Programación Avanzada

13 de febrero de 2001

1 Información General

- Curso: 2000/2001
- Código de la Asignatura: 90F
- Cuatrimestre: Segundo
- Créditos: 4.5
- Créditos Teóricos: 3
- Créditos Prácticos: 1,5
- Categoría: Optativa
- Departamento: Estadística, Investigación Operativa y Computación
- Profesor: Coromoto León Hernández
- URL: http://nereida.deioc.ull.es/~cleon/pa
- e-mail: coro@csi.ull.es

2 Método de Evaluación

- La nota de la asignatura está repartida de la siguiente manera:

  50% Teoría + 50% Prácticas

- La calificación obtenida en el examen teórico constituye el 50% de la nota final de la asignatura. Semanalmente se evalúan los ejercicios prácticos en el laboratorio, constituyendo la totalidad de los mismos el 50% de la nota final de la asignatura. Cada ejercicio tendrá un factor de ponderación de acuerdo con su dificultad.
- Para superar las prácticas es necesario tener Aptas el 75% de las realizadas.
- La asignatura cuenta con dos convocatorias en Junio y una en Septiembre.
- Los exámenes constaran de:
  1.-) Parte asociada a las Prácticas.
  2.-) Parte asociada a la Teoría.
- Los alumnos que superen las Prácticas con la evaluación continua no tendrán que realizar la parte práctica del examen.
- La parte práctica consistirá en la escritura del código de una práctica. El alumno no dispondrá de ordenador. Se exigirá una completa correción del código. La superación de la parte práctica del examen es ELIMINATORIA. En el caso de no aprobarse, no se corregirá la parte de teoría.

3 Programa de Teoría

1. Fundamentos de Programación en Java.

   ¿Qué es la plataforma Java?. La Plataforma Base de Java. Aplicaciones y Applets. ¿Dónde se despliega la plataforma Java?. Una mirada al Interior de la Plataforma Java. La Máquina Virtual de Java. Las APIs de Java. Ambientes Java de Compilación y Ejecución. Principales Características del Lenguaje de Programación Java. Tipos de datos primitivos. Operadores Relacionales y Aritméticos. Arrays. Strings. Sentencias Break. Manipulación de Memoria y Recolección de Basura. Características que se han eliminado de C y C++.

2. Conceptos de Programación Orientada a Objetos.

   ¿Qué son los objetos?. Clases. Instanciación de un Objeto desde su clase. Métodos y Mensajes. Genéricidad. Encapsulación de Información. Definición de Clase. Componentes de una clase. Acceso a las variables y métodos de instancia. Sobrecarga de Métodos. Variables y Métodos de clase. Variables estáticas. Variables y Métodos Final. Control de Acceso. Objetos que tienen objetos. Recolección de Basura. Uso de arrays. Uso de Strings.

3. Jerarquía de Clases en Java.

   Constructores. Subclases. Herencia Simple y Jerarquía de clases. Interfaces del Lenguaje Java. Control de accesos. Packages. Variables de Clase y Métodos de Clase. Métodos Abstractos.

4. Gráficos e Interfaces de Usuario.

   La jerarquía de la clase AWT (Abstract Window Toolkit). Marcos. Programación dirigida por eventos. Directores de disposición. Paneles. Dibujo de figuras geométricas. Interfaces de Usuario (Botones, Etiquetas, Areas de Texto, Campos de Texto, Listas, Menues, Dialogos). Applets.

5. Gestión de errores con el uso de excepciones.

   Excepciones y Tipos de Excepciones. Manipulación de Excepciones (Creación, Lanzamiento, Recogida). La Clausula *finally*. Precauciones al utilizar Excepciones.

6. *Threads*.

   La Clase *Thread*. La Interface *Runnable*. Estados de un *thread*. Prioridades de los *thread*. Grupos de *thread*. Sincronización.

7. Multimedia.

   Audio. Imágenes. Secuencias de Imágenes. Reducción del efecto de salto. Computación Cliente/Servidor. Applets Clientes.

8. Entrada/Salida.

   La clase *Stream*. *InputStream* y *Reader*. *OutputStream* y *Writer*. Procesamiento de Ficheros externos. Flujos array. Flujos Filtro. Flujos de Datos. Flujos de Impresion. Flujos con buffer.

