

CALCULADORA POO

Documento **main.js** donde se crea la clase **Calculadora** y se encapsulan las propiedades **Pantalla y Botones**, se crea el metodo **constructor**

// Definición de la clase Calculadora

```
class Calculadora {  
  /*Metodo Constructor para inicializar propiedades que ejecuta automaticamente cuando se crea una  
  nueva instancia de la clase*/  
  constructor() {  
    /* busca en el documento html el Elemento que tenga la clase "pantalla"  
    del DOM que representa la pantalla de salida*/  
    this.pantalla = document.querySelector(".pantalla");  
    // Se asignan a la propiedad botones del objeto actual "Nodelist".  
    this.botones = document.querySelectorAll(".btn");  
    // Inicializar eventos para cada botón  
    this.inicializarEventos();  
    // Se agregar evento de teclado "se ingresa por teclado los valores"  
    document.addEventListener("keydown", (event) => {  
      this.manejarTeclaPresionada(event.key);  
    });  
  }  
}
```

1. constructor() { : Esta línea define el constructor de una clase. El constructor es un método especial que se ejecuta automáticamente cuando se crea una nueva instancia de la clase.

2. this.pantalla = document.querySelector(".pantalla"); :: Aquí se busca en el documento HTML un elemento que tenga la clase "pantalla" y se asigna a la propiedad pantalla del objeto actual (this). Esto asume que hay un elemento en el documento HTML con la clase "pantalla", que probablemente represente la pantalla de salida de la calculadora.

3.this.botones = document.querySelectorAll(".btn"); :: Similar al paso anterior, se busca en el documento HTML todos los elementos que tengan la clase "btn" y se asignan a la propiedad botones del objeto actual. Esto representa una colección de botones que el usuario puede presionar para interactuar con la calculadora.

4.this.inicializarEventos(); :: Se llama al método inicializarEventos() para inicializar los eventos para cada botón. Este método establece escuchadores de eventos para los botones, como hacer que cada botón ejecute una determinada acción cuando se presiona

MÉTODO PARA MANEJAR LA LÓGICA POR TECLADO

// Método para manejar la lógica de las teclas presionadas

```
manejarTeclaPresionada(tecla) {
```

*/*evalúa si la tecla presionada es un número (mediante la función isNaN que devuelve true si el valor no es un número), el punto decimal, o alguno de los operadores matemáticos como suma, resta, multiplicación o división. */*

```
if (!isNaN(tecla) || tecla === "." || tecla === "+" || tecla === "-" || tecla === "*" || tecla === "/") {
```

*/*Si la condición del primer if se cumple, entonces se llama al método actualizarPantalla() de la calculadora, pasando como argumento la tecla presionada para actualizar el contenido en pantalla. */*

```
this.actualizarPantalla(tecla);
```

/ En caso de que la tecla presionada sea la tecla "Enter" (que representa la ejecución de la operación matemática), se ejecuta el bloque de código dentro de este else if.*/*

```
} else if (tecla === "Enter") {
```

```
this.ejecutarOperacion();
```

// eliminar un carácter de la pantalla de la calculadora.

```
} else if (tecla === "Backspace") {
```

```
this.borrarPantalla();
```

```
}
```

```
}
```

1. manejarTeclaPresionada(tecla) {: Esta línea define un método llamado manejarTeclaPresionada que recibe un parámetro tecla, que representa la tecla presionada en el teclado.

2. if (!isNaN(tecla) || tecla === "." || tecla === "+" || tecla === "-" || tecla === "*" || tecla === "/") {:

En esta línea se inicia una estructura condicional if, que evalúa si la tecla presionada es un número (mediante la función isNaN que devuelve true si el valor no es un número), el punto decimal, o alguno de los operadores matemáticos como suma, resta, multiplicación o división.

3. this.actualizarPantalla(tecla);: Si la condición del primer if se cumple, entonces se llama al método actualizarPantalla() de la calculadora, pasando como argumento la tecla presionada para actualizar el contenido en pantalla.

4. } else if (tecla === "Enter") {: En caso de que la tecla presionada sea la tecla "Enter" (que representa la ejecución de la operación matemática), se ejecuta el bloque de código dentro de este else if.

5. this.ejecutarOperacion();: Si la condición anterior se cumple, se llama al método ejecutarOperacion() de la calculadora para realizar el cálculo matemático basado en la expresión ingresada en pantalla.

6. } else if (tecla === "Backspace") {: En este bloque, se evalúa si la tecla presionada es "Backspace", que representa borrar un carácter de la pantalla.

