ASIGNATURA DE GRADO:
DISEÑO DE APLICACIONES ORIENTADAS A OBJETOS

Curso 2017/2018

(Código de asignatura : 71022011)

NOMBRE DE LA ASIGNATURA	DISEÑO DE APLICACIONES ORIENTADAS A OBJETOS
CÓDIGO	71022011
CURSO ACADÉMICO	2017/2018
DEPARTAMENTO	LENGUAJES Y SISTEMAS INFORMÁTICOS
TÍTULO EN QUE SE IMPARTE	**GRADO EN INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN**
CURSO - PERIODO - TIPO	GRADO EN INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN - SEGUNDO CURSO - SEMESTRE 2 - OBLIGATORIAS ESPECÍFICO PARA INGENIEROS TÉCNICOS EN INFORMÁTICA DE GESTIÓN EN UNED - OPTATIVAS CURSO - SEMESTRE 2 - OBLIGATORIAS ESPECÍFICO PARA INGENIEROS TÉCNICOS EN INFORMÁTICA DE GESTIÓN - OPTATIVAS CURSO - SEMESTRE 2 - OBLIGATORIAS
Nº ECTS	6
HORAS	150.0
IDIOMAS EN QUE SE IMPARTE	CASTELLANO

PRESENTACIÓN Y CONTEXTUALIZACIÓN

El objetivo de esta guía es orientar al alumno en el estudio de la asignatura. Se recomienda la lectura completa de la guía a comienzo del cuatrimestre para tener una idea completa de la temática de la asignatura y el calendario de prácticas, de forma que el alumno pueda planificar su trabajo para cumplir con las fechas de entrega.

La asignatura de Diseño de Aplicaciones Orientada a Objetos (DAOO en adelante) se imparte en el segundo cuatrimestre del segundo curso, consta de 6 créditos, con carácter de formación avanzada, para la titulación de grado en Informática.

Esta asignatura es una introducción al Diseño de Aplicaciones Orientadas a Objetos (DAOO) y tiene como objetivo principal el estudio y la puesta en práctica de los patrones de diseño como abstracción para la realización de aplicaciones y sistemas complejos basados en la metodología de orientación a objetos. Dado que el alumno ha tomado contacto con esta metodología en la asignatura de Programación Orientada a Objetos (POO), en este curso se profundizará en su uso y aplicación en escenarios más complejos. Así mismo, y dado que en POO el alumno ha desarrollado programas básicos utilizando el lenguaje de programación Java y la herramienta didáctica BlueJ, la asignatura DAOO mantendrá el uso de este lenguaje de programación motivando al alumno a implementar las actividades propuestas en el temario utilizando Eclipse como entorno de desarrollo estándar y mucho más extendido en la comunidad.

REQUISITOS Y/O RECOMENDACIONES PARA CURSAR LA ASIGNATURA

Al tratarse de una continuación de asignatura POO, se requiere que el alumno sabe lo básico del diseño y desarrollo usando la orientación a objetos y como hacer programas con Java.

EQUIPO DOCENTE

Nombre y Apellidos	TIMOTHY MARTIN READ **(Coordinador de Asignatura)**
Correo Electrónico	tread@lsi.uned.es
Teléfono	91398-8261
Facultad	ESCUELA TÉCN.SUP INGENIERÍA INFORMÁTICA
DEPARTAMENTO	LENGUAJES Y SISTEMAS INFORMÁTICOS

Nombre y Apellidos	JUAN MANUEL CIGARRAN RECUERO
Correo Electrónico	juanci@lsi.uned.es
Teléfono	91398-9828
Facultad	ESCUELA TÉCN.SUP INGENIERÍA INFORMÁTICA
DEPARTAMENTO	LENGUAJES Y SISTEMAS INFORMÁTICOS

Nombre y Apellidos	COVADONGA RODRIGO SAN JUAN
Correo Electrónico	covadonga@lsi.uned.es
Teléfono	91398-6487
Facultad	ESCUELA TÉCN.SUP INGENIERÍA INFORMÁTICA
DEPARTAMENTO	LENGUAJES Y SISTEMAS INFORMÁTICOS

HORARIO DE ATENCIÓN AL ESTUDIANTE

La enseñanza a distancia posee unas características que la diferencian de la presencial. Sin embargo, esto no impide que el alumno pueda disponer de la ayuda y los recursos necesarios para cursar las asignaturas en las que se matricule. Los mecanismos de los que dispone el alumno para facilitar el aprendizaje requerido en la asignatura son los siguientes:

