

Córdoba, 03-Marzo-2008

Universidad Tecnológica Nacional - Facultad Regional Córdoba -
 Departamento de Sistemas

Modalidad Académica

Asignatura: Paradigmas de Programación

Area: Programación

Coordinador de cátedra: Ing. Tymoschuk, Jorge Pablo

Objetivos Generales

- Comprender los fundamentos de los paradigmas de programación básicos que son utilizados por los lenguajes de programación actuales.
- Conocer el modelo formal u semiformal subyacente de cada paradigma y la forma en que el mismo es incorporado en un lenguaje de programación concreto.
- Aplicar los diferentes paradigmas en la solución de problemas.

Contenidos

Unidad I - Programación Orientada a Objetos avanzada

Introducción a colecciones	3
Interfaces de colecciones	
4Interfaces de colección e implementaciones separadas .	
5Colecciones y las interfaces de iteración en la librería Java	
5Colecciones concretas	8
Listas enlazadas (LinkedList).	8
public class LinkedListTest	12
public class SortedList extends LinkedList	13
Array de listas (ArrayList).	16
Acceso a los elementos de un ArrayList.	17
Inserción y eliminación de elementos intermedios	18
Una implementación de ArrItems extendiendo ArrayList	21
Conjuntos de hash	22
Usando class HashSet	23
Arboles	25
Árboles rojo-negro	25
Comparación de objetos	26
Class TreeSet	28
Arbol TreeSet ordenado según interfaz Comparable.	28
Arbol TreeSet ordenado según objeto Comparator	29
Conjuntos de bits	31
public class CribaEras extends BitSet	31
Herencia y Polimorfismo en Java. Caso concreto.	33
public class Progression	33
class ArithProgression extends Progression	34
class GeomProgression extends Progression	35
class FibonacciProgression extends Progression	35
class Tester	36
Tiempo de dictado .	3 semanas

Unidad II - Programación Orientada a Eventos (GUI Builder)

Introducción.	2
Creando el proyecto	2

Primeros pasos, Agregando un par de JPanel's al JFrame	3
Agregando títulos de borde (title borders) al panel	4
Incorporando componentes individuales, rótulo (JLabel).	5
Agregando, alineando, anclando	6
Dimensionado de componentes	6
Indentar varios JRadioButtons debajo de un JLabel	7
Tratamiento de eventos	8
Preparando la incorporación del tratamiento de eventos.	9
Incorporando el tratamiento de eventos.	12
Trazando una ejecución del Proyecto Contactos	14

Unidad II - PROGRAMACIÓN ORIENTADA A EVENTOS (sin GUI Builder)

Introducción	1
Los Applets	1
La Clase JApplet	2
El Ciclo de Vida de un Applet	2
"Hola Applet"	3
Páginas HTML y Etiquetas	3
Cargando el applet en el navegador	4
Componentes y Swing	5
Componentes y Contenedores	6
La clase JComponent	6
La clase JPanel	7
Componentes Básicos	7
Etiquetas (JLabel)	7
Botones (JButton)	7
Campos de edición (JTextField)	8
Casillas de verificación (JCheckBox)	8
listas desplegadas (JComboBox)	8
Listas (JList)	8
Contenedores Básicos	10
Paneles (JPanel)	10
Paneles Desplazables (JScrollPane)	10
Ventanas (JFrame)	10
Ventanas de Diálogo (JDialog)	11
Diálogos de Opción (JOptionPane)	11
Un Applet con Componentes	12
Manejo de Eventos	14
El Modelo de Delegación	14
Respondiendo al Ratón	15
Escuchas y Adaptadores	17
Principales Tipos de Eventos	18
Eventos de ratón (MouseEvent)	18
Eventos de teclado (KeyEvent)	18
Eventos de acción (ActionEvent)	19
Eventos de ventana (WindowEvent)	21
Un Applet con Eventos	22
Una Ventana con Eventos	23
Parte práctica	27
Ejercicio 1, 2, 3, 4, 5, 6	27
Ejercicio 7, 8, 9	28
Resolución de Ejercicios	29
	Tiempo de dictado .	2 semanas

Unidad III - Programación Concurrente

Introducción	2
Ventajas del procesamiento multihilos	2

Desafíos en programación multihilos	2
Exclusión mutua	3
Sincronización	3
Calendarización de hilos, punto muerto	3
Hilos en Java	4
class ParImpar extends Thread	5
public class BounceThread	6
La interfaz Runnable	9
Interrupción de threads	9
Estados de un thread	10
Cómo salir del estado bloqueado	13
Threads muertos, Threads servidores, Grupos	13
Prioridades de los threads	14
Threads egoístas.	16
Sincronización, Comunicación sin sincronización	17
public class UnsynchBankTest	18
Bloqueo de objetos	21
Los métodos wait y notify	22
public class SynchBankTest	25
Puntos muertos	27
CANCELACIÓN DE UN HILO	30
ESPERA A QUE UN HILO FINALICE	32
Bloques sincronizados	33
Métodos estáticos sincronizados	34
Por qué están censurados los métodos stop() y suspend()?	35
El modelo productor/consumidor	37
Caso 1 - Sincronización a nivel de instancia	37
Caso 2 - Sincronización combinada de instancia/clase	40
Usando Swing como interfaz de usuario (Barra de progreso)	43
Evaluacion Final 06-12-2006	47
Tiempo de dictado	3 semanas