7. this.borrarPantalla();: Si la tecla presionada es "Backspace", se llama al método borrarPantalla() para eliminar un carácter de la pantalla de la calculadora.

Este fragmento de código maneja las acciones a realizar en función de la tecla presionada: actualizar la pantalla si es un número, operador o punto decimal, ejecutar la operación si es "Enter" y borrar un carácter si es "Backspace".

MÉTODO PARA ASIGNAR EVENTOS DE CLIC A CADA BOTON

```
// Método para asignar eventos de clic a cada botón "se define el METODO"  
inicializarEventos() {  
  //Se utiliza el método forEach para recorrer cada elemento del NodeList botones.  
  this.botones.forEach(boton => {  
    //Para cada botón, se agrega un escuchador de eventos que estará atento al evento de clic.  
    boton.addEventListener("click", () => {  
      //se llama al método manejarBoton y se pasa como argumento el botón al que se le hizo clic.  
      this.manejarBoton(boton);  
    });  
  });  
}
```

1. // Método para asignar eventos de clic a cada botón: Este comentario indica que se va a definir un método para asignar eventos de clic a cada botón.

2. inicializarEventos() {: Se define el método inicializarEventos en el contexto del objeto actual.

3. this.botones.forEach(boton => {: Se utiliza el método forEach para recorrer cada elemento del NodeList botones.

4. boton.addEventListener("click", () => {: Para cada botón, se agrega un escuchador de eventos que estará atento al evento de clic.

5. this.manejarBoton(boton);: Cuando se hace clic en un botón, se llama al método manejarBoton y se pasa como argumento el botón al que se le hizo clic.

En resumen, este método inicializarEventos recorre todos los botones y les asigna un escuchador de eventos para el evento de clic. Cuando se hace clic en un botón, se llama al método manejarBoton para manejar la interacción con ese botón en particular.

MÉTODO FUNCIÓN PARA BORRAR CARÁCTER DE PANTALLA

// se define el Método función para borrar un carácter de la pantalla

```
borrarPantalla() {
```

```
  /*se realiza una condición con un bloque if. Se verifica si la longitud del texto en la pantalla es igual a 1 o si el texto en la pantalla es igual a "Error!". Si alguna de estas condiciones se cumple, se ejecuta el bloque de código dentro de las llaves.*/
```

```
  if (this.pantalla.textContent.length === 1 || this.pantalla.textContent === "Error!") {
```

```
    /*Si se cumple la condición anterior, se establece el contenido de la pantalla como "0".
```

```
    Esto hace que reinicie la pantalla si solo queda un carácter o si muestra un mensaje de error.*/
```

```
    this.pantalla.textContent = "0";
```

```
  /*Si la condición del if no se cumple, se ejecuta este bloque. Aquí se utiliza
```

```
  el método slice para eliminar el último carácter del texto que está en la pantalla.*/
```

```
  } else {
```

```
    this.pantalla.textContent = this.pantalla.textContent.slice(0, -1);
```

```
  }
```

```
}
```

1. // Método función para borrar un carácter de la pantalla: Este comentario indica que se va a definir un método que funcionará para borrar un carácter de la pantalla.

2. borrarPantalla() {: Se define el método borrarPantalla en el contexto del objeto actual.

3. if (this.pantalla.textContent.length === 1 || this.pantalla.textContent === "Error!") {: Aquí se realiza una condición con un bloque if. Se verifica si la longitud del texto en la pantalla es igual a 1 o si el texto en la pantalla es igual a "Error!". Si alguna de estas condiciones se cumple, se ejecuta el bloque de código dentro de las llaves.

4. this.pantalla.textContent = "0";: Si se cumple la condición anterior, se establece el contenido de la pantalla como "0". Esto probablemente se hace para reiniciar la pantalla si solo queda un carácter o si muestra un mensaje de error.

5. this.pantalla.textContent = this.pantalla.textContent.slice(0, -1);: Si la condición del if no se cumple, se ejecuta este bloque. Aquí se utiliza el método slice para eliminar el último carácter del texto que está en la pantalla.

Este método borrarPantalla borra un carácter de la pantalla, ya sea estableciendo el contenido como "0" si solo queda un carácter o si muestra un mensaje de error, o eliminando el último carácter del texto en la pantalla.

