



UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

GUÍA DOCENTE

CURSO: 2012/13

40813 - PROGRAMACIÓN I

CENTRO: 180 - Escuela de Ingeniería Informática

TITULACIÓN: 4008 - Grado en Ingeniería Informática

ASIGNATURA: 40813 - PROGRAMACIÓN I

Vinculado a : (Titulación - Asignatura - Especialidad)

4801-Doble Grado en Ingeniería Informática y - 48113-PROGRAMACIÓN I - 00

CÓDIGO UNESCO: 1203 **TIPO:** Obligatoria **CURSO:** 2 **SEMESTRE:** 1º semestre

CRÉDITOS ECTS: 6 **Especificar créditos de cada lengua:** **ESPAÑOL:** 6 **INGLÉS:**

SUMMARY

REQUISITOS PREVIOS

- Introducción a la Informática
- Fundamentos de Programación.

Datos identificativos del profesorado que la imparte.

Plan de Enseñanza (Plan de trabajo del profesorado)

Contribución de la asignatura al perfil profesional:

Programación I es una asignatura que aporta al estudiante una formación básica en el modelo de programación orientado a objetos y en el de programación guiada por eventos, mediante un lenguaje que ilustra de forma sencilla los principales elementos de ambos paradigmas y que facilita el adiestramiento en el desarrollo de interfaces gráficas de usuario. Además, se introducen los conceptos de tipos abstractos de datos y contenedor cuya concreción en las colecciones ofertadas por el lenguaje proporciona elementos muy útiles para la resolución de gran número de problemas. Asimismo, se inicia al estudiante en las técnicas de prueba de programas utilizando paquetes de apoyo específicos.

Competencias que tiene asignadas:

Competencias generales: G1, G2, G3, G5

Competencias nucleares de la Universidad de Las Palmas de Gran Canaria: N4

Competencias del título de Grado en Ingeniería Informática: T3, T7, T8, T9

Competencias comunes a la Ingeniería Informática: CII05, CII08

Objetivos:

- 1) Aplicar el paradigma de programación orientada a objetos en el desarrollo de programas y usar herramientas específicas de apoyo.
- 2) Probar programas utilizando JUnit
- 3) Utilizar las colecciones de Java de forma apropiada en el desarrollo de programas.
- 4) Aplicar el modelo de programación guiada por eventos de forma adecuada en el desarrollo de programas.

Contenidos:

1. Tipos Abstractos de Datos
 - 1.1. Abstracción de datos
 - 1.2. Tipos genéricos
2. Programación Orientada a Objetos en Java
 - 2.1. Introducción a la Programación Orientada a Objetos
 - 2.2. Clases, objetos y paquetes
 - 2.3. Entrada y salida a la consola. Clases útiles
 - 2.4. Expresiones, estructuras de control de flujo y arrays
 - 2.5. Control de excepciones
 - 2.6. Herencia, interfaces y polimorfismo
3. Prueba de Programas
 - 3.1. Técnicas de prueba de programas y diseño de casos de prueba
 - 3.2. Pruebas unitarias con JUnit
4. Contenedores
 - 4.1. Introducción
 - 4.2. Colecciones de Java
5. Fundamentos de Programación Guiada por Eventos
 - 5.1. Introducción
 - 5.2. Interfaz gráfica de usuarios (GUI) en Java

Metodología:

Todas las tareas a realizar en las sesiones académicas se van a desarrollar en equipos. En el horario establecido por el centro para las sesiones académicas teóricas y prácticas, se utilizan dos modalidades para los contextos de las actividades que los profesores y los alumnos desarrollarán: a nivel de grupo y a nivel de equipo.

Los quehaceres diarios durante las sesiones académicas teóricas y de problemas se resumen a continuación:

1. Cada profesor presenta a su grupo una síntesis del contenido de cada tema, previa publicación del material disponible en la página web de la asignatura, indica las tareas individuales y colectivas a realizar y establece un plazo de entrega.
2. Este lapso de tiempo se consume en las sesiones de trabajo colaborativo, cada equipo planifica y organiza su labor, que incluye la ejecución de las tareas colectivas con discusión de aportaciones individuales y resolución de dudas entre iguales.

Los quehaceres diarios durante las sesiones académicas prácticas se resumen a continuación:

Cada profesor presenta a su grupo la actividad dirigida a realizar por cada equipo, establece un plazo de entrega y el porcentaje que aporta a la nota del trabajo práctico. Este lapso de tiempo se consume en las sesiones de trabajo colaborativo de forma análoga a las sesiones teóricas y de

problemas.

El tiempo que dedica el profesor a cada equipo durante las diferentes sesiones académicas será aproximadamente el mismo, intentando optimizar su presencia en cada uno para poder ocuparse de todos los equipos.

El medio principal para suministrar información referente a la asignatura es la página web de la asignatura.

