



Teaching Guide				
Identifying Data				2015/16
Subject (*)	Programación Orientada a Obxectos	Code	614111636	
Study programme	Enxeñeiro en Informática			
Descriptors				
Cycle	Period	Year	Type	Credits
First and Second Cycle	1st four-month period	All	Optativa	4
Language	Spanish			
Teaching method	Face-to-face			
Prerequisites				
Department	Computación			
Coordinador	Alonso Ríos, David	E-mail	david.alonso@udc.es	
Lecturers	Alonso Ríos, David	E-mail	david.alonso@udc.es	
Web				
General description	<p>A programación orientada a obxectos (POO) é, hoxe en día, a paradigma de programación dominante no desenvolvemento de sistemas informáticos. A POO xorde como un paso máis na evolución da programación imperativa engadindo novas propiedades (como herdanza, polimorfismo, etc.) aos tipos abstractos de datos. O carácter optativo da materia pode levar a engano xa que a relativa antigüidade dos plans de estudo (1994) e das troncalidades (1991) non recollen correctamente a situación actual no ámbito da programación (Java só ten pouco máis de 10 anos de antigüidade).</p> <p>A filosofía de aprendizaxe marcada no plan de estudos configúrase como "Imperativo primeiro", é dicir, primeiro explícanse as linguaxes imperativas para dar paso máis adiante á programación orientada a obxectos. Desta forma a aprendizaxe da programación faise máis gradual, aínda que o paso aos conceptos da orientación a obxectos obriga a un cambio de filosofía na forma de programar cuxa adaptación pode resultar complexa ao alumno.</p> <p>As competencias académicas que se pretenden desenvolver son importantes para cursar outras materias ligadas directa ou indirectamente coa programación. Entre as máis directamente relacionadas podemos destacar: (II, ITIG e ITIS) Proxecto fin de Carreira, (II) Análise de Sistemas Informáticos, Deseño de Sistemas Informáticos, Integración de Sistemas, Análises e Deseño Orientado a Obxectos, e (ITIG) Principios de Análise Informática.</p> <p>Esta materia tamén resulta moi interesante á hora de configurar habilidades no contexto do ámbito profesional xa que a paradigma da orientación a obxectos é o dominante dentro das linguaxes de programación máis utilizados profesionalmente (Java, C#, C++, VisualBasic, Delphi, etc.).</p>			

Study programme competences / results	
Code	Study programme competences / results
A1	Aprender de maneira autónoma novos coñecementos e técnicas avanzadas axeitadas para a investigación, o deseño e o desenvolvemento de sistemas e servizos informáticos.
A3	Concibir e planificar o desenvolvemento de aplicacións informáticas complexas ou con requisitos especiais.
A6	Avaliar, definir, seleccionar e auditar plataformas hardware e software para a execución e desenvolvemento de aplicacións e servizos informáticos.
A7	Saber especificar, deseñar e implementar un sistema de información, empregando bases de datos.
B1	Aprender a aprender.
B2	Resolver problemas de forma efectiva.
B3	Aplicar un pensamento crítico, lóxico e creativo.
B4	Aprendizaxe autónoma.
B5	Traballar de forma colaborativa.
B6	Comportarse con ética e responsabilidade social como cidadán e como profesional.
B7	Comunicarse de maneira efectiva en calquera contorno de traballo.
B9	Capacidade para tomar decisións.



B10	Capacidade de xestión da informática (captación e análises da información).
B12	Capacidade para a análise e a síntese.
B15	Motivación pola calidade.
C3	Utilizar as ferramentas básicas das tecnoloxías da información e as comunicacións (TIC) necesarias para o exercicio da súa profesión e para a aprendizaxe ao longo da súa vida.
C4	Desenvolverse para o exercicio dunha cidadanía aberta, culta, crítica, comprometida, democrática e solidaria, capaz de analizar a realidade, diagnosticar problemas, formular e implantar solucións baseadas no coñecemento e orientadas ao ben común.
C6	Valorar criticamente o coñecemento, a tecnoloxía e a información dispoñible para resolver os problemas cos que deben enfrontarse.