MÉTODO FUNCIÓN PARA EJECUTAR OPERACIÓN MATEMÁTICA

// Método función para ejecutar la operación matemática ingresada en la pantalla

```
ejecutarOperacion() {  
  // Se almacena en la variable por pantalla  
  const operacion = this.pantalla.textContent;  
  // Verifica si el contenido de la pantalla incluye la cadena "/0", lo que indicaría una división por cero.  
  if (operacion.includes("/0")) {  
    // Si se encuentra "/0" en la operación, se muestra un mensaje de error por pantalla.  
    this.pantalla.textContent = "Error! División por cero";  
    // Si no se encuentra "/0", se ejecuta el bloque de código siguiente.  
  } else {  
    // Se inicia un bloque "try", que permite capturar errores durante la ejecución del código dentro de él.  
    try {  
      /* se utiliza la función eval() para evaluar la operación introducida por el usuario y se almacena  
      sea suma, resta, multiplicación o división el resultado en la variable resultado */  
      const resultado = eval(operacion);  
      // Verifica si el resultado es un número finito utilizando el método estático Number.isFinite().  
      if (Number.isFinite(resultado)) {  
        // Si el resultado es finito, se muestra en la pantalla.  
        this.pantalla.textContent = resultado;  
  
        // Si el resultado no es finito, se ejecuta el bloque de código siguiente.  
      } else {  
        // Se muestra un mensaje de error por pantalla indicando que el resultado no es válido.  
        this.pantalla.textContent = "Error! Resultado no válido";  
      }  
  
      // Se inicia un bloque "catch" para manejar cualquier error que ocurra dentro del bloque "try".  
    } catch {  
      // Si ocurre un error durante la evaluación de la operación, se muestra un mensaje de error en la pantalla.  
      this.pantalla.textContent = "Error! Operación inválida";  
    }  
  }  
}
```

1.ejecutarOperacion() {

Esta línea define la función llamada "ejecutarOperacion". Esta función se encargará de ejecutar la operación matemática ingresada en la pantalla.

2.const operacion = this.pantalla.textContent;

Aquí se almacena el contenido de la pantalla en la variable "operacion". Presumiblemente, "this.pantalla" hace referencia al elemento donde se muestra la operación en la interfaz.

3.if (operacion.includes("/0")) {

```
  this.pantalla.textContent = "Error! División por cero";  
} else {  
  // ...  
}
```

En esta sección, se verifica si la operación contiene la cadena "/0", lo que indicaría una división por cero. En caso de encontrarla, se muestra un mensaje de error en la pantalla.

4.try {: Dentro de un bloque "try", se utiliza la función eval() para evaluar la operación introducida por el usuario y se almacena el resultado en la variable "resultado". Luego se verifica si el resultado es un número finito. Si es así, se muestra en la pantalla; de lo contrario, se muestra un mensaje de error. En caso de que ocurra un error durante la evaluación de la operación, se maneja dentro del bloque "catch" mostrando un mensaje de error en la pantalla.

MÉTODO PARA ACTUALIZAR CONTENIDO DE LA PANTALLA

```
// Método para actualizar el contenido de la pantalla
actualizarPantalla(valor) {
  if (this.pantalla.textContent === "0" || this.pantalla.textContent === "Error!") {
    this.pantalla.textContent = valor;
  } else {
    //concatena o agrega el valor especificado al contenido actual que se muestra en la pantalla.
    this.pantalla.textContent += valor;
  }
}
```

1.actualizarPantalla(valor) {

Esta línea define la función llamada "actualizarPantalla" que recibe un parámetro "valor". Esta función se encarga de actualizar el contenido que se muestra en la pantalla.

```
2.if (this.pantalla.textContent === "0" || this.pantalla.textContent === "Error!") {
  this.pantalla.textContent = valor;
} else {
  // ...
}
```

Aquí se evalúa si el contenido actual de la pantalla es igual a "0" o "Error!". En caso de ser así, se reemplaza ese contenido con el nuevo valor proporcionado. Esto significa que si la pantalla muestra "0" o "Error!", al presionar un botón se reemplazará ese contenido con el nuevo valor.

3. this.pantalla.textContent += valor;

En caso de que el contenido actual de la pantalla no sea "0" ni "Error!", se concatena o agrega el valor especificado al contenido actual que se muestra en la pantalla. Esto significa que si ya hay un número en la pantalla y se presiona otro número, este último se agrega al final del número existente.

En resumen, esta función se encarga de manejar la lógica para actualizar el contenido que se muestra en la pantalla, ya sea reemplazando el contenido actual si es "0" o "Error!", o concatenando/agregando valores al contenido existente en la pantalla.

