

TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE TLAXIACO

Documentación del programa de convertidor de bases

MATERIA:
Arquitectura de computadoras

CARRERA:
Ingeniería en sistemas computacionales

GRUPO:
5BS

PRESENTA:
Jaczibeth Cruz Ramirez

ASESOR
Ing. Edward Osorio Salinas

Tlaxiaco, Oax., Septiembre de 2024.



"Educación, ciencia y tecnología, progreso día con día"®

Introducción

En esta práctica, se desarrollará un programa en Java que realiza la conversión de números entre distintas bases numéricas utilizando una interfaz gráfica de usuario (GUI). El propósito de este ejercicio es familiarizarse con la manipulación de datos numéricos en diferentes bases y el uso de interfaces gráficas para la interacción con el usuario.

Los sistemas numéricos que se manejarán son binario, octal, decimal y hexadecimal, cada uno con aplicaciones y representaciones únicas en el ámbito de la informática. La conversión entre estos sistemas es una habilidad esencial para tareas que van desde el desarrollo de software hasta la depuración de sistemas digitales.

Creación de un programa de cambio de bases.

Para poder realizar nuestro programa, necesitamos analizar el lenguaje de programación a utilizar, en el cual he decidido trabajar en Apache NetBeans IDE16.

Una vez hecho eso se crea una clase en la cual se trabajara el programa, en este caso le di el nombre de “CambioBase”, ahora procedemos a realizar la codificación para llegar al resultado buscado.

Código implementado:

Importaciones:

Import javax.swing. JOptionPane: Esta línea importa la clase JOptionPane de la biblioteca

javax.swing. JOptionPane se utiliza para mostrar cuadros de diálogo estándar, como cuadros de mensaje, cuadros de entrada y cuadros de confirmación.

```
import javax.swing.JOptionPane;
```

Ilustración 1 Importaciones

Comentario y declaración de la clase:

/...*/:** Este es un comentario de documentación, que se utiliza para describir la clase o el código que sigue. En este caso, indica el autor del código.

Public class CambioBase: Declaración de una clase pública llamada CambioBase.

```

6  |  /**
7  |  *
8  |  * @author Jaczibeth Cruz
9  |  */
10 |  public class CambioBase {

```

Ilustración 2 Comentarios y declaraciones de clase

Método principal:

Public static void main (String [] args): Este es el punto de entrada del programa. Es un método estático que es llamado por la JVM (Java Virtual Machine) cuando se ejecuta el programa.

```

public static void main(String[] args) {
    String[] bases = {"Binario", "Octal", "Decimal", "Hexadecimal"};
}

```

Ilustración 3 Método Principal "main"

Definición de bases:

String [] bases = {"Binario", "Octal", "Decimal", "Hexadecimal"}: Se define un arreglo de cadenas (String []) llamado bases, que contiene las diferentes bases numéricas que el usuario puede seleccionar: Binario, Octal, Decimal, y Hexadecimal

```
String[] bases = {"Binario", "Octal", "Decimal", "Hexadecimal"};
```

Ilustración 4 Definición de bases

Solicitud de base de entrada:

JOptionPane.showInputDialog (...): Muestra un cuadro de diálogo donde el usuario puede seleccionar la base de entrada.

Null: No se especifica un componente padre, por lo que el cuadro de diálogo aparecerá en el centro de la pantalla.

"Selecciona la base de entrada:": El mensaje que se muestra en el cuadro de diálogo.

"Cambio de Base": El título de la ventana.

JOptionPane.QUESTION_MESSAGE: Tipo de mensaje que muestra un icono de pregunta.

Bases: Las opciones que se presentan al usuario (el arreglo de bases).

Bases [0]: La opción predeterminada (en este caso, "Binario").

```
String[] bases = {"Binario", "Octal", "Decimal", "Hexadecimal"};

// Solicitar al usuario que elija la base de entrada
String baseEntrada = (String) JOptionPane.showInputDialog(
    parentComponent: null,
    message: "Selecciona la base de entrada:",
    title: "Cambio de Base",
    messageType: JOptionPane.QUESTION_MESSAGE,
    icon: null,
    selectionValues: bases,
    bases[0]);
```

Ilustración 5 Solicitud de entrada

Validación de selección de la base de entrada:

If (baseEntrada == null): Verifica si el usuario no seleccionó ninguna base (por ejemplo, si presiona "Cancelar").

