª edición. 2003, Addison-Wesley.

[17] [Fundamentals of object-oriented design in UML]. M. Page-Jones. 2002, Addison-Wesley.

Recomendacións

Materias que se recomenda ter cursado previamente

Programación I/V05G300V01205
