



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 Manso Vázquez, Mario Servia Rodríguez, Sandra			
Correo-e	francisco.fernandez@det.uvigo.es			
Web	http://www.faitic.es			

Descrición xeral O obxectivo xeral da materia é proporcionar ao estudante os fundamentos teóricos e as competencias prácticas que lle permitan analizar, deseñar, desenvolver e depurar aplicacións informáticas seguindo a paradigma orientado a obxectos. Esta é unha materia eminentemente práctica e neste sentido está orientada ao traballo dos alumnos na realización dun ou varios proxectos. Para facilitar o desenvolvemento dos proxectos na materia tamén se fai unha introdución 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 espregamento dos sistemas, senón que se tratarán principalmente as etapas de análises, deseño, implementación e depuración. En primeiro lugar presentárase a enxeñaría do software como disciplina imprescindible para o desenvolvemento de grandes aplicacións informáticas, mostrando os principais retos aos que se enfronta e os conceptos básicos que se utilizarán. A continuación analizaranse os elementos da paradigma orientado a obxectos utilizando elementos e diagramas UML que serán utilizados polos alumnos nos seus desenvolvementos. Para alcanzar este obxectivo xeral os contidos que se verán na materia pódense resumir nos seguintes ítems:

A paradigma Orientado a Obxectos.

Conceptos básicos da orientación a obxectos: clases e obxectos

Encapsulación. Principio de ocultación. Conceptos de desacoplamiento e cohesión

Herdanza, abstracción, polimorfismo e reutilización.

Relacións entre clases: Generalización, asociación e dependencia

Comunicación entre obxectos: métodos, eventos, mensaxes

Persistencia. Almacenamento en ficheiros e en bases de datos

Xeración, 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 aos procesos de desenvolvemento de software orientado a obxectos.

Fases principais no desenvolvemento OO: análise, deseño, implementación e probas

Introdución á linguaxe de modelado UML: estrutura e interacción

Competencias de titulación

Código

A6	CG6 Facilidade para o manexo de especificacións, regulamentos e normas de obrigado cumprimento.
A9	CG9 Capacidade para traballar nun grupo multidisciplinar e nunha contorna multilingüe e de comunicar, tanto por escrito como de forma oral, coñecementos, procedementos, resultados e ideas relacionadas coas telecomunicacións e a electrónica.

A59	(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.
A60	(CE51/T19) Capacidade de a aplicación básica das fases de análises, deseño, implantación e depuración de programas na POO.
A61	(CE52/T20) Capacidade de manexo de ferramentas CASE (editores, depuradores).
A62	(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.
B5	CG14 Capacidade para utilizar ferramentas informáticas de procura de recursos bibliográficos ou de información.

Competencias de materia

Resultados previstos na materia	Resultados de Formación e Aprendizaxe
Comprender os aspectos fundamentais da Programación Orientada a Obxectos (POO) e levalos a práctica usando a linguaxe de programación máis representativo (Xava).	A9 A59
Introducir no uso da linguaxe UML, linguaxe estándar de modelado de software, para a realización de diagramas de estrutura, comportamento e interacción, fundamental para a documentación nas fases de análise e deseño de programas de acordo á POO.	A6 A61 A62
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.	A60 A62
Adquirir madurez en técnicas de desenvolvemento e depuración de programas para permitir a aprendizaxe autónoma de novas capacidades e linguaxes de programación.	A62
Adquirir familiaridade co uso dunha contorna moderna de desenvolvemento de software (Eclipse) para facilitar o deseño, desenvolvemento e depuración de programas.	A60 A61

