UNIVERSIDADE DA CORUÑA

Guia docente				
Datos Identificativos				2016/17
Asignatura (*)	Programación Orientada a Objectos		Código	614111636
Titulación	Enxeñeiro en Informática			
Descriptores				
Ciclo	Periodo	Curso	Tipo	Créditos
1º y 2º Ciclo	1º cuatrimestre	Todos	Optativa	4
Idioma	Castellano			
Modalidad docente	Presencial			
Prerrequisitos				
Departamento	Computación			
Coordinador/a	Alonso Ríos, David	Correo electrónico	david.alonso@udc.es	
Profesorado	Alonso Ríos, David	Correo electrónico	david.alonso@udc.es	
Web				
Descripción general	La programación orientada a objetos (POO) es, hoy en día, el paradigma de programación dominante en el desarrollo de sistemas informáticos. La POO surge como un paso más en la evolución de la programación imperativa añadiendo nuevas propiedades (como herencia, polimorfismo, etc.) a los tipos abstractos de datos. El carácter optativo de la asignatura puede llevar a engaño ya que la relativa antigüedad de los planes de estudio (1994) y de las troncalidades (1991) no recogen correctamente la situación actual en el ámbito de la programación (Java sólo tiene poco más de 10 años de antigüedad). La filosofía de aprendizaje marcada en el plan de estudios se configura como ?Imperativo primero?, es decir, primero se explican los lenguajes imperativos para dar paso más adelante a la programación orientada a objetos. De esta forma el aprendizaje de la programación se hace más gradual, aunque el paso a los conceptos de la orientación a objetos obliga a un cambio de filosofía en la forma de programar cuya adaptación puede resultar compleja al alumno. Las competencias académicas que se pretenden desarrollar son importantes para cursar otras materias ligadas directa o indirectamente con la programación. Entre las más directamente relacionadas podemos destacar: (II, ITIG e ITIS) Proyecto fin de Carrera, (II) Análisis de Sistemas Informáticos, Diseño de Sistemas Informáticos, Integración de Sistemas, Análisis y Diseño Orientado a Objetos, e (ITIG) Principios de Análisis Informático. Esta materia también resulta muy interesante a la hora de configurar habilidades en el contexto del ámbito profesional ya que el paradigma de la orientación a objetos es el dominante dentro de los lenguajes de programación más utilizados profesionalmente (Java, C#, C++, VisualBasic, Delphi, etc.).			

Competencias / Resultados del título	
Código	Competencias / Resultados del título
A1	Aprender de manera autónoma nuevos conocimientos y técnicas avanzadas adecuadas para la investigación, el diseño y el desarrollo de sistemas y servicios informáticos.
A3	Concebir y planificar el desarrollo de aplicaciones informáticas complejas o con requisitos especiales.
A6	Evaluar, definir, seleccionar y auditar plataformas hardware y software para la ejecución y desarrollo de aplicaciones y servicios informáticos.
A7	Saber especificar, diseñar e implementar un sistema de información, empleando bases de datos.
B1	Aprender a aprender.
B2	Resolver problemas de forma efectiva.
B3	Aplicar un pensamiento crítico, lógico y creativo.
B4	Aprendizaje autónomo.
B5	Trabajar de forma colaborativa.
B6	Comportarse con ética y responsabilidad social como ciudadano y como profesional.
B7	Comunicarse de manera efectiva en cualquier entorno de trabajo.

B9	Capacidad para tomar decisiones.
B10	Capacidad de gestión de la informática (captación y análisis de la información).
B12	Capacidad para el análisis y la síntesis.
B15	Motivación por la calidad.
C3	Utilizar las herramientas básicas de las tecnologías de la información y las comunicaciones (TIC) necesarias para el ejercicio de su profesión y para el aprendizaje a lo largo de su vida.
C4	Desarrollarse para el ejercicio de una ciudadanía abierta, culta, crítica, comprometida, democrática y solidaria, capaz de analizar la realidad, diagnosticar problemas, formular e implantar soluciones basadas en el conocimiento y orientadas al bien común.
C6	Valorar críticamente el conocimiento, la tecnología y la información disponible para resolver los problemas con los que deben enfrentarse.

Resultados de aprendizaje			
Resultados de aprendizaje	Competencias / Resultados del título		
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

Contenidos	
Tema	**Subtema**
EXAMEN DE TEORÍA	TEMARIO DE LOS ÚLTIMOS AÑOS DE LA ASIGNATURA (resumido a continuación). Para más información consultar el Moodle o hablar con el profesor.
Introducción	Paradigmas de la programación Programación orientada a objetos El lenguaje Java
Modelado visual de objetos: UML	Introducción Elementos básicos del UML Diseño estático: diagrama de clases Diseño dinámico: diagrama de secuencia
Elementos básicos de la orientación a objetos	Clases Objetos
Propiedades básicas de la orientación a objetos	Abstracción Encapsulamiento Modularidad Jerarquía Polimorfismo Tipificación Ligadura dinámica

Patrones de diseño	Introducción Patrones creacionales: inmutable, singleton Patrones de comportamiento: estrategia, estado, iterador, observador, método plantilla Patrones estructurales: composición, adaptador
Principios de diseño	Principio abierto-cerrado Principio de sustitución de Liskov Diseño por contrato y principio de subcontratación Principio de inversión de la dependencia Principios de diseño y herencia
Herramientas	NetBeans JUnit
EXAMEN DE PRÁCTICAS	TEMARIO: SELECCIÓN DE CONCEPTOS CLAVE DE POO RELACIONADOS CON LA APLICACIÓN PRÁCTICA DEL TEMARIO DE TEORÍA Y CON LOS EJERCICIOS PRÁCTICOS PLANTEADOS EN LOS ÚLTIMOS AÑOS DE LA ASIGNATURA. Para más información consultar el Moodle o hablar con el profesor.