MÉTODO PARA MANEJAR LA LÓGICA DE LOS BOTONES

// Método que maneja la lógica de los botones

```
manejarBoton(boton) {
```

//Se obtiene el texto contenido en el botón que fue presionado y se guarda en la variable botonApretado.

```
const botonApretado = boton.textContent;
```

// Evaluar el ID del botón para determinar la acción

```
if (boton.id === "c") {
```

*/*se establece el contenido de la pantalla como "0"*

reiniciando la calculadora o borrando todo lo mostrado./*

```
this.pantalla.textContent = "0";
```

*/*Se verifica si el ID del botón es "borrar". Si es así, se llama al método borrarPantalla(), que borra un carácter de la pantalla.*/*

```
} else if (boton.id === "borrar") {
```

```
this.borrarPantalla();
```

//se evalúa si el ID del botón es "igual". Si es así, se llama al método ejecutarOperacion()

```
} else if (boton.id === "igual") {
```

```
this.ejecutarOperacion();
```

/ Si ninguna de las condiciones anteriores se cumple, se llama al método actualizarPantalla pasándole como argumento el texto contenido en el botón presionado (botonApretado).*/*

```
} else {
```

```
this.actualizarPantalla(botonApretado);
```

```
}
```

```
}
```

```
}
```

1. // Método que maneja la lógica de los botones: Este comentario indica que se va a definir un método que se encargará de manejar la lógica de los botones.

2. manejarBoton(boton) {: Se define el método manejarBoton que recibe como parámetro un botón.

3. const botonApretado = boton.textContent;: Se obtiene el texto contenido en el botón que fue presionado y se guarda en la variable botonApretado.

4. if (boton.id === "c") {: Se inicia una estructura condicional para evaluar el ID del botón presionado. En este caso, si el ID es "c", se ejecuta el siguiente bloque de código.

5. this.pantalla.textContent = "0";: Si el ID del botón es "c", se establece el contenido de la pantalla como "0", probablemente reiniciando la calculadora o borrando todo lo mostrado.

6. else if (boton.id === "borrar") {: En esta parte se verifica si el ID del botón es "borrar". Si es así, se llama al método borrarPantalla(), que probablemente borra un carácter de la pantalla.

7. else if (boton.id === "igual") {: Aquí se evalúa si el ID del botón es "igual". Si es así, se llama al método ejecutarOperacion(), que posiblemente realiza alguna operación matemática.

8. else { this.actualizarPantalla(botonApretado); }: Si ninguna de las condiciones anteriores se cumple, se llama al método actualizarPantalla pasándole como argumento el texto contenido en el botón presionado (botonApretado).

En resumen, este método manejarBoton se encarga de manejar la lógica de los botones de una calculadora, realizando diferentes acciones dependiendo del ID del botón presionado.


```
// Crear una instancia de la calculadora al cargar el DOM  
document.addEventListener("DOMContentLoaded", function () {  
  const calculadora = new Calculadora();  
});
```

1. document.addEventListener("DOMContentLoaded", function () {

Esta línea agrega un "event listener" que escucha el evento "DOMContentLoaded". Este evento se dispara cuando el HTML ha sido completamente cargado y parseado, sin esperar a que se carguen todos los recursos externos, como imágenes.

2. const calculadora = new Calculadora();

Dentro de la función que se ejecuta cuando se dispara el evento "DOMContentLoaded", se crea una nueva instancia de la clase "Calculadora" y se asigna a la variable "calculadora". Esto significa que cuando la página haya cargado por completo, se creará una nueva calculadora para su uso.

Estas líneas de código aseguran que una vez que el DOM (Document Object Model) esté listo, se creará una instancia de la calculadora para que esté disponible y lista para su uso.

Este ejercicio está orientado a objetos donde se utilizan las características de ENCAPSULAMIENTO y ABSTRACCIÓN, en ENCAPSULAMIENTO tiene la clase Calculadora donde encapsula Pantalla y Botones, los métodos de la clase son: manejarTeclaPresionada, inicializarEventos, borrarPantalla, ejecutarOperacion, actualizarPantalla, manejarBoton.

En Abstracción, la clase Calculadora abstrae el funcionamiento interno de la calculadora los detalles de implementación están encapsulados, y el usuario solo interactúa a través de la interfaz proporcionada por los métodos públicos.

Angel Eduardo Medina Rojas
ADSO 2617510-G2 NOCTURNA