

FACULTAD DE CIENCIA Y TECNOLOGÍA

ESCUELA DE INGENIERÍA ELECTRÓNICA

1. Datos generales

Materia: PROGRAMACIÓN II
Código: ELE0505
Paralelo: D
Periodo : Septiembre-2023 a Febrero-2024
Profesor: PATIÑO LEON PAUL ANDRES
Correo electrónico: andpatino@uazuay.edu.ec

Nivel: 5

Distribución de horas.

Docencia	Práctico	Autónomo: 16		Total horas
		Sistemas de tutorías	Autónomo	
32	32	16	0	80

Prerrequisitos:

Código: ELE0404 Materia: PROGRAMACIÓN I

2. Descripción y objetivos de la materia

Se pretenden cubrir los temas iniciales de la programación orientada a objetos con Java como son sentencias de control, métodos, arreglos, encapsulación, herencia, polimorfismo, manejo de excepciones, interfaces gráficas de usuario y manejo de archivos.

Esta materia se articula estrechamente con el resto de materias de programación de la carrera, además está también muy relacionada con materias como Sistemas Embebidos.

Esta materia está orientada al desarrollo de aplicaciones autónomas utilizando el lenguaje de programación Java con las técnicas de la programación orientada a objetos, cubriendo la programación del lado del cliente, permitiéndole a los estudiantes aplicar éstas técnicas a la adquisición de datos y sistemas de control para equipos electrónicos, lo cual determina su importancia y su contribución al perfil de egreso de los futuros profesionales de la carrera.

3. Objetivos de Desarrollo Sostenible



4. Contenidos

1.01.	Introducción. Su primer programa en Java: imprimir una línea de texto.
1.02.	Modificación de nuestro primer programa Java. Mostrando texto con printf.
1.03.	Otra aplicación: sumando enteros. Conceptos de memoria.
1.04.	Aritmética. Tomando de decisiones: operadores de igualdad y relacionales.
2.01.	Introducción. Declarando una clase con un método e instanciando un objeto de una clase.
2.02.	Declarando un método con un parámetro. Variables de instancia, métodos establecer y métodos obtener.
2.03.	Tipos primitivos contra tipos referencia. Inicializando objetos con constructores.
2.04.	Números de punto flotante y el tipo double.

3.01.	Introducción. Algoritmos. Pseudocódigo. Estructuras de control.
3.02.	Sentencia de selección simple if. Sentencia de selección doble if...else. Sentencia de repetición while.
3.03.	Formulando algoritmos: repetición controlada por contador, repetición controlada por centinela, sentencias de control anidadas.
3.04.	Operadores de asignación compuesta, operadores de incremento y decremento, tipos primitivos.
4.01.	Introducción. Esencia de la repetición controlada por contador. Sentencia de repetición for.
4.02.	Ejemplos usando la sentencia for. Sentencia de repetición do...while.
4.03.	Sentencia de selección múltiple switch. Sentencias break y continue.
4.04.	Operadores lógicos. Resumen de programación estructurada.
5.01.	Introducción. Módulos de programas en Java. Métodos static, campos static y la clase Math.
5.02.	Declarando métodos con múltiples parámetros. Notas sobre declarar y usar métodos. Pila de llamadas a métodos y registros de activación.
5.03.	Promoción y conversión de argumentos. Paquetes de la API de Java. Ejemplo práctico: generación de números aleatorios.
5.04.	Ejemplo práctico: un juego de probabilidad (introducción a las enumeraciones). Alcance de las declaraciones. Sobrecarga de métodos.
6.01.	Introducción. Arreglos. Declarando y creando arreglos. Ejemplos usando arreglos. Ejemplo práctico: simulación para barajar y repartir cartas.
6.02.	Sentencia for mejorada. Pasando arreglos a métodos. Ejemplo práctico: Clase LibroCalificaciones usando un arreglo para almacenar calificaciones.
6.03.	Arreglos multidimensionales. Ejemplo práctico: Clase LibroCalificaciones usando un arreglo bidimensional. Listas de argumentos de longitud variable.
6.04.	Usando argumentos de línea de comandos. La clase Arrays. Introducción a las Colecciones y a la clase ArrayList.
7.01.	Introducción. Ejemplo práctico de la clase Tiempo. Controlando el acceso a miembros. Referenciando a los miembros del objeto actual con la referencia this.
7.02.	Ejemplo práctico de la clase Tiempo: constructores sobrecargados. Constructores predeterminados y sin argumentos. Observaciones sobre los métodos establecer y obtener.
7.03.	Composición. Enumeraciones. Recolección de basura y el método finalize. Miembros de clase static.
7.04.	Importación static. Variables de instancia final. Ejemplo práctico de la clase Tiempo: creando paquetes. Acceso de paquete.
8.01.	Introducción. Superclases y subclases.
8.02.	Miembros protected. Relación entre las superclases y las subclases.
8.03.	Constructores en las subclases. Ingeniería de software con herencia.
8.04.	La clase Object.
9.01.	Introducción. Ejemplos de polimorfismo.
9.02.	Demostración de comportamiento polimórfico. Clases y métodos abstractos.
9.03.	Ejemplo práctico: sistema de nómina utilizando polimorfismo. Métodos y clases final.
9.04.	Ejemplo práctico: creando y usando interfaces.
10.01.	Introducción. Ejemplo práctico: división por cero sin manejo de excepciones. Ejemplo práctico: manejando ArithmeticExceptions y InputMismatchExceptions.
10.02.	Cuando utilizar el manejo de excepciones. Jerarquía de excepciones en Java. El bloque finally.
10.03.	Limpieza de la pila y obtener información del objeto excepción. Excepciones encadenadas. Declarando nuevos tipos de excepción.
10.04.	Precondiciones y postcondiciones. Aserciones.
11.01.	Introducción. El nuevo look and feel de Java: Nimbus. Entrada/Salida simple basada en GUI con JOptionPane. Generalidades de los componentes de Swing.
11.02.	Mostrando texto e imágenes en una ventana. Campos de texto y una introducción al manejo de eventos con clases anidadas. Tipos de eventos comunes de la GUI e interfaces de escucha.
11.03.	Como funciona el manejo de eventos. JButton. Botones que mantienen el estado (JCheckBox y JRadioButton). JComboBox: usando una clase interna anónima para el manejo de eventos.
11.04.	JList. Listas de selección múltiple. Manejando eventos de ratón.
11.05.	Clases adaptadoras. Subclase de JPanel para dibujar con el ratón. Manejando eventos de teclas.
11.06.	Introducción a los administradores de esquemas (FlowLayout, BorderLayout, GridLayout). Uso de paneles para administrar esquemas más complejos. JTextArea.
12.01.	Introducción. Contextos y objetos gráficos.