Contidos

Tema	
1. Introducción ao paradigma OO	a. Breve introdución á materia e a súa organización b. Nacemento do paradigma c. Bases: clases e obxectos d. Conceptos de encapsulación, herdanza (generalización), e polimorfismo e. Breve introdución a UML e PUM
2. Encapsulación	a. Clases, interfaces e paquetes b. Métodos e variables membro. Visibilidade. Resolución de ámbito. c. Método constructor d. Paso de parámetros: punteros e referencias e. Punteros a objetos
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. Utilización de diagramas UML
5. Polimorfismo	a. Sobrecarga e sobreescritura b. Clases abstractas e interfaces c. Clases genéricas
6. Xestión de excepcións	a. Fundamentos de excepcións b. Manipulación de excepcións en Xava

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
Probos 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. Estes 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 esta orientada a adquisición das competencias CE50, CE51 e CE53.
Resolución de problemas e/ou exercicios	No laboratorio, o profesor exporá 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 esta orientada as 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 lle expoña no laboratorio. As solucións e as dúbidas que xurdan ao abordar devanditos problemas serán postas en común para acordar a mellor forma de resolución. Esta metodoloxía esta orientada ás competencias CE50, CE51, CE53 e A9.
Estudo de casos/análises de situacións	Posta en común dos deseños expostos para solucionar o proxecto que han de 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 esta orientada ás competencias CE51 e CE52.
Proxectos	Os alumnos implementarán o sistema software exposto polo profesor. Disporá para iso da segunda parte do curso combinando traballo presencial no laboratorio co traballo fose do laboratorio. Esta metodoloxía esta orientada ás competencias CE50, CE53, A6 e A9.

Atención personalizada

Metodoloxías	Descrición
Resolución de problemas e/ou exercicios	A atención individualizada articularase co seguimento do traballo de cada alumno, monitorizando as solucións que propón para cada problema exposto nas prácticas de laboratorio, a exposición das mesmas que realice aos seus compañeiros e o seguimento do proxecto software que debe implementar.
Proxectos	A atención individualizada articularase co seguimento do traballo de cada alumno, monitorizando as solucións que propón para cada problema exposto nas prácticas de laboratorio, a exposición das mesmas que realice aos seus compañeiros e o seguimento do proxecto software que debe implementar.
Resolución de problemas e/ou exercicios de forma autónoma	A atención individualizada articularase co seguimento do traballo de cada alumno, monitorizando as solucións que propón para cada problema exposto nas prácticas de laboratorio, a exposición das mesmas que realice aos seus compañeiros e o seguimento do proxecto software que debe implementar.
Estudo de casos/análises de situacións	A atención individualizada articularase co seguimento do traballo de cada alumno, monitorizando as solucións que propón para cada problema exposto nas prácticas de laboratorio, a exposición das mesmas que realice aos seus compañeiros e o seguimento do proxecto software que debe implementar.

Avaliación

	Descrición	Cualificación
Proxectos	Os alumnos, organizados en grupos de 2 persoas, entregarán o proxecto software proposto durante a semana do 2 a o 6 de Decembro. Este 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 docentes é chave para superar esta avaliación. Os docentes valorarán en igual proporción o funcionamento do código entregado e o deseño utilizado para a implementación. Con esta proba avaliaranse as competencias CE53, CE50, A6 (CG6), A9(CG9) e B5.	30
Estudo de casos/análise de situacións	Os alumnos, organizados en grupos de 2 persoas, haberán de entregar o deseño dun proxecto software. Entregarase na semana do 4 a o 7 de Novembro. Con esta proba avaliaranse as competencias CE51, CE52 e A9.	10
Resolución de problemas e/ou exercicios	Exame escrito e individual, realizado na data aprobada por Xunta de Escola para iso, que constará da combinación dos seguintes tipos de preguntas: resolución de problemas, cuestións breves para resolver aplicando os conceptos teóricos explicados en clase, xustificar se unha ou varias afirmacións son verdadeiras ou falsas, pequenos tests sobre aspectos teóricos e de aplicación. Non se permite a utilización de apuntamentos, libros nin coleccións de problemas. O número e a combinación das devanditas preguntas fixarase para cada exame en particular. Con esta proba avaliaranse as competencias CE50, CE51 e CE53.	50
Probas prácticas, de execución de tarefas reais e/ou simuladas.	Na semana do 21 a o 25 de Outubro do período docente, os alumnos, organizados en grupos de 2 persoas, entregarán as prácticas de iniciación en Java propostas no laboratorio. Con esta proba avaliaranse as competencias CE50, CE52 e CE53.	10

Outros comentarios sobre a Avaliación

Existen dúas modalidades na avaliación da materia: avaliación continua (EC) e avaliación tradicional (ET).

