



Asignatura	Algoritmos y Estructuras de Datos
Ciclo Lectivo	2010
Vigencia del programa	Ciclo lectivo 2010
Plan	2008
Área	Programación
Carga horaria semanal	5 hs
Anual/ cuatrimestral	Anual
Coordinador de Cátedra	Ing. Tymoschuk, Jorge Pablo
Objetivos de la Materia	Desarrollar la capacidad de razonamiento y lógica, abordando problemas reales, analizándolos, identificando los conceptos involucrados en su tratamiento e implementándolos como clases, detallando el comportamiento necesario a nivel de métodos y atributos, verificando su correcto funcionamiento.

Programa Analítico

Unidad Nro. 1: Problemas con tipos de datos simples

Objetivos Específicos: Desarrollar la capacidad de razonamiento y lógica necesarios para identificar problemas algorítmicos que usan tipos de datos simples y resolverlos. Esto implica: abordar problemas reales, analizarlos, definir la estrategia de su resolución, explicitar los algoritmos necesarios y codificarlos en un lenguaje de programación.

Contenidos:

Objetivos de la Unidad	3
Pasos a seguir en la resolución de un problema	3
Introducción a la POO	4
Que es la programación Orientada a Objetos	5
Componentes básicos de la POO	5
Características de la POO	5
Programación estructurada, Programación orientada a objetos	9
Lenguaje Java, Características	11
Instalación de java	13
Compilación y ejecución de un programa	13
GRAMÁTICA DEL LENGUAJE JAVA	14
COMENTARIOS	14
IDENTIFICADORES, PALABRAS CLAVE Y RESERVADAS	15
Concepto de Dato	15
Concepto de Información, dif. entre datos e información	16
Tipos de Datos Simples, Variables	17
Genero de las variables, asignación, inicialización	18
Operadores: unarios, binarios	19
Separadores	21
Estructuras de control: secuencial, selectivas	21
Alternativa simple, doble	21
Alternativa múltiple	23
Estructuras de Control: Bucles	25
El Bucle While	25
Terminaciones anormales de un ciclo	26
Bucles controlados por centinela	27
Diseño eficiente de bucles	27
Bucles controlados por banderas	29
Sentencias de quiebre de control	30



Repetición: El Bucle For . . .	31
Repetición: El Bucle do-while . . .	34
Bucles Anidados . . .	35
Ejercicios propuestos para el alumno . . .	36

Bibliografía: Material la cátedra publicado por Educa y disponible en el sitio labsys.frc.utn.edu.ar, Sitios de las Cátedras, Algoritmos y Estructuras de Datos conjuntamente con todo el código Java utilizado en los prácticos.

Criterio de Evaluación: La resolución del problema planteado por el enunciado implica varias etapas:

- Analizar y comprender la codificación previa lista que el alumno recibe en el pre-proyecto
- Codificar e integrar con la codificación previa los puntos específicamente pedidos en el enunciado
- Ejecutar el proyecto completo

En este proceso estamos evaluando los siguientes aspectos:

- Capacidad de comprensión de codificación no propia.
- Capacidad de complementación con la codificación propia
- Capacidad de depurar errores detectados en la fase de ejecución

Nota: Este criterio de evaluación es común a todas las unidades de la asignatura

Bibliografía: Material la cátedra publicado por Educa y disponible en el sitio labsys.frc.utn.edu.ar, Sitios de las Cátedras, Algoritmos y Estructuras de Datos conjuntamente con todo el código Java utilizado en las prácticas.

Unidad Nro. 2: Problemas con Tipo Abstracto de Datos

Objetivos específicos:

Desarrollar la capacidad de razonamiento y lógica necesarios para identificar problemas algorítmicos que usan tipo abstracto de datos y resolverlos. Esto implica: abordar problemas reales, analizarlos, definir la estrategia de su resolución, explicitar los algoritmos necesarios y codificarlos en un lenguaje de programación.

En esta Unidad trataremos con algoritmos que resuelven problemas usando el comportamiento de una única clase.