Criterios y fuentes para la evaluación:

1) Exámenes individuales

Criterios relativos a las cuestiones:

-Conocimiento y comprensión de los contenidos

-Rigor y fluidez en la comunicación escrita

Criterios relativos al desarrollo de un programa:

-El profesor someterá el programa a las pruebas oportunas para formular una valoración que tenga en cuenta todos los aspectos implicados en su realización, con especial énfasis en la ejecución (funcionamiento, adecuación a las especificaciones, robustez, ...) y en el estilo (formato, comentarios, elección de identificadores, ...).

2) Trabajo práctico en equipo

Criterios relativos a una práctica:

-El profesor someterá los programas e información entregados a los análisis y pruebas oportunos para formular una valoración global que tenga en cuenta todos los aspectos implicados en su realización, con especial énfasis en la ejecución (funcionamiento, adecuación a las especificaciones, robustez, ...), en el estilo (formato, comentarios, elección de identificadores, ...) y la generación de una nueva propuesta colectiva o la selección entre las alternativas individuales. Cuando lo estime conveniente, el profesor podrá citar a los alumnos para formularles cuestiones que considere relevantes para la valoración global.

3) Coevaluación

Cada miembro del equipo valora a sus compañeros teniendo en cuenta su grado de:

-Colaboración

-Respeto

-Integración

-Distribución del trabajo

-Motivación

-Compromiso

-Conciliación

4) Examen escrito final

Los criterios para la evaluación de este examen son los mismos que los establecidos para la evaluación de los exámenes individuales.

Sistemas de evaluación:

1) Evaluación continua:

* Exámenes individuales

La escala que se utilizará para valorar esta parte es de 0 a 10 puntos y se efectuará a partir de exámenes individuales frente al ordenador que se irán realizando regularmente, este sistema se basa en cuestiones de respuesta corta y/o desarrollo de programas relativos a los temas tratados. Cada examen aporta el mismo porcentaje a la nota de esta parte y se valora de 0 a 10 puntos. La preparación de cada examen se realizará por medio de las actividades formativas en equipo, se requerirá la entrega de al menos el 90% del número de actividades planteado.

* Trabajo práctico en equipo

La escala que se utilizará para valorar el trabajo práctico es de 0 a 10 puntos y su evaluación se efectuará a partir de las prácticas de laboratorio que se irán planteando regularmente. El enunciado de cada práctica incluirá las especificaciones, plazos de entrega y porcentaje que aporta a la evaluación del trabajo práctico. Cada equipo deberá entregar, dentro de los plazos establecidos, los ficheros conteniendo el código fuente y demás información requerida, usando el medio que se establezca. La nota obtenida en cada práctica es la que le corresponderá a cada alumno componente del equipo que la realiza.

* Coevaluación

Cada alumno interviene en la evaluación al calificar la participación activa en el trabajo colaborativo de sus compañeros de equipo por medio de un cuestionario que se valorará de 0 a 10 puntos. La nota de un alumno por cada coevaluación realizada será la media de la otorgada por sus compañeros de equipo. La nota final de esta parte será la media de los resultados obtenidos en las coevaluaciones efectuadas a lo largo del semestre y se valorará de 0 a 10 puntos. Requiere la asistencia a las sesiones académicas, con un mínimo del 85%. Los resultados de la coevaluación serán supervisados por el profesor para evitar posibles disfunciones en la nota.

2) Examen escrito final

Aquellos alumnos que no superen la evaluación continua podrán optar a un examen escrito donde deberán demostrar sus conocimientos de la asignatura. El examen estará dividido en cuestiones de respuesta corta y/o desarrollo de programas, cada uno de los cuales aportará un porcentaje especificado a la nota. Se valorará de 0 a 10 puntos.

Criterios de calificación:

Para superar la asignatura habrá que obtener una nota mínima de 5 puntos en la evaluación continua de exámenes individuales. La nota final se calcula con la siguiente fórmula:

$$NF = NEI * 0.5 + NTP * 0.3 + NC * 0.2$$

Donde NF representa la nota final, NEI la nota de exámenes individuales, NTP la nota del trabajo práctico en equipo y NC la nota de la coevaluación. NEI será sustituida en la fórmula por la nota obtenida mediante el examen escrito final para aquellos alumnos que no alcancen la nota mínima establecida para la evaluación continua de exámenes individuales. La nota final de los alumnos que no cumplan las condiciones establecidas para superar la asignatura será como máximo de 4 puntos.

Plan de Aprendizaje (Plan de trabajo de cada estudiante)

Tareas y actividades que realizará según distintos contextos profesionales (científico, profesional, institucional, social)

Estudio de la documentación específica y técnica de la materia

Lecturas recomendadas directamente relacionadas con la materia

Actividades formativas, que se irán planteando regularmente, consistentes en la realización de programas en equipo que requieren del desarrollo individual para una posterior discusión colectiva.