Os alumnos deberán elixir unha das dúas modalidades tendo en conta as seguintes restricións:

- A EC inclúe as 4 probas descritas na apartado avaliación.
- Tanto por EC como por ET, os alumnos deberán realizar un proxecto de laboratorio. Para facilitar a elección de EC ou ET os alumnos disporán en Faitic do proxecto a realizar a partir do día 20 de Setembro.
- En ET o proxecto realizarase de forma individual.
- Os alumnos que opten pola EC deberán entregar na primeira semana de Novembro, o deseño UML do proxecto exposto no laboratorio (correspondente á 3ª proba de avaliación). Mediante dita entrega os alumnos comprométese a seguir a EC e renuncian a ET. Desde ese momento, estes estudantes non poderán figurar como "Non presentados".
- Os alumnos que non entreguen o deseño UML do proxecto na semana do 4 ao 7 de Novembro, renuncian á EC, de modo que serán avaliados mediante o mecanismo de ET. Non existe a posibilidade de sumarse á EC nas seguintes probas intermedias.
- As probas de EC non serán en ningún caso recuperables, non podendo repetirse fóra das datas estipuladas polos docentes.
- Non se gardarán cualificacións (de probas de EC nin de proxectos prácticos ou exames finais) dun curso a outro.
- A EC só se aplicará na primeira convocatoria, no resto de convocatorias rexe unicamente a ET.

Primeira convocatoria. Alumnos que opten pola EC. Serán avaliados como segue:

- Parte teórica:
 - Exame escrito (50%). Exame individual. Correspóndese coa proba 3 descrita na apartado "Avaliación". Non se permitirá material de apoio.
- Parte práctica:
 - Prácticas de iniciación en Xava (10%). En grupos de 2 alumnos. Correspóndese coa proba 4 descrita na apartado "Avaliación".
 - Proxecto (40%). En grupos de 2 alumnos. Desagregase en dous partes:
 - Deseño (10%). Correspóndese coa proba 2 descrita na apartado "Avaliación".
 - Implementación (30%). Correspóndese coa proba 3 descrita na apartado "Avaliación". Este proxecto deberá ser entregado individualmente a primeira semana de decembro do período docente. Para a súa avaliación realizarase, como requisito previo, unha proba ou entrevista de autoría:
 - Se o alumno non a supera, a avaliación da implementación realizácese por un exame practico.
 - Se o alumno supera a proba de autoría, a súa nota de avaliación (que será a mesma para ambos os membros do grupo) terá en conta: correcto deseño, correcta funcionalidade, calidade do código e uso de técnicas de POO.
- 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 exame práctico no seu caso).
 - Unha nota total (suma das 4 probas) igual ou superior a 5.

Para a proba de autoría da parte práctica (que pode supor preguntas individuais de diversa natureza) será obrigatorio que o código entregado poida ser compilado e executado nos equipos dos laboratorios docentes.

Primeira convocatoria. Alumnos que opten pola ET. Serán avaliados como segue:

- Parte teórica:
 - Un exame escrito (cuxa descrición coincide coa proba 3 da EC). O resultado deste exame suporá un 50%

da cualificación final. Non se permitirá material de apoio.

