

PROGRAMACIÓN AVANZADA
Profesores: Cristian Chiang - Loreto Telgie
Primer Semestre 2019

PROPÓSITO

Al finalizar el curso el estudiante logrará describir conceptos, formular algoritmos y soluciones a problemas aplicando el paradigma de la programación orientada al objeto, utilizando estructuras de datos (arreglos y listas con nexo) y técnicas de resolución de problemas como recursión, dividir para conquistar y backtracking. El lenguaje de programación utilizado es Java.

RESULTADOS DE APRENDIZAJE

Los estudiantes al finalizar la asignatura deben ser capaces de:

1. Construir software orientado al objeto, legible y documentado, incluyendo testing (casos de prueba y prueba unitaria) y control de excepciones
2. Construir contenedores genéricos implementados con arreglos y con nexos
3. Resolver problemas utilizando contenedores ya implementados en Java (arraylist, linkedlist) e iteradores
4. Resolver problemas utilizando técnicas de recursión, dividir para conquistar y backtracking

CONTENIDO

1. Introducción
2. Programación orientada al objeto en Java
3. Colecciones: Listas con nexo, Contenedores implementados en java (arraylist y linkedlist) e Iteradores
4. Estrategias de diseño de algoritmos: Recursión, Dividir para Conquistar y Backtracking

APUNTES

<http://www.ucn.cl/onlineucn> (En la plataforma EDUC@ v2.0)

BIBLIOGRAFIA**Textos Guía:**

- Objects First with Java, David J. Barnes & Michael Kölling, Fifth edition, Prentice Hall / Pearson Education, 2012.
- Loreto Telgie. Apuntes de la asignatura Programación Avanzada, 2016

Texto Complementario

- Data Structures and Problem Solving Using Java. Weiss, Mark Allen. Fourth edition, Addison-Wesley, 2009.

Máximo 5 minutos de atraso en la llegada a la clase

Requisito de asistencia a cátedra: 70%

Requisito de asistencia a ayudantías en laboratorio: 70% (5 de 7)

Requisito de asistencia a ayudantías en sala: 70% (5 de 7)

EXIGENCIAS

• 3 Pruebas de Cátedra

□	P0:	Sa 30 Marzo, 9:55 hrs.	Nivelación	
□	P1:	Sa 25 Mayo, 9:55 hrs	RA1 y parte del RA2	Capítulos 1 y 2
□	P2:	Sa 15 Junio, 9:55 hrs	RA1, resto del RA2 y RA3	Capítulo 3
□	P3:	Ju 11 Julio, 9:55 hrs.	RA4	Capítulo 4

Examen Recuperativo: Mi 17 Julio 9:55 hrs

• 5 Talleres obligatorios de grupos de a 2 estudiantes del mismo paralelo. Desde el taller 2 al 5, cada uno con nota ≥ 4.0 . Todas las entregas de los talleres son obligatorias. La documentación de los talleres debe estar en inglés. Enunciado del Taller 3 en inglés

Java con Eclipse o Netbeans

- **T1** Java con arreglos y subprogramas (sin orientación al objeto) **Mi 20 Marzo – Ma 16 Abril**
 Análogo al último taller de introducción a la programación
 Entrega 1: Estructuras para los datos (15%) Ma 26 Marzo
 Entrega 2: Estructura del programa (15%) Lu 1 Abril
 Entrega 3: Código (70%) Ma 16 Abril

- **T2:** Java sin herencia, navegabilidad y contenedores implementados con arreglos.
 Se usa arquitectura propuesta

Ma 16 Abril – Vi 10 Mayo

 Entrega 1 (15%): Modelo del dominio + Diagrama de clases dominio aplicación
 Ma 23 Abril
 Entrega 2 (20%): Contratos + Diagrama de Clases Ma 30 Abril
 Entrega Final (65%) Vi 10 Mayo

- **T3:** Java con herencia, polimorfismo, navegabilidad y contenedores implementados con arreglos
 Con arquitectura **Vi 10 Mayo - Ju 6 Junio**
 Entrega 1 (10%): Modelo del dominio + contratos (10%) Vi 17 Mayo
 Entrega 2 (15%): Diagrama de Clases (15%) Vi 24 Mayo
 Entrega Final (75%): Código Mi 5 Junio

- **T4:** Java con herencia, polimorfismo, navegabilidad y contenedores implementados con listas con
 nexos y uso de arraylist, linkedlist e iteradores. Con arquitectura **Mi 5 Junio – Ma 25 Junio**
 Entrega 1 (25%): Modelo del dominio + contratos + DC Mi 12 Junio
 Entrega Final (75%): Código Ma 25 Junio

- **T5:** Recursión, Dividir para conquistar y Backtracking **Ma 25 Junio – Lu 22 Julio**
 Entrega 1: Códigos recursivos, incluyendo dividir para conquistar
 Vi 5 Julio
 Entrega 2: Modelamiento del problema de backtracking Vi 12 Julio
 Entrega 3: Código del problema de backtracking Lu 22 Julio
 Todas las entregas con la misma ponderación

