UNIVERSIDAD DEL AZUAY

FACULTAD DE CIENCIA Y TECNOLOGÍA
ESCUELA INGENIERIA ELECTRONICA

1. Datos generales

Materia: PROGRAMACIÓN I

Código: CTE0227

Paralelo:

Periodo : Marzo-2020 a Agosto-2020

Profesor: PATIÑO LEON PAUL ANDRES

Correo electrónico andpatino@uazuay.edu.ec

Docencia	Práctico	Autónomo:		Total horas
		Sistemas de tutorías	Autónomo	
6				6

Prerrequisitos:

Código: CTE0203 Materia: MÉTODOS NUMÉRICOS

2. Descripción y objetivos de la materia

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.

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 Arquitectura de Computadoras, Microcontroladores II y Comunicación de datos y redes de computadoras.

3. Contenidos

1.	**Introducción a aplicaciones Java**
1.01.	Introducción. Su primer programa en Java: imprimir una línea de texto. (1 horas)
1.02.	Modificación de nuestro primer programa Java. Mostrando texto con printf. (1 horas)
1.03.	Otra aplicación: sumando enteros. Conceptos de memoria. (1 horas)
1.04.	Aritmética. Tomando de decisiones: operadores de igualdad y relacionales. (1 horas)
2.	**Introducción a clases, objetos, métodos y Strings**
2.01.	Introducción. Declarando una clase con un método e instanciando un objeto de una clase. (1 horas)
2.02.	Declarando un método con un parámetro. Variables de instancia, métodos establecer y métodos obtener. (1 horas)
2.03.	Tipos primitivos contra tipos referencia. Inicializando objetos con constructores. (1 horas)
2.04.	Números de punto flotante y el tipo double. (1 horas)
3.	**Sentencias de control: parte 1**
3.01.	Introducción. Algoritmos. Pseudocódigo. Estructuras de control. (1 horas)
3.02.	Sentencia de selección simple if. Sentencia de selección doble if...else. Sentencia de repetición while. (1 horas)
3.03.	Formulando algoritmos: repetición controlada por contador, repetición controlada por centinela, sentencias de control anidadas. (1 horas)
3.04.	Operadores de asignación compuesta, operadores de incremento y decremento, tipos primitivos. (1 horas)
4.	**Sentencias de control: parte 2**
4.01.	Introducción. Esencia de la repetición controlada por contador. Sentencia de repetición for. (1 horas)
4.02.	Ejemplos usando la sentencia for. Sentencia de repetición do...while. (1 horas)
4.03.	Sentencia de selección múltiple switch. Sentencias break y continue. (1 horas)
4.04.	Operadores lógicos. Resumen de programación estructurada. (1 horas)
5.	**Métodos: un análisis más detallado**