JOptionPane.showMessageDialog (null,...): Muestra un mensaje indicando que no se seleccionó ninguna base y luego finaliza el programa con return.

```
// Validar si el usuario seleccionó una opción
if (baseEntrada == null) {
    JOptionPane.showMessageDialog(parentComponent: null, message: "No se seleccionó ninguna base de entrada. Saliendo del programa.",
    return;
}
```

Ilustración 6 Validación de la base de entrada

Solicitud del número en la base seleccionada:

JOptionPane.showInputDialog (null,...): Solicita al usuario que ingrese un número en la base seleccionada. El mensaje se personaliza para incluir la base seleccionada.

```
// Solicitar el número en la base seleccionada
String numero = JOptionPane.showInputDialog( parentComponent: null,
    "Ingresa el número en base " + baseEntrada + ":");
```

Ilustración 7 Solicitud del número en la base seleccionada

Validación del número ingresado:

If (numero == null || numero.isEmpty()): Verifica si el usuario no ingresó ningún número (por ejemplo, si presiona "Cancelar" o deja el campo vacío).
JOptionPane.showMessageDialog (null,...): Muestra un mensaje indicando que no se ingresó ningún número y luego finaliza el programa con return.

```
// Validar si el usuario ingresó un número
if (numero == null || numero.isEmpty()) {
    JOptionPane.showMessageDialog (parentComponent, null, message: "No se ingresó ningún número. Saliendo del programa.");
    return;
}
```

Ilustración 8 Validación del número

Solicitud de la base de salida:

Similar a la solicitud de la base de entrada, este código solicita al usuario que seleccione la base a la que desea convertir el número.

```
// Solicitar al usuario que elija la base de salida
String baseSalida = (String) JOptionPane.showInputDialog(
    parentComponent: null,
    message: "Selecciona la base de salida:",
    title: "Cambio de Base",
    messageType: JOptionPane.QUESTION_MESSAGE,
    icon: null,
    selectionValues: bases,
    bases[0]);
```

Ilustración 9 Solicitud de Salida

Validación de selección de la base de salida:

Similar a la validación de la base de entrada, este código verifica si el usuario seleccionó una base de salida.

```
// Validar si el usuario seleccionó una opción
if (baseSalida == null) {
    JOptionPane.showMessageDialog (parentComponent, null, message: "No se seleccionó ninguna base de salida. Saliendo del programa.");
    return;
}
```

Ilustración 10 Validación de selección de base de salida

Conversión a decimal:

Int numeroDecimal: Declara una variable para almacenar el número en base decimal.

Switch (baseEntrada) {...}: Utiliza una expresión switch para convertir el número de la base seleccionada a decimal.

Integer.parseInt (numero, X): Convierte el número de la base X (donde X puede ser 2, 8, 10, o 16) a decimal.

Default: Si la base de entrada no es válida (no debería ocurrir), se lanza una excepción `IllegalArgumentException`.

Catch (NumberFormatException e): Si el número ingresado no es válido para la base seleccionada (por ejemplo, se ingresa un dígito que no es válido en esa base), se captura la excepción y se muestra un mensaje de error.

```
int numeroDecimal;
try {
    // Convertir el número de la base de entrada a decimal
    numeroDecimal = switch (baseEntrada) {
        case "Binario" -> Integer.parseInt(numero, radix:2);
        case "Octal" -> Integer.parseInt(numero, radix:8);
        case "Decimal" -> Integer.parseInt(numero, radix:10);
        case "Hexadecimal" -> Integer.parseInt(numero, radix:16);
        default -> throw new IllegalArgumentException("Base de entrada no válida.");
    };
} catch (NumberFormatException e) {
    JOptionPane.showMessageDialog(null, "El número ingresado no es válido para la base " + baseEntrada + ".");
    return;
}
```

Ilustración 11 Conversión a decimal

Conversión del número decimal a la base de salida

Switch (baseSalida) {...}: Similar a la conversión anterior, pero ahora convierte el número en base decimal a la base de salida seleccionada.

Integer.toBinaryString (numeroDecimal): Convierte el número decimal a su representación en binario.

Integer.toOctalString (numeroDecimal): Convierte el número decimal a su representación en octal.

Integer.toString (numeroDecimal): Convierte el número decimal a una cadena en base 10.

Integer.toHexString (numeroDecimal).toUpperCase (): Convierte el número decimal a su representación en hexadecimal, y lo convierte a mayúsculas para seguir la convención de representar hexadecimales en mayúsculas.