Unidad IV - Programación Distribuida

Introducción a la Programación Distribuida	3
Introducción al paradigma	6
Introducción a los objetos remotos	6
Introducción a INTERNET.	10
Terminología Internet	10
Direcciones Internet y dominios	12
public class InetAddressTest	12
Protocolos en TCP/IP	13
SERVICIOS EN INTERNET	14
PÁGINAS WEB, Qué es HTML	15
URL's, Enlaces entre páginas	16
Un poco de turismo.html	17
Formularios	19
Entrada básica de datos	19
Parámetros ocultos, Enviar datos.	20
Una consulta a mi profesor	21
XML, XHTML, PÁGINAS WEB DINÁMICAS	23
Conexión con un servidor	24
Implementación de clientes	26
Implementación de servidores	27
public class EchoClient, public class EchoServer	28
Servicio a varios clientes	30
public class ThreadedEchoServer, ThreadedEchoHandler	31
Un programa Chat	32
public class Cliente	35

public class Servidor, ServidorHilo	41
Envío de correo electrónico	44
public class MailTest, class MailTestFrame	46
Conexiones URL, public class ParseURL	49
Uso de URLConnection para recuperar información	50
public class URLConnectionTest	51
J2EE, Introducción	52
ARQUITECTURA J2EE MULTICAPA	53
Concepto de componente en J2EE	54
Contenedores J2EE	56
Tipos de Contenedores, Responsabilidad de la Capa Cliente	57
La capa Web, La capa EJB	58
SERVLETS, Características de un servlet	59
ESTRUCTURA DE UN SERVLET	60
Ciclo de vida de un servlet	61
Un servlet sencillo	62
Software necesario para ejecutar un servlet	63
INCLUIR PROCESOS ESCRITOS EN JAVA	63
PROCESAR FORMULARIOS	66
Tipos de peticiones	67
Petición HTTP GET, POST	67
LEER LOS DATOS ENVIADOS POR EL CLIENTE	68
DESCRIPTOR DE DESPLIEGUE	73
Iniciación de un servlet, seguimiento de una sesion	73
COOKIES	74
INICIACIÓN DE UN SERVLET	74
public class ContadorCook extends HttpServlet	75
Combinando documentos HTML, Servers, Cookies ...	76
Java Server Pages (JSP)	82
Ciclo de vida de una página JSP	84
Objetos implícitos, ámbito de atributos	84
APLICACIONES WEB UTILIZANDO JSP	85
Hipermercado TodoTodo (JSP, Colaboración Salvador Celia)	88
Carta.java	89
Comida.java	89
Hipermercado.html	89
Atención.JSP	90
Despacho.JSP	91
Web.XML	92
Java Server Pages Standard Tag Library (JSTL) (Salvador celia)	93
Apéndice A - INSTALACION DEL CONTENEDOR DE SERVLET/JSP TOMCAT 5	96
Apéndice B - EJECUTAR UN SERVLET EN EL SERVIDOR	97
Tiempo de dictado	5 semanas

Unidad V - Programación Funcional

Un poco de Historia.....	3
El lenguaje Haskell	5
A.INTRODUCCIÓN.....	5
B.¿QUE ES HASKELL?.....	5
C.¿POR QUE USAR HASKELL?.....	5
D.DESARROLLO FUTURO DE HASKELL.....	6
E.HASKELL EN ACCION.....	6
1-Tipos.....	6
1.1 Información de tipo.....	7
1.2 Tipos predefinidos.....	7
1.3 Funciones.....	7
2.-El entorno de Haskell - HUGS.....	7

3. Funciones.....	9
4. Listas.....	9
5. Tuplas	10
6. Ecuaciones con guardas.....	10
7. Definiciones locales.....	10
8. Expresiones <i>lambda</i>	11
9. Disposición del código.....	11
10.-Tipos definidos por el usuario.....	12
11. Tipos Recursivos.....	14
12.-Entrada/Salida.....	15
13. Sobrecarga y Clases en Haskell.....	17
14. Evaluación Perezosa(Lazy)	19
15. Ejercicios Prácticos	19
Anexo "La enseñanza de Haskell"	21
Bibliografía	22
Tiempo de dictado . 1 semanas	

UNIDAD VI - PROGRAMACIÓN LÓGICA..... ¡ERROR! MARCADOR NO DEFINIDO.