12.02.	Control de color. Manipulando Fonts.
12.03.	Dibujando líneas, rectángulos y óvalos. Dibujando arcos.
12.04.	Dibujando polígonos y polilíneas. La API Java 2D.
13.01.	Introducción. Fundamentos de caracteres y cadenas.
13.02.	La clase String. La clase StringBuilder.
13.03.	La clase Character. Tokenizando cadenas.
13.04.	Expresiones regulares, la clase Pattern y la clase Matcher.
14.01.	Introducción. Archivos y flujos.
14.02.	La clase File.
14.03.	Archivos de texto de acceso secuencial.
14.04.	Serialización de objetos.

5. Sistema de Evaluación

Resultado de aprendizaje de la carrera relacionados con la materia

Resultado de aprendizaje de la materia

Evidencias

. Aplica conocimientos teóricos y prácticos para dar solución a problemas en el área, fomentando el desarrollo tecnológico de la ingeniería.

-Aplica conceptos fundamentales de la programación estructurada y de la programación orientada a objetos.

-Evaluación escrita
-Trabajos prácticos -
productos

. Maneja herramientas informáticas de uso general y específico dentro de la Ingeniería Electrónica.

-Reconoce cómo se llevan a la práctica los conceptos que subyacen a la programación estructurada y de la programación orientada a objetos.

-Evaluación escrita
-Trabajos prácticos -
productos

. Utiliza su creatividad en la búsqueda de soluciones a problemas.

-Implementa soluciones eficientes.