5.01.	Introducción. Módulos de programas en Java. Métodos static, campos static y la clase Math. (1 horas)
5.02.	Declarando métodos con múltiples parámetros. Notas sobre declarar y usar métodos. Pila de llamadas a métodos y registros de activación. (1 horas)
5.03.	Promoción y conversión de argumentos. Paquetes de la API de Java. Ejemplo práctico: generación de números aleatorios. (1 horas)
5.04.	Ejemplo práctico: un juego de probabilidad (introducción a las enumeraciones). Alcance de las declaraciones. Sobrecarga de métodos. (1 horas)
6.	**Arreglos y ArrayList**
6.01.	Introducción. Arreglos. Declarando y creando arreglos. Ejemplos usando arreglos. Ejemplo práctico: simulación para barajar y repartir cartas. (1 horas)
6.02.	Sentencia for mejorada. Pasando arreglos a métodos. Ejemplo práctico: Clase LibroCalificaciones usando un arreglo para almacenar calificaciones. (1 horas)
6.03.	Arreglos multidimensionales. Ejemplo práctico: Clase LibroCalificaciones usando un arreglo bidimensional. Listas de argumentos de longitud variable. (1 horas)
6.04.	Usando argumentos de línea de comandos. La clase Arrays. Introducción a las Colecciones y a la clase ArrayList. (3 horas)
7.	**Clases y objetos: un análisis más detallado**
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. (2 horas)
7.02.	Ejemplo práctico de la clase Tiempo: constructores sobrecargados. Constructores predeterminados y sin argumentos. Observaciones sobre los métodos establecer y obtener. (2 horas)
7.03.	Composición. Enumeraciones. Recolección de basura y el método finalize. Miembros de clase static. (2 horas)
7.04.	Importación static. Variables de instancia final. Ejemplo práctico de la clase Tiempo: creando paquetes. Acceso de paquete. (2 horas)
8.	**Programación orientada a objetos: herencia**
8.01.	Introducción. Superclases y subclases. (2 horas)
8.02.	Miembros protected. Relación entre las superclases y las subclases. (2 horas)
8.03.	Constructores en las subclases. Ingeniería de software con herencia. (2 horas)
8.04.	La clase Object. (2 horas)
9.	**Programación orientada a objetos: polimorfismo**
9.01.	Introducción. Ejemplos de polimorfismo. (1 horas)
9.02.	Demostración de comportamiento polimórfico. Clases y métodos abstractos. (1 horas)
9.03.	Ejemplo práctico: sistema de nómina utilizando polimorfismo. Métodos y clases final. (2 horas)
9.04.	Ejemplo práctico: creando y usando interfaces. (2 horas)
10.	**Manejando de excepciones: un análisis más detallado**
10.01.	Introducción. Ejemplo práctico: división por cero sin manejo de excepciones. Ejemplo práctico: manejando ArithmeticExceptions y InputMismatchEceptions. (1 horas)
10.02.	Cuando utilizar el manejo de excepciones. Jerarquía de excepciones en Java. El bloque finally. (1 horas)
10.03.	Limpieza de la pila y obtener información del objeto excepción. Excepciones encadenadas. Declarando nuevos tipos de excepción. (2 horas)
10.04.	Precondiciones y postcondiciones. Aserciones. (4 horas)
11.	**Componentes GUI: parte 1**
11.01.	Introducción. El nuevo lookand feel de Java: Nimbus. Entrada/Salida simple basada en GUI con JoptionPane. Generalidades de los componentes de Swing. (2 horas)
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. (2 horas)
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. (2 horas)
11.04.	Jlist. Listas de selección múltiple. Manejando eventos de ratón. (2 horas)
11.05.	Clases adaptadoras. Sublase de JPanel para dibujar con el ratón. Manejando eventos de teclas. (2 horas)
11.06.	Introducción a los administradores de esquemas (FlowLayout, BorderLayout, GridLayout). Uso de paneles para administrar esquemas más complejos. JtextArea. (2 horas)
12.	**Gráficos y Java 2D**
12.01.	Introducción. Contextos y objetos gráficos. (1 horas)
12.02.	Control de color. Manipulando Fonts. (1 horas)
12.03.	Dibujando líneas, rectángulos y óvalos. Dibujando arcos. (2 horas)
12.04.	Dibujando polígonos y polilíneas. La API Java 2D. (2 horas)
13.	**Cadenas, caracteres y expresiones regulares**
13.01.	Introducción. Fundamentos de caracteres y cadenas. (1 horas)
13.02.	La clase String. La clase StringBuilder. (1 horas)
13.03.	La clase Character. Tokenizando cadenas. (2 horas)

13.04.	Expresiones regulares, la clase Pattern y la clase Matcher. (4 horas)
14.	**Archivos, flujos y serialización de objetos**
14.01.	Introducción. Archivos y flujos. (1 horas)
14.02.	La clase File. (1 horas)
14.03.	Archivos de texto de acceso secuencial. (2 horas)
14.04.	Serialización de objetos. (2 horas)
14.05.	Clases adicionales del paquete java.io. (2 horas)
14.06.	Abriendo archivos con JfileChooser. (6 horas)