INDICE.....	¡ERROR! MARCADOR NO DEFINIDO.
1. INTRODUCCION A LA PROGRAMACION LOGICA	¡ERROR! MARCADOR NO DEFINIDO.
2. LOGICA PROPOSICIONAL.....	¡ERROR! MARCADOR NO DEFINIDO.
3. INTRODUCCION A PROLOG	¡ERROR! MARCADOR NO DEFINIDO.
4. CALCULO DE RELACIONES.....	¡ERROR! MARCADOR NO DEFINIDO.
4.1 RELACIONES	¡ERROR! MARCADOR NO DEFINIDO.
4.2 RELACIONES EN PROLOG.....	¡ERROR! MARCADOR NO DEFINIDO.
5. BUSCANDO LAS SOLUCIONES	¡ERROR! MARCADOR NO DEFINIDO.
5.1. EL CONTROL EN PROLOG	¡ERROR! MARCADOR NO DEFINIDO.
5.2. EL BACKTRACKING.....	¡ERROR! MARCADOR NO DEFINIDO.
5.3. EL CORTE (!).....	¡ERROR! MARCADOR NO DEFINIDO.
5.4. LA NEGACIÓN COMO FRACASO	¡ERROR! MARCADOR NO DEFINIDO.
5.5. REALIZANDO EL SEGUIMIENTO DE UN PROGRAMA	¡ERROR! MARCADOR NO DEFINIDO.
6. ESTRUCTURA DE UN PROGRAMA EN PROLOG	¡ERROR! MARCADOR NO DEFINIDO.
7. OBJETOS COMPUESTOS	¡ERROR! MARCADOR NO DEFINIDO.
8. RECURSIVIDAD	¡ERROR! MARCADOR NO DEFINIDO.
9. LISTAS	¡ERROR! MARCADOR NO DEFINIDO.
PARTE PRACTICA.....	¡ERROR! MARCADOR NO DEFINIDO.
SOLUCIONES.....	¡ERROR! MARCADOR NO DEFINIDO.
REFERENCIAS Y FUENTES	¡ERROR! MARCADOR NO DEFINIDO.
Tiempo de dictado . 1 semanas	

Bibliografía Principal

- [1] Apunte teórico/práctico de la cátedra
- [2] Guía de prácticos en BBS/WEB.
- [3] Java 2, Cay S. Horstman/Gary Cornell, Ed. Prentice Hall
- [4] Wang, Paul S., "Java, con POO y aplicaciones en la Web"
- [5] Fco Javier Ceballos Java 2 Interfaces gráficas y aplicaciones Internet

Esquema de Regularización/Promoción:

- **Prácticos** (Grupales) Mínimo de 4(cuatro) y hasta 5(cinco). Los prácticos se receptan una única vez. El último de ellos es obligatorio y debe tratar un tema de **Programación distribuida**. El promedio de prácticos es una nota a considerarse en el esquema de Regularización.
- **Parciales** (Individuales) Dos parciales aprobados, siendo que 1(unos) puede recuperarse. El recuperatorio no recubre la nota del parcial. Es una nota más y se promedia.

Modalidad de examen final

- Los alumnos con promedio 8 y nota mínima 7 tienen promovida la parte práctica del examen final. O sea que rinden un examen de evaluación de conceptos, estrategias, interpretación, etc. También pueden optar por presentar un trabajo de investigación, constituyendo un grupo de a lo sumo 3(tres) integrantes.
- Los alumnos con promedio inferior a 8 rinden previamente parte práctica. Aprobada, pasan al teórico.
- Se toma cualquier punto del programa.
- Cualquier profesor de la cátedra puede examinar a cualquier alumno.

Fechas

- **Evaluaciones parciales.** Quedan establecidas las siguientes fechas:
1er parcial. Unidades I y II, 1 semana después de concluido el dictado.
- **2do parcial.** Unidades III y IV(inclusive Sockets), la penúltima semana de dictado
- **Recuperatorio:** Semana siguiente a la finalización del dictado, o fecha exámenes generales 6to llamado, a opción del alumno, una única vez (Solo se recupera un parcial).

Trabajos de investigación

El tema puede ser indicado por el profesor a cargo del curso o ser una iniciativa del grupo, autorizada por el profesor. Se entiende que se tratará de programación funcionando en computadora. El profesor a cargo es responsable por :

- 1) Definir la forma de presentación del trabajo.
- 2) Controlar su originalidad.
- 3) Orientar/Asesorar el grupo de trabajo.
- 4) Evaluar lo realizado.
- 5) Presentar un informe al director de cátedra.

Ing. TYMOSCHUK, Jorge
Coordinador de Cátedra