- Parte práctica:
 - A realización dun proxecto software que suporá o outro 50% da cualificación final. De realización individual. Este proxecto constará de deseño (diagramas UML), o código Xava e a documentación xerada explicativa da implementación. A nota de avaliación terá en conta: correcto deseño, correcta funcionalidade, calidade do código e uso de técnicas de POO. Este proxecto deberá ser entregado individualmente a primeira semana de decembro.
 - A realización dunha entrevista co profesor titor orientada a determinar a autoría do proxecto. Dita entrevista terá lugar no laboratorio na última semana lectiva do curso. Se o alumno non supera a proba de autoría deberá ir a un exame practico.
- 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 exame práctico segundo o caso.
 - Unha nota total (suma das 2 probas) igual ou superior a 5.

Para a proba de autoría da parte práctica (que pode supor preguntas individuais de diversa natureza) será obrigatorio que o código entregado poida ser compilado e executado nos equipos dos laboratorios docentes.

Segunda convocatoria / Convocatoria de fin de curso / Convocatoria extraordinaria.

Nesta convocatoria que non rexe a EC. A avaliación será como segue:

- Parte teórica:
 - Un exame escrito (cuxa descrición coincide coa proba 3 de a EC). O resultado deste exame suporá o 50% de a cualificación final. Non se permitirá material de apoio.
- Parte práctica:
 - Os alumnos que non entreguen o proxecto na primeira convocatoria, avaliaranse cun exame de programación individual no laboratorio que terá lugar na data fixada pola Xunta de Escola para iso. A avaliación desta proba suporá un 50% da cualificación final.
 - A parte practica a realizar para os alumnos que entreguen o proxecto na primeira convocatoria, dependerá da nota do proxecto obtida na primeira convocatoria, segundo o seguinte:
 - *Nota >= 1,5.* Manteráselles a nota, non tendo que presentarse ao exame práctico da segunda convocatoria. Poderán, con todo, mellorar a puntuación do proxecto entregando unha nova versión da da primeira convocatoria xunto coa(s) novas funcións a realizar que se publicasen no seu momento en Faitic. Así mesmo, deberán entregar un documento que recolla os cambios e actualizacións realizadas ao proxecto sobre a versión que entreguen na primeira convocatoria.
 - *Nota entre 1.5 e 1.* Poderán optar entre ir ao exame practico ou realizar o proxecto da segunda convocatoria. Non se lles mantén a nota do proxecto da primeira convocatoria, pero se a de iniciación e a de deseño UML.
 - *Nota < 1.* Poderán optar entre ir directamente ao exame practico ou realizar o proxecto ampliado. En calquera caso pérdese a nota de prácticas da parte de iniciación e deseño UML. É dicir, avaliáseselles sobre 5, independentemente de se optan polo proxecto ou o 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 exame segundo o caso.
 - Unha nota total (suma das 2 probas) igual ou superior a 5.

Bibliografía. Fontes de información

Manuais básicos

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

Referencias adicionales

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

[3] *Data Structures & Algorithms in Java*. Michale T. Goodrich, Roberto Tamassia, 5ª edición. 2010, Willey.

[4] *Java Tools*. Andreas Eberhart, Stefan Fischer. 2002, Wiley

[5] *Java In A Nutshell*. David Flanagan, 5ª edición. 2005, O'Reilly.

[6] *Thinking in Java*. Bruce Eckel, 4ª edición. 2006, Prentice Hall

[7] *Learning Java*. Patrick Niemeyer, 3ª edición. O'Reilly Media

[8] *How to Think Like a Computer Scientist. Java™ Version*. 4ª version. Online:
<http://www.greenteapress.com/thinkajava/>

[9] *Java notes*. Fred 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/>

[14] *Open-oriented Analysis and Design with Applications*. Grady Booch, Robert Maksimchuk, Michael Engel, Bobbi Young, Jim Conallen, Kelli Houston, 3ª edición. 2007, Addison Wesley.

[17] *Fundamentals of Object-oriented design in UML*. Meilir Page-Jones. 2002, Addison Wesley.

Recomendacións

Materias que se recomienda ter cursado previamente

Programación I/V05G300V01205