Nota: Inmediatamente después de cada entrega, se publica una alternativa de solución

Promedio Talleres

Si $T2 \geq 4,0$ y $T3 \geq 4,0$ y $T4 \geq 4,0$ y $T5 \geq 4,0$
 Prom. Talleres = $(T1 + T2 + T3 + T4 + T5) / 5$
 Sino
 PromTalleres = $\text{mínimo}\{T1, T2, T3, T4, T5\}$

No hay recuperativa por reprobación de talleres

Grupos de talleres: 2 estudiantes del mismo paralelo

Ayudantes de Talleres: Michael Bravo-Brian Pardo

- Diseñar y corregir talleres, coordinado con Cristian Chiang
- Atender alumnos por consultas de talleres
- Atender alumnos por reclamos de talleres

Ayudante de corrección Loreto : Juan Carlos Lamas

Ayudante de corrección Cristian: Juan Martinez

- Corregir trabajos clases y controles

Ayudante de Cátedra (Laboratorio/Sala): Loreto: Juan Martinez **Cristian:**

- Diseño de ayudantías y laboratorios coordinado con Loreto
- Hacer ayudantías y laboratorios

Planificación Ayudantías de Cátedra (Sala/Laboratorios): Viernes F (paralelo Loreto)

#	Fecha	Objetivo	Apoyo a	S/L
1	Vi15 Marzo	<ul style="list-style-type: none"> • Ruteo de programa Java • Instalación y uso de Java, Eclipse o Netbeans • Construir programas Java sencillos: <ul style="list-style-type: none"> ➤ Imprimir “hola mundo” • Debugging 	Nivelación	Laboratorio
2	Vi 22 Marzo	Construir programas Java con: <ul style="list-style-type: none"> • Con arreglos • Deben partir dibujando la estructura de los datos y la estructura del programa • Hagan uso de algoritmos de ordenamiento, búsqueda, inserción, etc. • Debugging 	Nivelación	Laboratorio
3	Vi 29 Marzo	Construir programas Java con: <ul style="list-style-type: none"> • Con arreglos • Deben partir dibujando la estructura de los datos y la estructura del programa • Hagan uso de algoritmos de ordenamiento, búsqueda, inserción, etc. • Hagan uso de archivos y subprogramas • Debugging 	Nivelación	Laboratorio

4	Vi 5 Abril	<p>Ejercicios pequeños:</p> <ul style="list-style-type: none"> • Modelo del dominio • Diagrama de clases del dominio de la aplicación, con sus atributos, constructor y métodos get y set. • Sin herencia, ni asociaciones entre clases • Construcción del código de las clases y main creando objetos • Uso de contenedores • Sin arquitectura <p>Ejemplo: Listas de objetos</p>	Prueba 1	Sala
5	Vi 26 Abril	<p>Ejercicios pequeños:</p> <ul style="list-style-type: none"> • Diagrama de clases • Con asociaciones entre clases • Sin herencia • Con contenedores • Con arquitectura propuesta 	Prueba 1	Sala
6	Vi 3 Mayo	<p>Ejercicio grande:</p> <ul style="list-style-type: none"> • Modelo del dominio • Diagrama de clases del dominio de la aplicación • Con asociaciones entre clases • Sin herencia • Con contenedores • Contratos • DC • Con arquitectura propuesta 	Prueba 1	Sala
7	Vi 10 Mayo	<p>Ejercicio simple, incluyendo todo</p> <p>Lista de objetos, donde el objeto puede ser de tipo A1 o A2, ambos subclases de A</p>	Prueba 1	Laboratorio
8	Vi 17 Mayo	Ejercicio tipo prueba, incluyendo todo	Prueba 1	Sala
9	Vi 24 Mayo	<p>Problemas simples que utilicen listas con un nexa. Incluir:</p> <ul style="list-style-type: none"> • Diagrama de clases • Código para la clase nodo y clase lista. • Uso de arquitectura 	Prueba 2	Sala
10	Vi 7 Junio	<p>Problema con listas con nexa, usando arquitectura</p> <p>El mismo problema de la ayudantía anterior, usando cualquier variación en listas con nexa.</p>	Prueba 2	Sala
11	Vi 14 Junio	Problema tipo prueba, incluyendo Arraylist, Linkedlist, interface list, Iteradores	Prueba 2	Laboratorio
12	Vi 21 Junio	<ul style="list-style-type: none"> • Ruteo de algoritmos recursivos • Algoritmos recursivo • Sin dividir para conquistar 	Prueba 3	Laboratorio
13	Vi 28 Junio	<ul style="list-style-type: none"> • Dividir para conquistar • Modelamiento de problema de backtracking 	Prueba 3	Sala
14	Vi 5 Julio	Construcción del código del problema de backtracking modelado la ayudantía anterior	Prueba 3	Laboratorio