Universidad Zaragoza

Curso : 2020/21

27024 - Informática II

Información del Plan Docente

Año académico: 2020/21
Asignatura: 27024 - Informática II
Centro académico: 100 - Facultad de Ciencias
Titulación: 453 - Graduado en Matemáticas
Créditos: 6.0
Curso: 4
Periodo de impartición: Primer semestre
Clase de asignatura: Optativa
Materia: ---

1.Información Básica

1.1.Objetivos de la asignatura

La asignatura y sus resultados previstos responden a los siguientes planteamientos y objetivos:

Esta asignatura de carácter optativo pretende avanzar en la formación en programación adquirida mediante la asignatura de formación básica *Informática I*, presentando técnicas de programación avanzada de aplicaciones. Concretamente, está concebida como una introducción a la programación orientada a objetos (POO), utilizando el lenguaje de programación Java y algunos elementos del lenguaje de diseño UML. Además de los fundamentos de la POO (clases, herencia y polimorfismo), el curso realiza también una breve introducción a la programación genérica y a la guiada por eventos.

1.2.Contexto y sentido de la asignatura en la titulación

Se trata de una asignatura optativa de la titulación, incluida en la materia *Informática* del módulo *Fundamentos de Informática.* Respecto a los itinerarios sugeridos en el Grado, se encuentra en el bloque central del itinerario en *Informática y Cálculo Científico* y en el bloque fronterizo de los siguientes cuatro itinerarios: *Álgebra, Geometría y Topología; Astrodinámica; Estadística; y Matemática Aplicada*.

1.3.Recomendaciones para cursar la asignatura

Para cursar esta asignatura es recomendable haber superado la de Informática I. Para alcanzar los objetivos de aprendizaje es imprescindible asistir a todas las sesiones prácticas y resolver los problemas que en ellas se plantean. El aprendizaje de la programación de computadores es una tarea acumulativa, por lo que no es posible progresar adecuadamente sin haber afianzado los conceptos previos. Esto hace imprescindible el trabajo continuado desde el comienzo del curso, y el uso adecuado de las tutorías académicas.

2.Competencias y resultados de aprendizaje

2.1.Competencias

Al superar la asignatura, el estudiante será más competente para...

Desenvolverse en el manejo de los objetivos descritos en el apartado ?Resultados de Aprendizaje?. Entre las competencias de la titulación que se desarrollan especialmente en esta asignatura se encuentran:

Tener la capacidad de reunir e interpretar datos relevantes, particularmente en el área de las Matemáticas, para emitir juicios, usando la capacidad de análisis y abstracción, que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.

Aprender nuevos conocimientos y técnicas de forma autónoma.

Proponer, analizar, validar e interpretar modelos de situaciones reales sencillas, utilizando las herramientas más adecuadas a los fines que se persigan.

Desarrollar algoritmos y programas que resuelvan problemas matemáticos, utilizando para cada caso el entorno computacional adecuado.

2.2.Resultados de aprendizaje

El estudiante, para superar esta asignatura, deberá demostrar los siguientes resultados...

Es capaz de abordar la solución de problemas aplicando el paradigma de la programación orientada a objetos.

Comprende y utiliza la herencia y el polimorfismo en el diseño de aplicaciones.

Comprende y utiliza los diagramas de clases en el desarrollo de aplicaciones.

Es capaz de desarrollar interfaces gráficas de usuario guiadas por eventos.

Conoce los fundamentos de la programación genérica y puede aplicarlos para el uso y, eventualmente, la construcción de tipos de datos.

2.3.Importancia de los resultados de aprendizaje

Proporcionan una formación de carácter optativo dentro del Grado. Además de afianzar las técnicas de programación estructurada obtenidas en varias asignaturas, en particular la de Programación I, el paradigma orientado a objeto proporciona un grado de abstracción que facilita la resolución de problemas de mayor alcance.

3.Evaluación

3.1.Tipo de pruebas y su valor sobre la nota final y criterios de evaluación para cada prueba

El estudiante deberá demostrar que ha alcanzado los resultados de aprendizaje previstos mediante las siguientes actividades de evaluacion

1) Resolución de cuatro problemas de programación y diseño propuestos periódicamente, que serán similares a los realizados en las clases prácticas. Los problemas no entregados en el plazo previsto se calificarán con la nota 0. Esta actividad supone el 20% de la calificación final.