4. Sistema de Evaluación

Resultado de aprendizaje de la carrera relacionados con la materia

Resultado de aprendizaje de la materia	Evidencias
ab. Presentan de manera oral y escrita resultados finales o parciales derivados de alguna tarea encomendada	
-Presenta de manera escrita los deberes extra clases y pruebas indicadas para las diferentes evaluaciones.	-Proyectos -Trabajos prácticos - productos
af. Emplea el enfoque sistémico en el análisis y resolución de problemas	
-Desarrolla aplicaciones autónomas empleando el enfoque sistémico que se desprende del uso de la programación orientada a objetos utilizando Java.	-Evaluación escrita -Proyectos -Trabajos prácticos - productos
ai. Aplica lógica algorítmica en el análisis y solución de problemas en base los fundamentos de la programación	
-Desarrolla aplicaciones autónomas aplicando la lógica algorítmica basada en los fundamentos de la programación orientada a objetos utilizando Java.	-Proyectos -Trabajos prácticos - productos

Desglose de evaluación

Evidencia	Descripción	Contenidos sílabo a evaluar	Aporte	Calificación	Semana
Trabajos prácticos - productos	Trabajos		APORTE	3	Semana: 6 (06/05/20 al 11/05/20)
Evaluación escrita	Evaluación		APORTE	7	Semana: 6 (06/05/20 al 11/05/20)
Evaluación escrita	Evaluación		APORTE	7	Semana: 12 (17/06/20 al 22/06/20)
Trabajos prácticos - productos	Trabajos		APORTE	3	Semana: 12 (17/06/20 al 22/06/20)
Evaluación escrita	Evaluación		APORTE	7	Semana: 12 (17/06/20 al 22/06/20)
Trabajos prácticos - productos	Trabajos		APORTE	3	Semana: 16 (15/07/20 al 20/07/20)
Evaluación escrita	Examen		EXAMEN	20	Semana: 17-18 (21-07-2020 al 03-08-2020)
Evaluación escrita	Supletorio		SUPLETORIO	20	Semana: 19 (al)

Metodología

p { margin-bottom: 0.21cm; direction: ltr; color: rgb(0, 0, 10); }p.western { font-family: "Times New Roman",serif; font-size: 12pt; }p.cjk { font-family: "WenQuanYi Micro Hei","MS Mincho"; font-size: 12pt; }p.ctl { font-family: "Lohit Hindi","MS Mincho"; font-size: 12pt; }a.western:link { }a.ctl:link { }

El método empleado para el desarrollo de la materia está basado en la solución de problemas de ejemplo en clases. Se enuncian y explican los conceptos teóricos correspondientes y mediante el uso de una computadora y un proyector:
Se presenta un ejemplo de código fuente para explicar los conceptos teóricos a través del mismo.
Se compila dicho ejemplo para verificar la ausencia de errores en su escritura.
Se ejecuta para comprobar que se comporta como se espera que lo haga.
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

En los deberes, pruebas y exáme final se evaluará el cumplimiento de los requisitos y condiciones exigidas para la solución del trabajo indicado, así como la calidad de dicha solución de los estudiantes para desarrollar una aplicación en respuesta a un problema planteado por el profesor. En todos los casos serán inaceptables situaciones de plagio y copia (se tendrá en cuenta la ortografía y redacción).
La asistencia no puede considerarse como un aporte y el Reglamento de la Universidad del Azuay no contempla exoneración del examen final.

5. Referencias

Bibliografía base

Libros

Autor	Editorial	Título	Año	ISBN
Cay Horstman y Gary Cornell	Pearson. Prentice Hall	Core Java 2 Volumen II Características Avanzadas	2006	
Paul Deitel & Harvey Deitel	Pearson. Prentice Hall	Java How to Program	2011	
Bruce Eckel	Pearson. Prentice Hall	Piensa en Java	2007	9788489660342
Cay Horstman y Gary Cornell	Pearson. Prentice Hall	Core Java 2 Volumen I Fundamentos	2006	9786073238021

Web

Software

Bibliografía de apoyo

Libros

Web

Software

Docente

Director/Junta

Fecha aprobación: 19/03/2020

Estado: Aprobado