- Tutores en los centros asociados. Los tutores serán los encargados de desarrollar las sesiones presenciales de seguimiento y control de las prácticas. Para la realización de las prácticas el alumno debe ponerse en contacto con el tutor correspondiente, para conocer cuanto antes el horario y las sesiones previstas.
- Tutorías presenciales o virtuales en el centro asociado correspondiente.
- Entorno Virtual. A través de CiberUNED el equipo docente de la asignatura pondrá a disposición de los alumnos diverso material de apoyo en el estudio, así como el enunciado de la práctica obligatoria. Dispone además de foros donde los alumnos podrán plantear sus dudas para que sean respondidas por los tutores o por el propio equipo docente. Es el soporte fundamental de la asignatura, y supone la principal herramienta de comunicación entre el equipo docente, los tutores y los alumnos, así como de los alumnos entre sí.

Horario de asistencia al estudiante:

- Timothy Read: Miércoles 09:30 - 13:30
- Juan Cigarrán: Jueves 16:00 - 20:00

COMPETENCIAS QUE ADQUIERE EL ESTUDIANTE

Con el estudio de la asignatura el alumno debe adquirir una visión general de la DAOO. En concreto, se pretende que al finalizar el curso el alumno: a) Haya reforzado y ampliado sus conocimientos de programación orientada a objetos, b) Conozca los patrones de uso general de creación, estructura y comportamiento, c) Sea capaz de identificarlos en la etapa de diseño de una aplicación software, y; d) Conozca el modo en el que estos se implementan en el lenguaje Java.

RESULTADOS DE APRENDIZAJE

Con el estudio de la asignatura el alumno debe adquirir una visión general de la DAOO. En concreto, se pretende que al finalizar el curso el alumno: a) Haya reforzado y ampliado sus conocimientos de programación orientada a objetos, b) Conozca los patrones de uso general de creación, estructura y comportamiento, c) Sea capaz de identificarlos en la etapa de diseño de una aplicación software, y; d) Conozca el modo en el que estos se implementan en el lenguaje Java.

Los resultados de aprendizaje que los alumnos van a conseguir en esta asignatura, según la memoria de verificación de ANECA son:

- R4. Realizar pruebas de validación y depuración de un programa dado.
- R5.Identificar clases de objetos con los datos de un problema.
- R6. Crear clases y objetos y manipularlos.
- R7. Identificar los métodos que caracterizan a una clase de objetos.
- R8. Utilizar el mecanismo de la herencia para crear clases.
- R9. Identificar las relaciones entre clases en distintos casos de uso.
- R10. Diseñar y documentar interfaces de usuario.
- R11. Usar Patrones de Diseño
- R12. Abordar metodológicamente proyectos de programación medios.
- R13. Dominar un lenguaje de programación acorde al paradigma del contenido de la materia.

CONTENIDOS

Parte 1. El diseño orientado a objetos

Parte 2. Los patrones de diseño

METODOLOGÍA

La modalidad y tipo de actividades que se contemplan incluye: trabajo con contenidos teórico prácticos utilizando la bibliografía y el material complementario. Trabajo autónomo con las actividades de ejercicios y pruebas de autoevaluación disponibles, y realización de una práctica bajo la supervisión del tutor, con las herramientas y directrices preparadas por el equipo docente. La interacción con el equipo docente y los tutores se describe más adelante.

Para solicitar plaza/turno de prácticas de laboratorio/experimentales, el estudiante tendrá que acceder a la aplicación de prácticas desde su escritorio. En estas imágenes puede ver desde dónde se puede realizar el acceso a dicha aplicación:

https://descargas.uned.es/publico/pdf/guias/ACCESO_PRACTICAS_GRADOS_2017.pdf

Si al acceder a ella no encuentra ninguna oferta, deberá ponerse en contacto con el centro asociado donde está matriculado.