Contenido

Programación orientada a Objetos, Principios de diseño . . .	3
Abstracción, Tipo Abstracto de Datos . . .	3
Encapsulamiento, Modularidad . . .	4
CLASES, ABSTRACCIONES CON PROCEDIMIENTOS Y FUNCIONES	5
ESTRUCTURA GENERAL . . .	5
DECLARACIÓN Y DEFINICIÓN, EL CUERPO DE LA CLASE	6
Acceso a miembros, Ciclo de Vida de los Objetos.	7
DECLARACIÓN DE LOS MIEMBROS DE UNA CLASE . . .	8
Modificadores de acceso a miembros de clases . . .	8
Separación de la interfaz . . .	10
ATRIBUTOS DE UNA CLASE . . .	12
MÉTODOS DE UNA CLASE . . .	13
Llamadas a métodos . . .	14
El objeto actual (puntero this) . . .	15
Pasaje de Parámetros . . .	16
Métodos sobrecargados . . .	18
Resolución de llamada a un método . . .	18



Métodos constructores	19
Constructores, Por defecto, Con argumentos, Copiadores	19
Caso especial	20
public class	22
Interfases	23
Implementación de interfaces	23

Bibliografía: Ídem Unidad 1

Criterio de Evaluación: Ídem Unidad 1

Unidad Nro. 3: Estrategias de Resolución

Objetivos específicos

Desarrollar la capacidad de razonamiento y lógica necesarios para que el alumno pueda visualizar y desarrollar la solución de problemas como comportamiento de varias clases de objetos, los cuales interaccionan entre si mediante mensajes. Entrenar a los alumnos en el uso de herramientas que facilitan y racionalizan estos desarrollos, como ser herencia, polimorfismo, colecciones.

Entrenar al alumno en el uso de programación existente, como ser el de búsqueda y ordenamiento en arreglos. Instruir al alumno en el uso de soluciones alternativas al tratamiento iterativo, como es la recursividad.

Contenido

Interacción de objetos, Abstracción	3
Abstracción en software, Relaciones entre objetos	4
Composición usando una sucesión de objetos	6
Composición usando una sucesión de objetos	8
Tratamiento de frases	12
Agrupar objetos.	14
Colecciones de tamaño fijo - Arreglos	15
Declaración, creación, uso de variables arreglo	16
Composición usando un arreglo de objetos	17
Composición usando una secuencia de números	18
Colecciones de tamaño flexible, agenda personal.	21
Características importantes de ArrayList.	23
Procesar colección completa, ciclo for-each	24
Recorrer una colección	26
Herencia	29
Usar herencia	31
Jerarquías de herencia	32
Herencia e inicialización	34
Agregando nuevos elementos a una jerarquía existente.	35
Ventajas, Subtipos, Subtipos y asignación	36
Variables polimórficas, Enmascaramiento de tipos	37
La clase Object	38
Tipo estático y tipo dinámico	39
Búsqueda dinámica del método	40
Llamada a super en métodos	42
Métodos polimórfico	43
Paquetes (package)	46
Tratamiento de Excepciones	48
Lanzamiento de excepciones	49
Atrapado de excepciones	50
Generando nuestras propias excepciones.	53
Algoritmos de búsqueda, recorrido y ordenamiento	
Búsqueda en arreglos	55
Búsqueda secuencial	57
Búsqueda binaria	58



Nociones de Complejidad Computacional ,	
Operaciones Primitivas	59
Conteo, Notación asintótica	61
Ordenamiento - Introducción,	
Algoritmos básicos y mejorados.	63
class OrdBurb, OrdSac, OrdPein, ordPeinCol	64
Determinación del Orden de Complejidad. Tiempos	68
Arrays multidimensionales	70
Acceso a elementos mediante bucles	71
Matriz de objetos Item.	73
Estructuras lineales.	79
Pila, concepto, diversas implementaciones	79
public class Tiempos	81
Cola, concepto, diversas implementaciones	83
Implementación usando nodos enlazados.	84
Recursividad, concepto, formulación recursiva del algoritmo	85

Bibliografía: Ídem Unidad 1

Criterio de Evaluación: Ídem Unidad 1

Unidad Nro. 4: Almacenamiento de Datos

Objetivos específicos: Instruir y dar las herramientas para que el alumno pueda guardar sus datos de forma persistente, usando flujos a archivos conteniendo datos primitivos y objetos. Instruir al alumno como esta persistencia se logra usando Bases de datos: Como conectarse, y mínimamente seleccionar y actualizar datos en una base simple, de dos o tres tablas.

Nota: Este último punto se incorpora en el presente año, y se dictará a título piloto en uno o dos cursos.