Learning outcomes			
Learning outcomes	Study programme competences / results		
Coñecer as diferenzas entre a paradigma de programación imperativa tradicional e a paradigma da orientación a obxectos.	A1	B1	C3
Comprender os conceptos básicos nos que se basea a orientación a obxectos: clases, obxectos, mensaxes, etc.	A1	B1 B2 B4	C3
Analizar, comprender e relacionar as propiedades básicas propias da orientación a obxectos: herdanza, polimorfismo, ligadura dinámica, etc.	A1	B1 B2 B3 B4	C3
Coñecer os aspectos básicos do deseño UML, sobre todo o que involucra o deseño estático (diagrama de clases) e dinámico (diagrama de secuencia) dun programa orientado a obxectos.	A1 A3	B1 B2 B3 B4	C3
Coñecer, comprender e identificar os principios e patróns de deseño básicos para lograr un programa orientado a obxectos flexible e reusable.	A1 A3	B1 B2 B3 B4	C3
Construír clases básicas que representen obxectos do dominio e que encapsulen o estado e o comportamento dos devanditos obxectos.	A1	B2 B3 B4 B9 B10 B12	C3
Desenvolver e organizar clases sacando proveito das propiedades propias da orientación a obxectos para desenvolver código flexible e reusable.	A1 A3	B2 B3 B4 B9 B10 B12	C3
Deseñar un programa orientado a obxectos utilizando o estándar UML e desenvolver posteriormente o código resultante do devandito deseño.	A1 A3 A7	B2 B3 B4 B5 B9 B10 B12	C3



Deseñar un programa utilizando principios e patróns de deseño garantindo así que as propiedades propias da orientación a obxectos utilízanse adecuadamente.	A1 A3	B2 B3 B4 B5 B9 B10 B12	C3
Usar a programación orientada a obxectos para resolver problemas reais.	A1	B2 B3 B4 B9 B10 B12	C3 C6
Manexar con solvencia unha contorna integrada de desenvolvemento (IDE) na programación.	A1 A6	B2 B3 B4	C3
Asumir a responsabilidade do traballo e as accións propias.		B5 B6	C4
Valorar e respectar o traballo propio e o alleo.		B5 B6	C4
Colaborar cos compañeiros para levar a cabo unha práctica en grupo.		B5 B7	C4
Valorar a importancia da análise e o deseño no desenvolvemento de programas orientados a obxectos.		B10 B15	C3 C6
Preferir as boas prácticas de programación orientada a obxectos sobre os antipatrones (malas prácticas) propias dos principiantes ou dos malos programadores.		B15	C6

Contents	
Topic	Sub-topic
EXAME DE TEORÍA	TEMARIO DOS ÚLTIMOS ANOS DA MATERIA (resumido a continuación). Para máis información consultar o Moodle ou falar co profesor.
Introdución	Paradigmas da programación Programación orientada a obxectos A linguaxe Xava
Modelado visual de obxectos: UML	Introdución Elementos básicos do UML Deseño estático: diagrama de clases Deseño dinámico: diagrama de secuencia
Elementos básicos da orientación a obxectos	Clases Obxectos
Propiedades básicas da orientación a obxectos	Abstracción Encapsulamiento Modularidad Xerarquía Polimorfismo Tipificación Ligadura dinámica



Patróns de deseño	<p>Introdución</p> <p>Patróns creacionais: inmutable, singleton</p> <p>Patróns de comportamento: estratexia, estado, iterador, observador, método persoal</p> <p>Patróns estruturais: composición, adaptador</p>
Principios de deseño	<p>Principio aberto-pechado</p> <p>Principio de substitución de Liskov</p> <p>Deseño por contrato e principio de subcontratación</p> <p>Principio de inversión da dependencia</p> <p>Principios de deseño e herdanza</p>
Ferramentas	<p>NetBeans</p> <p>JUnit</p>
EXAME DE PRÁCTICAS	<p>TEMARIO: SELECCIÓN DE CONCEPTOS CRAVE DE POO RELACIONADOS COA APLICACIÓN PRÁCTICA DO TEMARIO DE TEORÍA E COS EXERCICIOS PRÁCTICOS EXPOSTOS NOS ÚLTIMOS ANOS DA MATERIA. Para máis información consultar o Moodle ou falar co profesor.</p>

Planning				
Methodologies / tests	Competencies / Results	Teaching hours (in-person & virtual)	Student?s personal work hours	Total hours
Objective test	A1 A3 A6 A7 B1 B2 B3 B4 B5 B6 B7 B9 B10 B12 B15 C3 C4 C6	5	90	95
Personalized attention		5	0	5

(*)The information in the planning table is for guidance only and does not take into account the heterogeneity of the students.

Methodologies	
Methodologies	Description
Objective test	<p>Exame escrito, dividido en:</p> <ul style="list-style-type: none"> - Exame de teoría. - Exame de prácticas. <p>Observacións: En proceso de extinción non hai ningún tipo de avaliación continua. Non se recollerán exercicios prácticos, sólo haberá exames.</p> <p>Para máis información consultar o Moodle ou falar co profesor.</p>

Personalized attention	
Methodologies	Description
Objective test	Resolución de dúbidas

