

CARRERA	Tecnicatura Superior en Análisis de Sistemas con Orientación en Redes Informáticas			
PLAN DE ESTUDIOS	Resolución 2561/06			
DOCENTE	Gustavo Daniel Reyes			
UNIDAD CURRICULAR	Período	Año	Régimen	Hs. Cátedra
Programación IV (PIV)	2023	3ro.	Anual	3 (tres)

1. FUNDAMENTACIÓN

La cátedra de Programación IV es un espacio fundamental de la carrera Tecnicatura Superior en Análisis de Sistemas con Orientación a Redes Informáticas, puesto que brindará al futuro profesional, el conocimiento suficiente para desarrollar programas de aplicación siguiendo el Paradigma Orientado a Objetos (POO) como alternativa a la programación procedimental.

Intentar conjugar los distintos aspectos teóricos y prácticos, con el objetivo de que el alumno complete el ciclo del problema hasta obtener su solución con computadora, mediante el planteamiento de un programa como una modelización de los elementos del mundo real, es una tarea que no puede estar sujeta a improvisaciones, y que conlleva una serie de actividades de decisión y planificación complejas y variadas, para las cuales es necesario tener una adecuada respuesta metodológica.

En esta asignatura se introducirán los conceptos fundamentales y necesarios para abstraer un problema del mundo real, aplicando el paradigma de POO, convertirlo en un modelo computable, escribir una solución simbólica del mismo y traducirla en una solución concreta y ejecutable. La suma de estos conocimientos, permitirá a los futuros profesionales incorporar el enfoque basado en objetos para el desarrollo de programas, reconocer las diferencias que dicho paradigma presenta con respecto a la Programación Imperativa y dominar los fundamentos del Lenguaje de Programación Java.

En el desarrollo de la Unidad Curricular, los alumnos incorporarán habilidades de modelado de datos e información, las cuales se integrarán y complementarán con las capacidades adquiridas en los trayectos formativos de las cátedras de “Programación III” “Análisis de Sistemas I”, “Análisis de Sistemas II” y “Práctica Profesional II”, completando de esta manera las competencias de los alumnos necesarias para el modelado de sistemas software.

Como resultado de dicha integración, los alumnos desarrollarán una aplicación software, construida aplicando el enfoque de la POO y utilizando Java. Esta aplicación deberá desarrollarse por grupos de alumnos y constituirá su proyecto grupal de cursada. La especificación de requisitos del trabajo a desarrollar será fijada por la Cátedra.

2. OBJETIVOS

- Dominar los conceptos principales del paradigma de POO.
- Comprender las diferencias entre el paradigma OO y otros paradigmas de programación.
- Adquirir habilidades en diseño, construcción y ejecución de algoritmos basados en POO.
- Entender las reglas semánticas, sintácticas y las estructuras de control del lenguaje de programación Java.
- Describir e implementar los conceptos básicos y avanzados de la POO utilizando el lenguaje de POO Java.
- Diseñar un conjunto de clases interrelacionadas, de cuya interacción se obtenga una solución a un problema del mundo real.
- Implementar aplicaciones Orientadas a Objetos que creen y manipulen archivos para guardar y recuperar información.
- Comprender los mecanismos avanzados para la prevención y gestión de errores de ejecución de programas.
- Adquirir los conceptos fundamentales para la construcción de programas con interfaces gráficas de usuario.
- Reconocer los aspectos más importantes de los diagramas de estructura estática de UML, para el modelado de clases, objetos, sus atributos y relaciones.
- Dominar un Entorno de Desarrollo integrado (IDE) para la creación de aplicaciones software en Java.

3. NÚCLEOS PROGRAMÁTICOS

A) INTRODUCCIÓN AL PARADIGMA DE PROGRAMACIÓN ORIENTADO A OBJETOS (POO)

¿Qué es un Paradigma? ¿Qué es un paradigma de programación? El paradigma orientado a objetos. Pilares de la POO: Abstracción, Encapsulación, Herencia y Polimorfismo.

¿Por qué es importante la POO? Desventajas de POO. Conceptos básicos de POO. Reusabilidad y extensibilidad.

BIBLIOGRAFÍA ESPECÍFICA NUCLEO A			
AUTOR	TÍTULO	EDITORIAL	LUGAR Y AÑO DE ED.
Booch, G.	Análisis y diseño OO con aplicaciones	Addison-Wesley	España, 1996
Joyanes, L y Zhonero, M.	Programación en Java 6. Algoritmos, Estructuras de datos y POO	Mc Graw Hill	México, 2011
Martin, J y Odell, J.	Análisis y Diseño Orientado a Objetos	Prentice-Hall	México, 1992
Reyes, D.	Apuntes de Cátedra.	Del autor	Argentina, 2021

B) INTRODUCCIÓN AL LENGUAJE JAVA

La plataforma Java. El lenguaje de programación Java. Tipos de programas en Java. El proceso de compilación en Java. ¿Qué es una API y para qué sirve? Versiones y distribuciones de Java. Evolución del Java estándar. El JDK y el JRE.