Contenido

Flujos	3
Clases File Input/Output Stream	5
Procesamiento Básico	5
Clases ByteArray Input/Output Stream	6
Clases Pipe Input/Output Stream	6
Clases Filtro	7
Clases Data Input/Output Stream	7
La Clase File, ejemplos.	10
Archivos secuenciales	12
Creación de un archivo secuencial	11
Consulta de un archivo secuencial	14
Actualización de un archivo secuencial	16
Archivos de acceso aleatorio	16
Creación de un Random AccessFile	17
Métodos de posicionamiento	17
Creando un archivo directo	19
Class MiCasa	19
Class CreaCountry	20
Consultas en un archivo directo	21
Class MiCasona	22
Class ConsultaCountry	22
Flujos de tipo Objeto	24
Almacenamiento de objetos de distinto tipo.	29
Recordando interfaces	30
Implementando interfaces	30
Acceso a bases de datos. (a dictar en un par de cursos piloto)	
Instalación del driver. Conectarse.	



Accesos de consulta y actualización a una Base de datos	
Bibliografía: Ídem Unidad 1 Criterio de Evaluación: Ídem Unidad 1	
Metodología de enseñanza y aprendizaje	Teórico y práctico desarrollado en laboratorio. Los conceptos teóricos son inmediatamente llevados a prácticos en lenguaje Java y luego son profundizados en clases prácticas (laboratorio) y prácticos grupales (Domicilio).
Sistema de evaluación	Los prácticos grupales son aceptados una vez que funcionan. (Se ejecutan correctamente). Los parciales se evalúan de acuerdo al grado de cumplimiento de la meta propuesta (Corrección en línea sobre la computadora). Se verifican resultados y codificación.
Condiciones de regularidad	Totalidad de prácticos grupales (mínimo cuatro) mas dos parciales aprobados. Se puede recuperar un parcial. Se promedian prácticos (1 nota) y parciales(2 notas)
Modalidad de examen final	Práctico, si aprobado se pasa al teórico. El práctico se realiza en máquina. Si el alumno tiene promedio ≥ 8 (ocho) y ninguna nota < 7 (siete) rinde examen teórico o puede presentar un trabajo de investigación consistente en un desarrollo en lenguaje Java de tópicos que representen una novedad o un avance sobre lo desarrollado en aula.
Actividades en laboratorio	La totalidad de las clases se desarrollan en laboratorio. Los enunciados se analizan y se programan en lenguaje Java usando un entorno de programación (Netbeans, Blue J).
Horas/año totales de la asignatura	160
Cantidad de horas prácticas totales	112 (Agregar 8 hs destinadas a las evaluaciones parciales)
Cantidad de horas teóricas totales	48
Tipo de formación práctica (marque la que corresponde y si es asignatura curricular -no electiva-)	<input checked="" type="checkbox"/> Formación experimental – asignatura curricular, no electiva <input type="checkbox"/> Resolución de problemas de ingeniería <input type="checkbox"/> Actividades de proyecto y diseño <input type="checkbox"/> Prácticas supervisadas en los sectores productivos y /o de servicios
Cantidad de horas afectadas a la formación práctica indicada	112
Descripción de los prácticos	Unidad Nro 1 - Problemas con tipos de datos simples Clase con comportamiento (10 métodos) para obtener diversos datos a partir de los lados del triángulo. Diversos ejemplos simples de ciclos while. Diversos ejemplos simples de ciclos for. Clase tratando varias figuras geométricas. Ciclos anidados. Comparativo de dibujo de triángulos isósceles usando ciclos simples y anidados. Unidad Nro 2 - Problemas con tipos de datos abstractos Se lee un texto, se detectan palabras repetidas, se las



	<p>exhibe y se las suprime del texto original. Leer sucesivas líneas y concatenarlas; informar cuantas líneas. Leer sucesivas líneas, concatenarlas. Informar orden de la línea mas larga, cuantos caracteres contiene, contenido. Utilizamos el concepto de separador para detectar palabras. Ordenarlas palabras de la línea en orden decreciente de longitud. Suprimir clave de texto. Mostrar texto resultante.</p> <p>Unidad Nro 3 - Estrategias de Resolución Una agenda basica, ABM de Strings sobre ArrayList Recorriendo agenda con ciclos for each, while e iterador Búsqueda de claves en agenda con ciclos for each, while e iterador Contabiliza alternancias consonante/vocal en una cadena de texto Análisis comparativo de cuatro distintas implementaciones de una pila Análisis comparativo de tres métodos de ordenamiento Inserción ordenada en una colección implementada en un ArrayList Comportamiento de un array de ítems Cuatro programas sencillos usando matrices multidimensionales. Una colección ArrayList con elementos ArrItems. Su mantenimiento Comportamiento adicional sobre la colección anterior. Búsqueda binaria sobre objetos Item. Búsqueda secuencial idem. Programas simples usando comportamiento de las clases String y StringBuffer Comportamiento de una matriz bidimensional de números enteros Ordenamiento mediante método burbuja Ordenamiento mediante método peinado Ordenamiento mediante método sacudida Comportamiento para tratar palabras dentro de textos. Comportamiento para tratar una sucesión de números enteros. Variante de SecuNum usando ciclo For Proyecto tratando series numéricas. Uso de herencia y polimorfismo Interaccionando con dos LinkedList se genera lista ordenada, etc Comportamiento tratando sucesión de caracteres Variante de SecuNum usando ciclo do while Varios proyectos con comportamiento simple sobre vectores conteniendo enteros</p> <p>Unidad Nro 4 - Almacenamiento de Datos Comportamiento para generar/agregar y leer archivos secuenciales tipo texto. Comportamiento para grabar/leer archivos con datos de objetos. Usando la clase File para obtener atributos de archivos Usando herencia para definir clases y sus especializaciones</p>
--	---