Trabajo práctico consistente en la realización de las prácticas de laboratorio en equipo que requieren del desarrollo individual para una posterior discusión colectiva.

Las prácticas de laboratorio se intercalarán con las actividades formativas.

Uso del entorno de desarrollo Eclipse

Desarrollo de pequeñas aplicaciones con interfaz gráfica en Java.
Prueba de programas empleando JUnit.

Temporalización semanal de tareas y actividades (distribución de tiempos en distintas actividades y en presencialidad - no presencialidad)

Los estudiantes dedicarán 2 horas semanales a asistir a sesiones académicas teóricas y otras dos a sesiones académicas prácticas, que incluirán la realización de ejercicios de aplicación de las técnicas introducidas en la teoría y la solución de problemas de programación (60 horas en 15 semanas). Los exámenes que se realizarán a lo largo del semestre consumen 7.5 horas presenciales en total.

En promedio, realizarán cada semana 5 horas de trabajo no presencial: 2 horas dedicadas a la consulta y asimilación de materiales documentales y 3 horas para la realización de ejercicios y trabajos; ello supone 75 horas en 15 semanas.

Previsión de dedicación temporal en la relación de prácticas y actividades formativas.

De la semana 1 a la 4 se realizarán actividades formativas prácticas no evaluables. De la semana 5 a la 15 se realizarán 5 prácticas que constarán de dos semanas (una inicial de actividad individual y otra de tarea en equipo). Las prácticas a realizar tendrán la siguiente temática:

- 1) Clases, métodos y constructores
- 2) Herencia y Polimorfismo.
- 3) Prueba de programas
- 4) Contenedores
- 5) Interfaz gráfica

Las horas restantes las distribuirá el estudiante, en función de sus necesidades particulares, entre atención tutorial individual, tiempo adicional de estudio y realización de ejercicios y trabajos.

Recursos que tendrá que utilizar adecuadamente en cada uno de los contextos profesionales.

Bibliografía, documentación específica sobre el contenido de la asignatura y documentación técnica.

Herramientas de desarrollo general para Java.

Herramientas de edición de interfaz gráfica en Java.

Herramientas de prueba de programas en Java.

Herramientas de diseño con UML

Herramientas del campus virtual de la ULPGC.

Resultados de aprendizaje que tendrá que alcanzar al finalizar las distintas tareas.

- Explicar el concepto, implementar y usar de forma adecuada en el desarrollo de programas los tipos abstractos de datos.
- Describir los elementos básicos de la programación orientada a objetos y aplicarlos de forma adecuada en el desarrollo de programas.
- Explicar los conceptos de contenedor e iterador y usarlos de forma adecuada en el desarrollo de programas.
- Emplear técnicas de prueba para detectar errores en programas de complejidad media.
- Nombrar y usar librerías estándar de clases.
- Explicar los fundamentos de la programación guiada por eventos y usarla en el desarrollo de programas simples.

Atención presencial individualizada (incluir las acciones dirigidas a estudiantes en 5ª, 6ª y 7ª convocatoria)

Se usará para aclarar dudas y asesorar al alumno en las tareas individuales.

Atención presencial a grupos de trabajo

Se usarán para aclarar dudas, orientar las actividades colectivas dirigidas, mejorar el rendimiento académico, hacer un seguimiento del trabajo.

Atención telefónica

Los profesores podrán atender consultas telefónicas del alumnado en su horario de tutorías.

Atención virtual (on-line)

La atención virtual on-line se realizará haciendo uso de las herramientas del campus virtual de la ULPGC.

Bibliografía

[1 Básico] Programación orientada a objetos con Java /

Francisco Durán, Francisco Gutiérrez, Ernesto Pimentel.
Thomson Paraninfo,, Madrid : (2007)
9788497325721

[2 Básico] UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado /

Larman Craig.
Prentice Hall,, Madrid : (2003) - (2ª ed.)
8420534382

[3 Básico] Java tools for Extreme Programming: mastering open source tools including Ant, JUnit, and Cactus /

Richard Hightower, Nicholas Lesiecki.
John Wiley & Sons,, New York : (2002)
0-471-20708-X

[4 Recomendado] Programador Java certificado: curso práctico /

Antonio J. Martín
Sierra.
Ra-Ma,, Paracuellos de Jarama, Madrid : (2010) - (3ª ed.)
978-84-7897-972-1

[5 Recomendado] Test-driven development: a practical guide /

Dave Astels.
Prentice Hall PTR,, Upper Saddle River, N.J : (2003)
978-0-13-101649-1

[6 Recomendado] Ingeniería del software: un enfoque práctico /

Roger S. Pressman.
McGraw Hill,, México [etc.] : (2010) - (7ª ed.)
9786071503145