Estructura de un programa en Java. La clase principal y el método main. Comentarios. Identificadores. Tipos primitivos de dato. Formato de representación de los datos numéricos reales. Declaraciones de variables. Declaración de variables final o constantes. Conversiones entre tipos de datos.

Operadores: concepto. Operador de asignación; aritméticos; aritméticos incrementales; aritméticos combinados; de relación; lógicos o booleanos; condicional. Operador concatenación de cadenas. Separadores. Expresiones. Prioridad entre operadores. Sentencias selectivas o condicionales. Sentencias de control del flujo de un programa. Sentencia break y continue.

BIBLIOGRAFÍA ESPECÍFICA NUCLEO C			
AUTOR	TÍTULO	EDITORIAL	LUGARY AÑO DE ED.
Deitel, P y Deitel, H.	Cómo programar en Java, 9na. edición	Pearson	México, 2012
Dean, J y Dean, S.	Introducción a la programación con Java	Mc Graw Hill	México, 2009
Joyanes, L y Zhonero, M.	Programación en Java 6. Algoritmos, Estructuras de datos y POO	Mc Graw Hill	México, 2011
Lewis, J. y Chase, J.	Estructura de datos con Java	Pearson	México, 2009
Horstmann, C.	Java concepts, 6th ed.	John Wiley	EEUU, 2009
Wu, T	Introducción a la Programación Orientada a Objetos con Java	Mc Graw Hill	España, 2000
Reyes, D.	Apuntes de Cátedra.	Del autor	Argentina, 2021

C) INTRODUCCIÓN A LA SINTAXIS DE JAVA

Declaración y uso de métodos: Parámetros, return y void. Sobrecarga de métodos. Objetos y clases. Modificadores de visibilidad. Clases anidadas e internas. El operador instanceof. Paquetes: Identificación de un paquete, Acceso a los componentes de un paquete, Modificadores de acceso. Tipos de atributos: Variables de instancia, de clase y variables finales. Constructores. Constructor por defecto. Declaración de un constructor. Llamada a un constructor desde otro constructor. Constructores sobrecargados. La referencia this. Tipos de métodos: de instancia y de clase. Parámetros. Variables locales. Referencias estáticas. Herencia. Jerarquía de clases. Acceso a miembros y herencia. La referencia super. La clase Object. Herencia y constructores. Constructores de subclases. Sobrescritura de métodos. Polimorfismo. Casting de objetos y herencia. Clases y métodos finales. Clases abstractas. Métodos abstractos. Interfaces. Declaración e implementación de una interfaz en una clase. Implementación de múltiples interfaces en una clase. Jerarquía entre interfaces. Interfaz como un tipo de dato. Interfaces y clases abstractas. El Garbage Collector (recolector de basura). El método finalize.

BIBLIOGRAFÍA ESPECÍFICA NUCLEO C			
AUTOR	TÍTULO	EDITORIAL	LUGARY AÑO DE ED.
Deitel, P y Deitel, H.	Cómo programar en Java, 9na. edición	Pearson	México, 2012
Dean, J y Dean, S.	Introducción a la programación con Java	Mc Graw Hill	México, 2009
Joyanes, L y Zhonero, M.	Programación en Java 6. Algoritmos, Estructuras de datos y POO	Mc Graw Hill	México, 2011
Lewis, J. y Chase, J.	Estructura de datos con Java	Pearson	México, 2009
Weiss, M.	Estructuras de datos en Java, 4ta. ed.	Pearson	México, 2013
Reyes, D.	Apuntes de Cátedra.	Del autor	Argentina, 2021

D) COLECCIONES Y MANEJO DE ERRORES EN JAVA

Arrays en Java. Arrays unidimensionales. Recorrido con for-each. Arrays como parámetros de métodos. Arrays como resultado de métodos. Arrays multidimensionales. La clase Arrays. Array con números aleatorios. La clase Random. Array de objetos. Clases Anónimas en Java. Genéricos en Java. Parámetros de tipo. Clase genérica. Métodos Genéricos. Herencia de genéricos. Conversión de tipos con genéricos. Sobrecarga de Métodos Genéricos. Tipos crudos (raw types). Comodines. Clases autoreferenciales. Genéricos y clases anónimas.