Planificación				
Metodologías / pruebas	Competencias / Resultados	Horas lectivas (presenciales y virtuales)	Horas trabajo autónomo	Horas totales
Prueba objetiva	A1 A3 A6 A7 B1 B2 B3 B4 B5 B6 B7 B9 B10 B12 B15 C3 C4 C6	5	90	95
Atención personalizada		5	0	5
(*)Los datos que aparecen en la tabla de planificación són de carácter orientativo, considerando la heterogeneidad de los alumnos				

Metodologías	
Metodologías	Descripción
Prueba objetiva	Examen escrito, dividido en: - Examen de teoría. - Examen de prácticas. Observaciones: En proceso de extinción no hay ningún tipo de evaluación continua. No se recogerán ejercicios prácticos, sólo habrá exámenes. Para más información consultar el Moodle o hablar con el profesor.

Atención personalizada	
Metodologías	Descripción
Prueba objetiva	Resolución de dudas.

Evaluación			
Metodologías	Competencias / Resultados	Descripción	**Calificación**

Prueba objetiva	A1 A3 A6 A7 B1 B2 B3 B4 B5 B6 B7 B9 B10 B12 B15 C3 C4 C6	Examen de teoría: - Calificación mínima exigida: 4 de 10. - 60% de la calificación global. Examen de prácticas: - Prueba escrita de la parte de práctica. - Calificación mínima exigida: 4 de 10. - 40% de la calificación global.	100
Otros			

Observaciones evaluación
Calificación en actas: - Aprobado o superior: Calificación final mayor o igual que 5 y cumplir todos los requisitos exigidos para superar la materia. - Suspenso: Calificación final menor que 5 o calificación final mayor o igual que 5 y no cumplir todos los requisitos exigidos para superar la materia. En el segundo caso, la calificación en actas será 4,5. - No Presentado: Equivale a no presentado a ambos exámenes finales (teoría y prácticas). Para más información consultar el Moodle o hablar con el profesor.

Fuentes de información	
Básica	
Complementária	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. 3Edición. 2002. Signatura FIC: D15 BUD. Knoernschild, K.. Java desing: objects, UML and process. Addison-Wesley, Boston, MA. 2002. Signatura FIC: D32 (Jav) KNO.Recursos webEl principal recurso web de la asignatura es la página de la misma existente en la facultad virtual de la UDC. En dicha página se encontrará toda la información necesaria para el desarrollo de la asignatura (transparencias, boletines de ejercicios, prácticas, tests, preguntas frecuentes, links en Internet, etc.). El acceso a dicha página está limitado a los alumnos matriculados en la asignatura. Para información general sobre Java se recomienda acudir a la página web oficial del lenguaje (http://java.sun.com) en donde podremos acceder a noticias, documentación, descarga del herramientas, etc. Para estar al día sobre las novedades del lenguaje se recomienda acceder a páginas de noticias como por ejemplo JavaHispano (http://javahispano.org).Otros materiales de apoyoLa biblioteca de la Facultad dispone de varias revistas científicas que están a disposición del alumno, y que revisten interés para la profundización en algún aspecto particular del temario, o para disponer de actualizaciones (estados del arte) de alguna de las técnicas y metodologías expuestas en clase. Algunas de las más utilizadas en la asignatura de POO son:Communications of the ACM Journal of Object Oriented Programming (ya no se publica pero hay números antiguos) IEEE Transactions on Knowledge and Data Engineering Data and Knowledge Enginnering ACM transactions on programming languages and systems Lecture Notes in Computer Science (aquellos volúmenes que hacen referencia a congresos sobre objetos como ECOOP ? European Conference on Object Oriented Programming).

Recomendaciones
Asignaturas que se recomienda haber cursado previamente
Estructura de Datos y de la Información/614111102 Programación/614111109
Asignaturas que se recomienda cursar simultáneamente
Metodología de la Programación/614111205
Asignaturas que continúan el temario

Análisis de Sistemas Informáticos/614111402
Diseño de Sistemas de Información/614111403
Integración de Sistemas/614111503
Proyecto Fin de Carrera/614111505
Análisis y Diseño Orientado a Objectos/614111602
Otros comentarios

(*) La Guía Docente es el documento donde se visualiza la propuesta académica de la UDC. Este documento es público y no se puede modificar, salvo cosas excepcionales bajo la revisión del órgano competente de acuerdo a la normativa vigente que establece el proceso de elaboración de guías