-Evaluación escrita
-Trabajos prácticos -
productos

Desglose de evaluación

Evidencia	Descripción	Contenidos sílabo a evaluar	Aporte	Calificación	Semana
Evaluación escrita	Evaluación	Introducción a aplicaciones Java, Introducción a clases, objetos, métodos y Strings, Métodos: un análisis más detallado, Sentencias de control: parte 1, Sentencias de control: parte 2	APORTE	7	Semana: 5 (16-OCT-23 al 21-OCT-23)
Trabajos prácticos - productos	Trabajos	Introducción a aplicaciones Java, Introducción a clases, objetos, métodos y Strings, Métodos: un análisis más detallado, Sentencias de control: parte 1, Sentencias de control: parte 2	APORTE	3	Semana: 5 (16-OCT-23 al 21-OCT-23)
Evaluación escrita	Evaluación	Arreglos y ArrayList, Clases y objetos: un análisis más detallado, Manejando de excepciones: un análisis más detallado, Programación orientada a objetos: herencia, Programación orientada a objetos: polimorfismo	APORTE	7	Semana: 10 (20-NOV-23 al 25-NOV-23)
Trabajos prácticos - productos	Trabajos	Arreglos y ArrayList, Clases y objetos: un análisis más detallado, Manejando de excepciones: un análisis más detallado, Programación orientada a objetos: herencia, Programación orientada a objetos: polimorfismo	APORTE	3	Semana: 10 (20-NOV-23 al 25-NOV-23)
Evaluación escrita	Evaluación	Archivos, flujos y serialización de objetos, Cadenas, caracteres y expresiones regulares, Componentes GUI: parte 1, Gráficos y Java 2D	APORTE	7	Semana: 15 (al)
Trabajos prácticos - productos	Trabajos	Archivos, flujos y serialización de objetos, Cadenas, caracteres y expresiones regulares, Componentes GUI: parte 1, Gráficos y Java 2D	APORTE	3	Semana: 15 (al)
Evaluación escrita	Examen	Archivos, flujos y serialización de objetos, Arreglos y ArrayList, Cadenas, caracteres y expresiones regulares, Clases y objetos: un análisis más detallado, Componentes GUI: parte 1, Gráficos y Java 2D, Introducción a aplicaciones Java, Introducción a clases, objetos, métodos y Strings, Manejando de excepciones: un análisis más detallado, Métodos: un análisis más detallado, Programación orientada a objetos: herencia, Programación orientada a objetos: polimorfismo, Sentencias de control: parte 1, Sentencias de control: parte 2	EXAMEN	20	Semana: 19-20 (21-01-2024 al 27-01-2024)
Evaluación escrita	Supletorio	Archivos, flujos y serialización de objetos, Arreglos y ArrayList, Cadenas, caracteres y expresiones regulares, Clases y objetos: un análisis más detallado, Componentes GUI: parte 1, Gráficos y Java 2D, Introducción a aplicaciones Java, Introducción a clases, objetos, métodos y Strings, Manejando de excepciones: un análisis más detallado, Métodos: un análisis más detallado, Programación orientada a objetos: herencia,	SUPLETORIO	20	Semana: 20 (al)

Evidencia	Descripción	Contenidos sílabo a evaluar	Aporte	Calificación	Semana
		Programación orientada a objetos: polimorfismo, Sentencias de control: parte 1, Sentencias de control: parte 2			

Metodología

Descripción	Tipo horas
Los alumnos deberán desarrollar ejercicios, considerando el cumplimiento de los requisitos funcionales y no funcionales (si así se determina)	Autónomo
El método empleado para el desarrollo de la materia está basado en la solución de problemas de ejemplo en clases.	Total docencia
Se enuncian y explican los conceptos teóricos correspondientes, para luego presentar ejemplos de código fuente que afiancen los conocimientos. En los ejemplos se verifica la ausencia de errores y su correcta ejecución.	
De esta forma la clase es viva pues se recorre todo el proceso en la solución de un problema: diseño, programación e implementación del mismo.	

Criterios de evaluación

Descripción	Tipo horas
Se evaluará el cumplimiento de los requisitos y condiciones exigidas para la solución del trabajo indicado.	Autónomo
En los deberes, pruebas y examen final se evaluará el cumplimiento de los requisitos y condiciones exigidas para la solución del trabajo indicado. También se calificará la calidad de la solución propuesta.	Total docencia

6. Referencias

Bibliografía base

Libros

Autor	Editorial	Título	Año	ISBN
HARVEY DEITEL Y PAUL DEITEL	Pearson Prentice Hall	JAVA HOW TO PROGRAM	2012	978-0-13-257566

Web

Software

Revista

Bibliografía de apoyo

Libros

Autor	Editorial	Título	Año	ISBN
Cay Horstman y Gary Cornell	Pearson. Prentice Hall	Core Java 2 Volumen I Fundamentos	2006	

Web

Autor	Título	Url
StackOverflow	StackOverflow	https://stackoverflow.com
Oracle Technology Network	Java™ Platform, Standard Edition 8 API Specification	http://docs.oracle.com/javase/8/docs/api/index.html

Software

Autor	Título	Url	Versión
Apache Software Foundation, Oracle Corporation.	NetBeans	https://netbeans.org/downloads/	8.2

Docente

Director/Junta

Fecha aprobación: **08/09/2023**

Estado: **Aprobado**