	Almacenando recuperando objetos que contienen otros objetos Usando serialización para manipular flujos de objetos Grabando objetos diversos, su reconocimiento en lectura Comportamiento combinado de los dos proyectos anteriores. Conectarse y formulación de sentencias SQL simples sobre una base de datos.
Criterios de evaluación de los prácticos	Mínimo de un 80% de los puntos especificados en el enunciado funcionando correctamente (en ejecución).
Formato de presentación de los prácticos	Programación Java en un entorno adecuado, normalmente NetBeans.
Cronograma de actividades de la asignatura , incluyendo semana prevista para cada práctico	Unidad Nro 1 - 6(seis) semanas, finaliza el 23 de abril 2010 Unidad Nro 2 - 6(seis) semanas, finaliza el 04 de junio 2010 Unidad Nro 3 - 12(doce) semanas, finaliza el 24 de setiembre 2010 Unidad Nro 4 - 6(seis) semanas, finaliza el 05 de noviembre 2010
Descripción de metodología propuesta de consultas y cronograma de consultas	Uso de email y foro del aula virtual, todo el año. No hay un cronograma que limite o restrinja la posibilidad de los alumnos de realizar consultas.
Plan de integración con otras asignaturas	El contenido de esta asignatura es pre-requisito del contenido dictado en Paradigmas de Programación.
Bibliografía Obligatoria	Material la cátedra publicado por Educa y disponible en el sitio labsys.frc.utn.edu.ar, Sitios de las Cátedras, Algoritmos y Estructuras de Datos conjuntamente con todo el código Java.
Bibliografía Complementaria	Estructuras de Datos y Algoritmos en Java, Goodrich/ Tamassia, CECSA



Distribución de docentes por curso	Curso	Turno	Día y Horas	Profesor	JefeTrab.Práct.	Ayudante
	1k01	Mañana	Mar 5,6,7 Mie 1,2	Fritelli Valerio	Bett Federico	Romaní German
1k02	Mañana	Lun 1,2,3 Mie 1,2	Ligorria Karina	Tartabini Marcela	Párraga Adriana	
1k03	Mañana	Lun 6,7 Vie 4,5,6	Serra Silvio	Steffolani Felipe		
1k04	Mañana	Mar 3,4 Jue 1,2,3	Fritelli Valerio	Teicher Romina	Fernandez Julieta	
1k05	Mañana	Mar 4,5,6 Vie 6,7	Ligorria Karina	Carena Gustavo	Romaní German	
1k06	Mañana	Lun 4,5,6 Jue 6,7	Guzman Analia	Corso Cynthia	Luna Karina	
1k07	Mañana	Mie 1,2,3 Vie 6,7	Ligorria Karina	Teicher Romina		
1k08	Mañana	Lun 1,2,3 Mie 1,2	Guzman Analía	Steffolani Felipe	Cárdenas Marina	
1k90	Mañana	Mar 5,6,7 Mie 1,2	Guzman Analia	Corso Cynthia	Colacioppo Nicolas	
1k09	Tarde	Mar 4,5,6 Vie 5,6	Tymoschuk Jorge	Brochero Carlos	Colacioppo Nicolas	
1k10	Tarde	Mar 3,4 Vie 0,1,2	Arias Silvia	Párraga Adriana	Luna Karina	
1k11	Tarde	Lun 2,3 Mie 1,2,3	Tymoschuk Jorge	Carena Gustavo	Castillo Julio	
1k12	Noche	Lun 0,1 Vie 1,2,3	Serra Silvio	Tartabini Marcela	Párraga Adriana	
1k13	Noche	Mie 5,6 Vie 1,2,3	Cresta Tomas	Bett Federico	Parraga Adriana	