Colecciones en Java. El Framework de Colecciones de Java. Tipos de colecciones en Java. Métodos comunes. Implementaciones más comunes de las interfaces de colecciones. Las interfaces Iterable e Iterator. Las interfaces Comparable y Comparator.

Fundamentos de la gestión de excepciones. Tipos de excepciones. Excepciones no capturadas. Uso de try y catch. Descripción de una excepción. Cláusulas catch múltiples. Sentencias try anidadas. Cláusulas throw, throws y finally. Excepciones integradas en Java. Excepciones propias. Excepciones en Colecciones. Excepciones comunes en operaciones de colecciones. Excepciones integradas en operaciones de colecciones. Casos prácticos y ejemplos.

BIBLIOGRAFÍA ESPECÍFICA NUCLEO D			
AUTOR	TÍTULO	EDITORIAL	LUGARY AÑO DE ED.
Deitel, P y Deitel, H.	Cómo programar en Java, 9na. edición	Pearson	México, 2012
Dean, J y Dean, S.	Introducción a la programación con Java	Mc Graw Hill	México, 2009
Joyanes, L y Zhonero, M.	Programación en Java 6. Algoritmos, Estructuras de datos y POO	Mc Graw Hill	México, 2011
Lewis, J. y Chase, J.	Estructura de datos con Java	Pearson	México, 2009
Naftalin, M. y Wadler, P.	Java Generics and Collections	O'Reilly Media	EEUU, 2006
Weiss, M.	Estructuras de datos en Java, 4ta. ed.	Pearson	México, 2013
Reyes, D.	Apuntes de Cátedra.	Del autor	Argentina, 2021

E) FUNDAMENTOS DE PROGRAMACIÓN GUI CON JAVA

Programación Dirigida por Eventos. Interfaces Gráficas de Usuario. Interfaces Gráficas de Usuario (GUI) en Java. AWT y Swing. Jerarquía de Swing. Componentes y contenedores. Clase JFrame. Componentes en Java. Componente JLabel. Componente JTextField. Clases internas. Clases internas anónimas. Componente JButton. Ventanas de diálogo y clase JOptionPane. Gestión de Eventos: el modelo de delegación de eventos en Java. Eventos, Fuentes y Oyentes. Clases de eventos. Interfaces de Oyentes. Clases adaptadoras. Registro de Eventos. Color. Agrupación de clases GUI.

Distribución de componentes. Diseño GUI. Posicionamiento de los componentes: Layouts. Interfaz gráfica para el desarrollo de GUI. Gestores de posicionamiento. Clase JPanel. Componente JTextArea. Componente JcheckBox. Componente Jradio-Button. Componente JComboBox. Más componentes de Swing. Hilos (threads).

BIBLIOGRAFÍA ESPECÍFICA NUCLEO E			
AUTOR	TITULO	EDITORIAL	LUGAR Y AÑO DE ED.
Deitel, P y Deitel, H.	Cómo programar en Java, 9na. edición	Pearson	México , 2012
Dean, J y Dean, S.	Introducción a la programación con Java	Mc Graw Hill	México, 2009
Lewis, J. y Chase, J.	Estructura de datos con Java	Pearson	México, 2009
Robinson, M & Vorobiev, P.	Swing, 2nd. Edition	Manning	EEUU, 2003
Schildt, H.	Swing a beginner's guide	Mc Graw Hill	EEUU, 2009
Weiss, M.	Estructuras de datos en Java, 4ta. ed.	Pearson	México , 2013
Reyes, D.	Apuntes de Cátedra.	Del autor	Argentina, 2021

4. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Trabajos grupales e individuales.
- Elaboración de informes escritos.
- Exposiciones grupales e individuales
- Aprendizaje cooperativo.
- Aula invertida.

5. EVALUACIÓN

CRITERIOS

- Participación comprometida en los trabajos propuestos, ya sean individuales y/o grupales.
- Compromiso y responsabilidad en las tareas a desarrollar.
- Presentación de trabajos en tiempo y forma.
- Manejo adecuado de la bibliografía.
- Coherencia, claridad conceptual, integración y pertinencia tanto en oratoria como en el discurso escrito.
- Dominio, comprensión y transposición de los contenidos del programa.
- Capacidad de resolución de problemáticas presentadas.
- Capacidad de organizar, coordinar y elaborar la tarea encomendada.
- Aplicación de conocimientos y dominio de técnicas, mediante la respuesta a preguntas y la resolución de problema.
- Participación pertinente en el desarrollo de las clases.