4 Programa de Prácticas

1. Cálculo del coste de una Hipoteca.

   Escribir un programa java que calcule los pagos mensuales de una hipoteca.

2. Números Racionales.

   Escribir una clase Rational para encapsular números racionales.

3. Array de Números Racionales.

   Considerando la clase Rational desarrollada en la práctica anterior. Escribir una clase cliente en la que se cree un array de números racionales generados de forma aleatoria.

4. Diseño y Extensión de Clases Abstractas.

   Diseño de una clase genérica que implemente las operaciones de suma y producto de matrices de cualquier tipo.

5. Manipulación de Eventos Simples.

   En esta práctica se proponen dos ejercicios: Escribir un programa que cree un marco (Frame) y muestre un recuadro sólido en el punto en el que se ha presionado el ratón. Escribir un programa que cree un marco que muestre un boton (Button) que al ser seleccionado cierre la ventana de la misma forma que se cierra si se presiona la "x"de la esquina superior derecha.

6. Dibujar un Reloj.

   En esta práctica se proponen escribir un programa que cree un marco que contenga un dibujo de un reloj que muestre la hora actual, así como la misma hora en formato numérico.

7. Ejecución de un Programa como un Applet y como una Aplicación.

   Modificar la práctica anterior de manera que se pueda ejecutar como applet y como aplicación.

8. Manipulación de Excepciones.

   Dado un programa Java que calcula el conciente entre dos números, modificarlo para atrapar las excepciones que se presentan.

9. Implementación de Flujos de Control Múltiples (Threads).

   En las prácticas anteriores se han desarrollado aplicaciones y applets que dibujan un reloj y muestran la fecha y la hora actuales. El reloj no camina después de mostrar la hora. La cuestión que nos plantearemos en está práctica es: ¿Cómo podemos conseguir que el reloj muestre la hora actualizada cada segundo?.

10. Productor/Consumidor.

    El objetivo de esta prácitca es mostrar que se puede trabajar con múltiples flujos de control de forma que sean asíncronos e independientes.

11. Multimedia.

    Añadir a la práctica del reloj que se ha venido desarrollando una alarma.

12. Entrada/Salida.

    El objetivo de esta práctica es elaborar una clase que nos permita recoger de la entrada estándar distintos tipos de datos y mostrarlos con el formato correcto en la salida estándar.

5 Bibliografía

1. Y. Daniel Liang. *Introduction to Java Programming.* QueE&T. 1999
2. S. Davis. *Aprenda Java ya. McGraw-Hill.* Microsoft Press. 1996.
3. J. Gosling, H. McGilton. *The Java Language Enviroment.* A White Paper. Sun Microsystems. Java Soft. 1996.
4. D. Flanagan. *Java en pocs Palabras.* McGrawHill-O'Reilyy. 1998.
5. J.F. Macary, C. Nicolas. *Programación Java.* Eyrolles.

6 Normas de Funcionamiento de las Prácticas

- Si se quiere cambiar de grupo es necesario encontrar a alguien de otro grupo y realizar un intercambio.
- La ficha de la asignatura se formalizará durante la primera práctica, para ello hay que presentar una foto y el justificante de trabajo.
- Se seguirán las normas generales de realización de prácticas del DEIOC, que se relacionan a continuación:
  - La asistencia a las prácticas es obligatoria.
  - Las prácticas se calificarán como Aptas y No Aptas. Las prácticas Aptas tendrán distintas valoraciones según el criterio de cada profesor.
  - Las prácticas a las que el alumno no asista se calificarán como No Aptas.
  - Para aprobar las prácticas, el alumno deberá tener calificadas como Aptas, al menos, el 75% de las prácticas realizadas en la asignatura.
  - Un alumno sólo podrá presentarse al examen de teoría cuando tenga aprobadas las prácticas. En caso de no tener superado al menos el 75% de las prácticas, los alumnos tendrán derecho a un examen de prácticas.
  - En cada asignatura deberá especificarse claramente cuales son los objetivos de cada práctica, así como el lenguaje de programación y las herramientas a utilizar.
  - La calificación de las prácticas supondrá como máximo, un 20% de la nota final.
  - Las calificaciones de prácticas podrán ser guardadas, a criterio de cada profesor, por un máximo de 2 años.
  - Estas normas de prácticas serán comunes a todas las asignaturas que el departamento imparta.