2) Diseño y programación de una aplicación. Esta actividad supone el 20% de la calificación final.

3) Examen final de la asignatura. Para superar esta prueba será necesario obtener una nota mínima de 3 sobre 10. La nota obtenida en este examen supone el 60% de la calificación final.

No obstante, coforme a la normativa de evaluación de la Universidad de Zaragoza, el estudiante podrá superar la asignatura mediante una prueba global única.

4.Metodología, actividades de aprendizaje, programa y recursos

4.1.Presentación metodológica general

El proceso de aprendizaje que se ha diseñado para esta asignatura se basa en lo siguiente:

Presentación de conceptos teóricos en clases magistrales.

Resolución de problemas, tanto individual como colaborativamente, en las prácticas con ordenador.

4.2.Actividades de aprendizaje

Presentación de los conceptos teóricos y tecnológicos en clases magistrales (2 horas/semana)

Resolución e implementación de problemas en sesiones tutorizadas de prácticas con ordenador (2 horas/semana)

Trabajo personal, en particular el relacionado con las pruebas (1) y (2) reseñadas en el apartado de Evaluación.

Las actividades docentes y de evaluación se llevarán a cabo de modo presencial salvo que, debido a la situación sanitaria, las disposiciones emitidas por las autoridades competentes y por la Universidad de Zaragoza dispongan realizarlas de forma telemática.

4.3.Programa

1. Introducción. El paradigma de programación orientado a objetos. La máquina virtual de Java: compilación y ejecución de programas. Desarrollo de programas en un IDE.

2. Revisión de elementos de la programación estructurada en el lenguaje Java. Tipos primitivos; variables y constantes; operadores y expresiones; funciones matemáticas usuales: la clase Math. Sentencias elementales: asignación, lectura básica por teclado (la clase Scanner), escritura en pantalla. Composición secuencial, condicional e iterativa. Definición e invocación de métodos de clase. Sobrecarga de métodos. Recursión.

3. Introducción a la programación orientada a objetos. Objetos, clases y variables referencia: la referencia null. El ciclo de vida de un objeto: creación, el operador new y los métodos constructores; uso, acceso a miembros y paso de mensajes; la destrucción de un objeto. Los objetos arrays de Java.

4. Definición de una clase. Miembros de instancia y de clase. Definición de los métodos constructores. Niveles de acceso a miembros: la interfaz pública de una clase. Espacios de nombres: paquetes de clases.

5. Introducción al diseño orientado a objetos. Diagramas de clases de UML. Asociaciones y clases asociativas. Roles y navegación.

6. Herencia: concepto y tipos; la redefinición de métodos. Jerarquía de clases: la clase Object de Java. Polimorfismo: métodos virtuales. Generalización y especialización: clases y métodos abstractos.

7. El sistema de tipos de Java: interfaces. Programación con genéricos en Java. Aplicación a la implementación de tipos de

datos: las colecciones de Java.

8. Gestión de errores en tiempo de ejecución. Excepciones: generación, tratamiento y notificación.

9. Persistencia: entrada/salida de texto y binaria. Persistencia de objetos: la interface Serializable. Acceso a recursos remotos: las clases File y URL.

10. Programación dirigida por eventos: el mecanismo de notificación-suscripción. Programación de interfaces gráficas de usuario: contenedores, menús y controles básicos.

4.4.Planificación de las actividades de aprendizaje y calendario de fechas clave

Calendario de sesiones presenciales y presentación de trabajos

La información relativa al periodo de clases y fechas de exámenes está disponible en la web de la Facultad de Ciencias (https://ciencias.unizar.es/web/horarios.do).

Las fechas de entrega aproximadas de las "Actividades de evaluación" son:

- Problemas de programación y diseño: semanas 5, 7, 9 y 11
- Diseño y programación de una aplicación: antes del examen

No obstante, en la plataforma Moodle se publicarán con antelación suficiente las fechas concretas de entrega de estas pruebas.

4.5.Bibliografía y recursos recomendados

http://biblos.unizar.es/br/br_citas.php?codigo=27024&year=2020