

PROGRAMACIÓN ORIENTADA A OBJETOS

Curso 2016/2017

(Código: 71901072)

1. PRESENTACIÓN DE LA ASIGNATURA

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 Programación Orientada a Objetos (POO en adelante) se imparte en el segundo cuatrimestre del primer curso, consta de 6 créditos, con carácter de formación básica, para la titulación de grado en Informática.

2. CONTEXTUALIZACIÓN EN EL PLAN DE ESTUDIOS

Esta asignatura es una introducción a la POO y tiene como objetivo principal que el alumno adquiera los conocimientos y habilidades necesarios para poder desarrollar programas básicos utilizando la metodología de la orientación a objetos. Dado el hecho de que es una asignatura de primer curso, en lugar de emplear un entorno estándar de desarrollo de software como (por ejemplo) Eclipse, se usa BlueJ debido a su naturaleza didáctica.

Se trata de una continuación de la asignatura Fundamentos de Programación que se centra en los tipos abstractos de datos. En Fundamentos los alumnos ven un lenguaje híbrido C±, que mezcla partes de los lenguajes de programación C y C++, pero (en términos de la abstracción de datos) que no va más allá que la utilización de la abstracción STRUCT. Así que esta asignatura, que presenta mecanismos de la POO como la herencia y el polimorfismo, representa una extensión natural de aquélla. El hecho de haber cursado Fundamentos ayudará a los estudiantes a entender muchos conceptos de la POO.

Las competencias adquiridas en la asignatura son:

(G.2) Competencias cognitivas superiores: selección y manejo adecuado de conocimientos, recursos y estrategias cognitivas de nivel superior apropiados para el afrontamiento y resolución de diversos tipos de tareas/problemas con distinto nivel de complejidad y novedad: Análisis y Síntesis. Aplicación de los conocimientos a la práctica Resolución de problemas en entornos nuevos o poco conocidos. Pensamiento creativo. Razonamiento crítico. Toma de decisiones.

(G.4) Competencias de expresión y comunicación (a través de distintos medios y con distinto tipo de interlocutores): Comunicación y expresión escrita. Comunicación y expresión oral. Comunicación y expresión en otras lenguas (con especial énfasis en el inglés). Comunicación y expresión matemática, científica y tecnológica (cuando sea requerido y estableciendo los niveles oportunos).

(FB.5) Conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, así como de los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería.

(BC.1) Capacidad para diseñar, desarrollar, seleccionar y evaluar, aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a los principios éticos y a la legislación y normativa vigente.

(BTEisw.1) Capacidad para desarrollar, mantener y evaluar servicios y sistemas software que satisfagan todos los requisitos del usuario y se comporten de forma fiable y eficiente, sean asequibles de desarrollar y mantener y cumplan normas de calidad, aplicando las teorías, principios, métodos y prácticas de la Ingeniería del Software.



3.REQUISITOS PREVIOS REQUERIDOS PARA CURSAR LA ASIGNATURA

Al tratarse de una asignatura de carácter introductorio, no se requiere ningún requisito previo más allá de los conocimientos que un alumno debe tener en este nivel de enseñanza y del manejo de un ordenador personal a nivel de un usuario doméstico.

4.RESULTADOS DE APRENDIZAJE

Con el estudio de la asignatura el alumno debe adquirir una visión general de la POO. La POO es una evolución de la programación procedural basada en funciones. La POO nos permite agrupar secciones de código con funcionalidades comunes. Una de las principales desventajas de la programación procedural basada en funciones es su construcción, cuando una aplicación bajo este tipo de programación crece, la modificación del código se hace muy trabajosa y difícil debido a que el cambio de una sola línea en una función, puede acarrear la modificación de muchas otras líneas de código pertenecientes a otras funciones que estén relacionadas.

Con la programación orientada a objetos se pretende agrupar el código encapsulándolo y haciéndolo independiente, de manera que una modificación debida al crecimiento de la aplicación solo afecte a unas pocas líneas. La organización de una aplicación en POO se realiza mediante estructuras de código, también llamados objetos. Estos objetos contienen una serie de procedimientos e información destinados a resolver un grupo de tareas con un denominador común. Un procedimiento que este situado en un objeto no podrá ser usado por otro procedimiento perteneciente a otro objeto, si no es bajo una serie de reglas. Los datos que mantenga el objeto, permanecerán aislados del exterior y sólo se podrá acceder a ellos siguiendo ciertas normas. El objetivo de POO es catalogar y diferenciar el código, en base a estructuras jerárquicas dependientes, al estilo de un árbol genealógico.

Los resultados de aprendizaje para las competencias adquiridas en la asignatura son:

- R4. Realizar pruebas de validación y depuración de un programa dado (competencias: G.2, FB.5, BC.1).
- R5. Identificar clases de objetos con los datos de un problema (competencias: G.2, G.4, BC.1, FB.5, BTEIsw.4)
- R6. Crear clases y objetos y manipularlos (competencias: G.2, FB.5, BC.1)
- R7. Identificar los métodos que caracterizan a una clase de objetos (competencias: G.2, FB.5, BC.1)
- R8. Utilizar el mecanismo de la herencia para crear clases (competencias: G.2, FB.5, BC.1)
- R9. Abordar metodológicamente proyectos de programación sencillos (competencias: G.2, FB.5, BC.1)
- R10. Dominar un lenguaje de programación acorde al paradigma del contenido de la materia (competencias: G.2, FB.5, BC.1)

5.CONTENIDOS DE LA ASIGNATURA

Los contenidos de este curso se organizan en dos partes:

Parte I. Fundamentos de programación orientada a objetos.