Default: Similar al caso anterior, lanza una excepción si la base de salida no es válida (lo que tampoco debería ocurrir).

```
// Convertir el número decimal a la base de salida
String resultado = switch (baseSalida) {
    case "Binario" -> Integer.toBinaryString(i:numeroDecimal);
    case "Octal" -> Integer.toOctalString(i:numeroDecimal);
    case "Decimal" -> Integer.toString(i:numeroDecimal);
    case "Hexadecimal" -> Integer.toHexString(i:numeroDecimal).toUpperCase();
    default -> throw new IllegalArgumentException(s:"Base de salida no válida.");
};
```

Ilustración 12 Convertir num_decimal a la base de salida

Mostrar el resultado:

JOptionPane.showMessageDialog (null,...): Muestra un mensaje de diálogo con el resultado de la conversión, indicando al usuario el número original, la base de entrada, y el número convertido en la base de salida.

```
// Mostrar el resultado
JOptionPane.showMessageDialog( parentComponent: null,
    "El número " + numero + " en base " + baseEntrada +
    " es igual a " + resultado + " en base " + baseSalida);
```

Ilustración 13 Resultado

Descripción del programa:

Este programa es una aplicación en Java que permite a los usuarios convertir números entre diferentes bases numéricas (binario, octal, decimal y hexadecimal) a través de una interfaz gráfica basada en cuadros de diálogo. El usuario puede seleccionar la base numérica del número de entrada, ingresar el número en esa base, y luego elegir a cuál base desea convertirlo. El programa realiza las conversiones correspondientes y muestra el resultado final en la base seleccionada.

Ejecución del programa terminado:

Base de entrada:



Ilustración 14 Selección de la base

Ingreso del número deseado a convertir:



Ilustración 15 Número ingresado

Selección de base de salida:



Ilustración 16 Base de salida

Resultado:

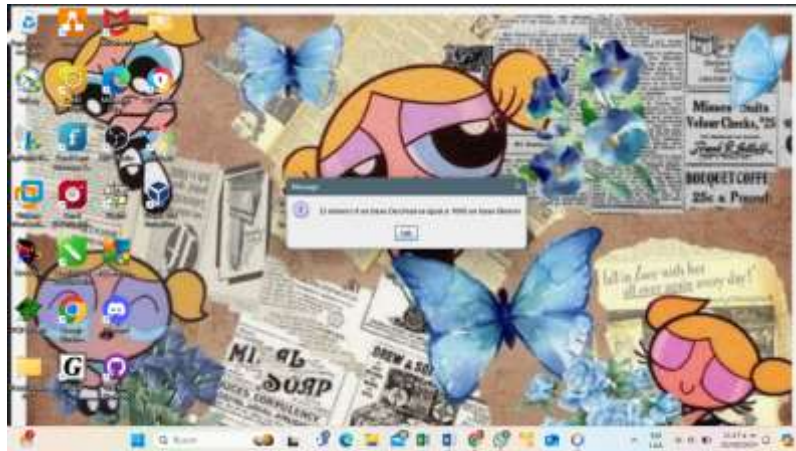


Ilustración 17 Resultado del convertidor

Conclusión

En esta práctica, se ha desarrollado un programa en Java que permite la conversión de números entre diferentes bases numéricas, incluyendo binario, octal, decimal y hexadecimal. Utilizando la biblioteca JOptionPane, el programa proporciona una interfaz gráfica de usuario (GUI) que facilita la interacción del usuario y simplifica el proceso de conversión.

Esta práctica no solo refuerza el conocimiento en la conversión entre bases numéricas, sino que también desarrolla habilidades en el diseño de interfaces gráficas en Java. La capacidad de convertir entre diferentes sistemas numéricos es fundamental para muchas aplicaciones en programación y ciencias de la computación.

El desarrollo de este programa ha demostrado ser una herramienta educativa eficaz para comprender los conceptos de bases numéricas y la interacción del usuario en aplicaciones Java, además de proporcionar una base sólida para futuros proyectos que requieran manipulación de datos numéricos y diseño de interfaces gráficas.

Tabla de ilustraciones

Ilustración 1 Importaciones.....	2
Ilustración 2 Comentarios y declaraciones de clase	3
Ilustración 3 Método Principal"main"	3
Ilustración 4 Definición de bases.....	3
Ilustración 5 Solicitud de entrada	4
Ilustración 6 Validación de la base de entrada	4
Ilustración 7 Solicitud del número en la base seleccionada	4
Ilustración 8 Validación del número	5
Ilustración 9 Solicitud de Salida.....	5
Ilustración 10 Validación de selección de base de salida	5
Ilustración 11 Conversión a decimal.....	6
Ilustración 12 Convertir num_decimal a la base de salida	7
Ilustración 13 Resultado.....	7
Ilustración 14 Selección de la base	8
Ilustración 15 Número ingresado.....	8
Ilustración 16 Base de salida	9
Ilustración 17 Resultado del convertidor	9