Assessment			
Methodologies	Competencies / Results	Description	Qualification



Objective test	A1 A3 A6 A7 B1 B2 B3 B4 B5 B6 B7 B9 B10 B12 B15 C3 C4 C6	<p>Exame de teoría:</p> <ul style="list-style-type: none"> - Cualificación mínima esixida: 4 de 10. - 60% da cualificación global. - A cualificación desta proba escrita se garda entre convocatorias se é un 5 ou máis. <p>Existe tamén a posibilidade de facer de novo o exame para mellorar a cualificación (con isto, o alumno renunciaría á anterior cualificación).</p> <p>Exame de prácticas:</p> <ul style="list-style-type: none"> - Proba escrita da parte de práctica. - Cualificación mínima esixida: 4 de 10. - 40% da cualificación global. - A cualificación desta proba escrita se garda entre convocatorias se é un 5 ou máis. <p>Existe tamén a posibilidade de facer de novo o exame para mellorar a cualificación (con isto, o alumno renunciaría á anterior cualificación).</p>	100
Others			

Assessment comments

Cualificación en actas:- Aprobado ou superior: Cualificación final maior ou igual que 5 e cumprir todos os requerimentos esixidos para superar a materia.- Suspenso: Cualificación final menor que 5 ou cualificación final maior ou igual que 5 e non cumprir todos os requerimentos esixidos para superar a materia. No segundo caso, a cualificación en actas será 4,5.- Non Presentado: Equivale a non presentado a ambos exames finais (teoría e prácticas).

Para máis información consultar o Moodle ou falar co profesor.

Sources of information

Basic	<p>Eckel, B.. Piensa en Java. Prentice-Hall, Madrid. 4ª Edición. 2007. Signatura FIC: D32 (Jav) ECK. Sierra, K., Bates, B. Head First Java, O'Reilly, Sebastopol, CA, 2nd Edition, 2005. Cohoon, J., Davidson, J., Programación en Java 5.0, McGraw-Hill, Madrid, 2006. Booch, G., Rumbaugh, J. y Jacobson. El lenguaje unificado de modelado, 2ª Ed.. Addison-Wesley, Madrid. 2006. Signatura FIC: D22 BOO. Gamma, E. et al. Design patterns: elements of reusable object oriented software. Reading, MA. Addison-Wesley. 1995. FIC: D211 DES - D123 PAT (español).</p>
Complementary	<p>McLaughlin, B., Flanagan, D., Java 1.5 Tiger: A Developer's Notebook, O'Reilly, Sebastopol, CA, 2004. Martin, R.C. UML para programadores Java, Pearson, Madrid, 2004. Rumbaugh, J., Jacobson, I., Booch, G., El lenguaje unificado de modelado: Manual de referencia, Addison-Wesley Iberoamericana, Madrid, 2000. Budd, T.. An introduction to object-oriented programming. Pearson. 3ª Edición. 2002. Signatura FIC: D15 BUD. Knoernschild, K.. Java desing: objects, UML and process. Addison-Wesley, Boston, MA. 2002. Signatura FIC: D32 (Jav) KNO. Recursos web O principal recurso web da materia é a páxina da mesma existente na facultade virtual da UDC. Na devandita páxina atoparase toda a información necesaria para o desenvolvemento da materia (transparencias, boletíns de exercicios, prácticas, tests, preguntas frecuentes, links en Internet, etc.). O acceso á devandita páxina está limitado aos alumnos matriculados na materia. Para información xeral sobre Java recoméndase acudir á páxina web oficial da linguaxe (http://java.sun.com) onde poderemos acceder a noticias, documentación, descarga das ferramentas, etc. Para estar ao día sobre as novidades da linguaxe recoméndase acceder a páxinas de noticias por exemplo JavaHispano (http://javahispano.org). Outros materiais de apoio A biblioteca da Facultade dispón de varias revistas científicas que están a disposición do alumno, e que revisten interese para a profundización nalgún aspecto particular do temario, ou para dispor de actualizacións (estados da arte) dalgunha das técnicas e metodoloxías expostas en clase. Algunhas das máis utilizadas na materia de POO son: Communications of the ACM Journal of Object Oriented Programming (xa no se publica pero hai números antigos) IEEE Transactions on Knowledge and Data Engineering Data and Knowledge Engineering ACM transactions on programming languages and systems Lecture Notes in Computer Science (aqueles volumes que fan referencia a congresos sobre obxectos como ECOOP ? European Conference on Object Oriented Programming).</p>



Recommendations

Subjects that it is recommended to have taken before

Estrutura de Datos e da Información/614111102

Programación/614111109

Subjects that are recommended to be taken simultaneously

Metodoloxía da Programación/614111205

Subjects that continue the syllabus

Análise de Sistemas Informáticos/614111402

Deseño de Sistemas de Información/614111403

Integración de Sistemas/614111503

Proxecto fin de Carreira/614111505

Análise e Deseño Orientado a Obxectos/614111602

Other comments

(*)The teaching guide is the document in which the URV publishes the information about all its courses. It is a public document and cannot be modified. Only in exceptional cases can it be revised by the competent agent or duly revised so that it is in line with current legislation.