Capítulo 1. Objetos y clases

Capítulo 2. Definiciones de clases.

Capítulo 3. Interacción de objetos.

Capítulo 4. Agrupación de objetos.

Capítulo 5. Comportamiento más sofisticado.

Capítulo 6. Diseñar clases.

Capítulo 7. Objetos con buen comportamiento.



Parte 2. Estructuras de las aplicaciones.

Capítulo 8. Mejora de la estructura mediante herencia.

Capítulo 9. Más acerca de la herencia.

Capítulo 10. Técnicas de abstracción adicionales.

Capítulo 11. Construcción de interfaces gráficas de usuario.

Capítulo 12. Tratamiento de errores.

Capítulo 13. Diseño de aplicaciones.

6.EQUIPO DOCENTE

- [TIMOTHY MARTIN READ](#)
- [JUAN MARTINEZ ROMO](#)
- [JUAN MANUEL CIGARRAN RECUERO](#)
- [JOSE LUIS DELGADO LEAL](#)
- [ANDRES DUQUE FERNANDEZ](#)

7.METODOLOGÍA Y ACTIVIDADES DE APRENDIZAJE

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.

8.EVALUACIÓN

Se aplican las siguientes modalidades

Evaluación continua

* La autoevaluación supone un aspecto muy importante dentro de un proceso general de aprendizaje. En esta asignatura, planteamos a los alumnos un proceso de autoevaluación basado en la realización de preguntas tipo test y en la resolución de problemas de examen de convocatorias anteriores. Para ello, en el módulo de contenidos dentro del entorno virtual CiberUNED los alumnos pueden encontrar dos apartados de "Autoevaluación" correspondientes a cada una de las Unidades Didácticas del curso. En dichos apartados el alumno podrá encontrar tanto cuestiones y problemas resueltos como sus



soluciones, con las que podrá realizar una autoevaluación de sus conocimientos.

* Práctica obligatoria. Realización de una práctica obligatoria a lo largo del cuatrimestre, bajo la supervisión del tutor. La práctica aporta 10% de la nota final y requiere la presencialidad del alumno en el Centro Asociado para su realización (en una única sesión presencial) y corrección por parte del tutor. La nota mínima para aprobar la práctica es 5.0.

La evaluación final

Se llevará a cabo a partir de las siguientes pruebas:

* Calificación de la práctica obligatoria, es necesario aprobarla para la superación de la asignatura.

* Realización de un examen teórico/práctico que aporta 90% de la nota final. La nota mínima para aprobar el examen es 5.0.

El examen constará de dos partes, una teórica formada por preguntas tipo test y que será eliminatoria, y una segunda parte práctica formada por un problema de programación con varios apartados y en los que el alumno demostrará el nivel de los conocimientos adquiridos.

Se incluirán también preguntas sobre la práctica obligatoria.

Criterios de Calificación

Para que el examen de un alumno sea calificado deberá haber asistido, como mínimo, a dos sesiones presenciales de prácticas en su centro asociado y haber entregado y aprobado la práctica obligatoria.

Criterios de Evaluación

Para que un alumno pueda aprobar la asignatura deberá haber superado un mínimo de preguntas establecido en la parte teórica (tipo test) del examen. En la evaluación de la asignatura se tendrán en cuenta especialmente los aspectos relativos al diseño más que a los detalles propios de la implementación.

9. BIBLIOGRAFÍA BÁSICA

ISBN(13): 9788490355312

Título: PROGRAMACION ORIENTADA A OBJETOS CON JAVE USANDO BLUEJ (6 edicion)

Autor/es: David J. Barnes, Michael Kölling ;

Editorial: PEARSON EDUCACION

Buscarlo en Editorial UNED



Buscarlo en librería virtual UNED

Buscarlo en bibliotecas UNED

Buscarlo en la Biblioteca de Educación

Comentarios y anexos:

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

- * Apuntes de introducción a la informática accesibles vía web desde la página web de la asignatura "Programación en Java" de la Universidad Carlos III de Madrid:
- * <http://www.it.uc3m.es/java/infoAdicional/apuntes/RDA.pdf>
- * Resumen de Java del tutor de Málaga (<http://www.lsi.uned.es/lp/CursoJava.pdf>)

10. BIBLIOGRAFÍA COMPLEMENTARIA

11. RECURSOS DE APOYO

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.

12. TUTORIZACIÓN

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.



- profesor D. José Luis Delgado Leal, horario de asistencia al estudiante:

Lunes, de 08:30 a 09:30 horas. Jueves, de 08:30 a 11:30 horas. Despacho 2.13 de la ETSI Informática tño: 91 398 87 36

Ámbito: GUI - La autenticidad, validez e integridad de este documento puede ser verificada mediante el "Código Seguro de Verificación (CSV)" en la dirección <https://sede.uned.es/valida/>



A6445483F913E5F5F650BF13AF757C9B