SISTEMA DE EVALUACIÓN

TIPO DE PRUEBA PRESENCIAL

Tipo de examen	Examen mixto
Preguntas test	10
Preguntas desarrollo	2
Duración del examen	120 (minutos)
Material permitido en el examen	No está permitido el uso de material adicional.
Criterios de evaluación	Para que el examen de un alumno sea calificado deberá haber asistido, como mínimo, a una sesión presencial de prácticas en su centro asociado y haber entregado y aprobado la práctica obligatoria. Para que un alumno pueda aprobar la asignatura deberá haber superado un mínimo de preguntas establecido en la parte teórica (tipo test) del examen.
% del examen sobre la nota final	90
Nota mínima del examen para aprobar sin PEC	
Nota máxima que aporta el examen a la calificación final sin PEC	
Nota mínima en el examen para sumar la PEC	
Comentarios y observaciones	

PRUEBAS DE EVALUACIÓN CONTINUA (PEC)

¿Hay PEC?	
Descripción	
Criterios de evaluación	
Ponderación de la PEC en la nota final	
Fecha aproximada de entrega	
Comentarios y observaciones	

OTRAS ACTIVIDADES EVALUABLES

¿Hay otra/s actividad/es evaluable/s?	
Descripción	**Práctica Obligatoria** El trabajo del curso incluye la realización de una práctica obligatoria de programación.
Criterios de evaluación	Las prácticas son corregidas por los Tutores de los Centros Asociados, quienes las reenvían después a la sede central. La nota asignada por el tutor podrá incrementar hasta un máximo de 1 punto en la nota final de la asignatura, por supuesto, siempre que esté aprobada. Las notas de las prácticas no se guardan de un curso para otro.
Ponderación en la nota final	1
Fecha aproximada de entrega	31/05/2018
Comentarios y observaciones	

¿Cómo se obtiene la nota final?

La ponderación final tanto de la práctica como de cada una de las partes del examen en la nota final de la asignatura quedaría de la siguiente manera:

- Práctica Obligatoria: 1 punto sobre 10
- Parte Teórica del Examen: 2,5 puntos sobre 10
- Parte Práctica del Examen: 6,5 puntos sobre 10

BIBLIOGRAFÍA BÁSICA

ISBN(13): 9788478290598
Título: PATRONES DE DISEÑO
Autor/es: Gamma, Erich ; Otros ;
Editorial: PEARSON ADDISON WESLEY

Este libro no trata de una introducción a la tecnología orientada a objetos ni al diseño orientado a objetos. Es un libro de patrones de diseño que describe soluciones simples y elegantes para problemas específicos del diseño de software orientado a objetos.

En este libro, Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides introducen los principios de los patrones de diseño y ofrecen un catálogo de dichos patrones. Así, este libro realiza dos importantes contribuciones. En primer lugar, muestra el papel que los patrones pueden desempeñar diseñando la arquitectura de sistemas complejos. En segundo lugar, proporciona una referencia práctica de un conjunto de excelentes patrones que el desarrollador puede aplicar para construir sus propias aplicaciones.

BIBLIOGRAFÍA COMPLEMENTARIA

Como material complementario, el equipo docente recomienda la consulta de los siguientes documentos:

- Patterns in Java: A Catalog of Reusable Design Patterns Illustrated with UML, 2nd Edition, Volume 1. Mark Grand. Wiley. 2002
- Pattern Hatching: Design Patterns Applied. John Vlissides. Addison-Wesley. 1998.
- Design Patterns Explained: A New Perspective on Object-Oriented Design. Alan Shalloway, James R. Trott. Addison-Wesley. 2001.
- Resumen de Luis Fernandez (http://software.guisho.com/patrones-de-diseno)

RECURSOS DE APOYO Y WEBGRAFÍA

Como materiales adicionales para el estudio de la asignatura se ofrece en el curso virtual:

- Esta guía de estudio y una versión extendida de la misma, la guía didáctica.
- Enunciados y soluciones de ejercicios teórico prácticos que el alumno puede usar como ejercicios de autoevaluación.
- Exámenes resueltos de anteriores convocatorias.
- Lista de preguntas frecuentes, que recogen dudas de años anteriores.

Además, a través de CiberUNED se pondrán a disposición de los alumnos herramientas necesarias para el desarrollo de la práctica. Los alumnos pueden acceder al curso virtual en este entorno con su identificador y clave de acceso. En los centros asociados los alumnos dispondrán de ordenadores en donde el entorno de desarrollo BlueJ deberá estar instalado. Además, los alumnos que dispongan de un ordenador personal podrán instalarse dicho entorno de desarrollo. En el entorno CiberUNED se encontrará este paquete software y las instrucciones para su instalación. El entorno virtual se usará como medio para que los estudiantes puedan acceder a material complementario de estudio. El equipo docente añadirá aquellos documentos o referencias que considere adecuados para que los alumnos puedan complementar los conocimientos adquiridos a través del estudio de la bibliografía básica.

TUTORIZACIÓN EN CENTROS ASOCIADOS

Revisión de calificaciones

Se podrá solicitar revisión de las calificaciones en el plazo y forma establecidos por la UNED.