INSTRUMENTOS DE EVALUACIÓN

Para valorar los aprendizajes de los alumnos y sistematizar dichas valoraciones, se utilizarán los siguientes instrumentos:

- Evaluaciones orales y escritas.
- Desarrollo de proyecto de software.
- Trabajos prácticos.
- Coloquios.

PARA OBTENER LA REGULARIDAD

Para regularizar la asignatura, el alumno deberá cumplir con las condiciones de regularización descriptas a continuación, las cuales serán expuestas a los alumnos durante la presentación de la Unidad Curricular y del plan de trabajo de la misma:

- Presentar y aprobar, con nota 5 o superior, de un proyecto grupal de desarrollo de software para una determinada organización. El software obtenido debe ser funcional, conforme a los requisitos establecidos en el proyecto, en función de las necesidades de la institución cliente y su desarrollo será en Java, usando POO.
- Tener el 80% de asistencia a las clases o de participación en el aula virtual.
- Presentar a término y aprobar el 90% de los trabajos prácticos o actividades propuesto.
- Aprobar el 100% de los exámenes parciales propuestos ó de sus respectivos exámenes recuperatorios. La nota mínima obtenida no deber ser menor de 5 (cinco), en una escala de 1 (uno) a diez (10).
- Obtener una calificación final mínima de 5 (cinco) puntos, en una escala de 1(uno) a 10(diez), al terminar el cursado.

PARA OBTENER LA PROMOCIONALIDAD

Según lo establece la normativa vigente (RAM), la Unidad Curricular NO ES promocionable.

PARA RENDIR LA UNIDAD CURRICULAR EN CALIDAD DE LIBRE

En base a la normativa vigente, las condiciones establecidas para rendir esta Unidad Curricular en calidad de alumno libre son:

- Presentación y aprobación de la totalidad de los trabajos prácticos o de un trabajo práctico integrador equivalente, en ambos casos, propuestos por la Cátedra. Para ello, el alumno deberá requerirlo a la Cátedra con treinta (30) días de anticipación a la fecha del examen final, a fin de obtener el conjunto de trabajos prácticos que deberá resolver. Los mismos deberán ser entregados al docente responsable cinco (5) días antes de la fecha de examen final.
- Superada la instancia anterior, el alumno estará en condiciones de rendir un primer examen final escrito sobre aspectos teóricos (25%) y prácticos (75%) de la Unidad Curricular.
- En caso de aprobar el examen escrito, el alumno estará en condiciones de rendir un segundo examen final de características similares al que rinden los alumnos regulares, sobre aspectos teóricos de la materia. Este examen será tomado en forma oral y versará sobre las temáticas abordadas en el programa de la asignatura, desde la perspectiva de la bibliografía específica.

PARA RENDIR LA UNIDAD CURRICULAR EN CALIDAD DE ALUMNO REGULAR

En base a la normativa vigente (RAM), para rendir esta Unidad Curricular en calidad de alumno regular el alumno deberá rendir un primer examen final escrito sobre aspectos teóricos de la misma. Este examen versará sobre las temáticas comprendidas en el programa de la asignatura, desde la perspectiva de la bibliografía obligatoria. Los criterios de evaluación se basan en la acreditación de la lectura de la bibliografía, la argumentación sólida que utilice la terminología y los conceptos del campo de estudio y la apropiación crítica de los mismos.

6. MATERIALES PARA EL APRENDIZAJE

- Gabinete de Informática, con la suficiente cantidad de computadoras con el siguiente software debidamente instalado: Windows 7+, Java Development Kit 19+, BlueJ 5+ y Apache NetBeans 17+.
- Apuntes de Cátedra.
- Guía de trabajos prácticos
- Pizarra y fibrones.
- Notebook.
- Cañón proyector.

7. DISTRIBUCIÓN EN EL TIEMPO DE LOS CONTENIDOS PROGRAMADOS

	Marzo	Abril	Mayo	Junio	Agosto	Septiembre	Octubre	Noviembre
INTRODUCCIÓN AL PARADIGMA DE POO	R R							
INTRODUCCIÓN AL LENGUAJE JAVA			F J					
INTRODUCCIÓN A LA SINTAXIS DE JAVA								
1ER. EXAMEN PARCIAL				J				
RECUPERATORIO DEL 1ER. EXAMEN PARCIAL								
MANEJO DE ERRORES					R R			
PROGRAMACIÓN DE GUI CON JAVA								
2DO. EXAMEN PARCIAL								
RECUPERATORIO DEL 2DO. EXAMEN PARCIAL								

Referencias:

- F** Día feriado o no laborable
- J** Jornada educativa
- R** Receso de verano/invierno

.....
 Firma del docente