

Explicación - Código de Tartas

Typescript

Introducción

Vamos a ver un ejemplo de polimorfismo y herencia en TypeScript usando tartas. Tenemos varias clases de tartas, cada una con sus ingredientes y su forma de hornearse. Además, podemos interactuar con ellas desde la web.

1 Clase base Tarta

Primero tenemos la clase abstracta Tarta.

Es abstracta porque no podemos crear tartas genéricas, solo clases concretas que hereden de ella.

Declaramos propiedades sabor e imagen y las inicializamos en el constructor usando `this.sabor` y `this.imagen`.

Esto significa que cada instancia de tarta tendrá su sabor y su imagen.

Tenemos un método abstracto `hornear()` que todas las tartas concretas deben implementar. Esto nos obliga a definir cómo se hornea cada tarta.

También hay un método protegido `ingredientesEspecificos()`, accesible solo desde la clase base y las clases hijas, que devuelve los ingredientes únicos de cada tarta.

El método `ingredientes()` combina los ingredientes comunes con los específicos usando `spread operator`

El getter `nombre` nos permite acceder a `this.nombre` como si fuera una propiedad, pero en realidad se calcula al momento de usarlo.

2 Clases concretas

Ahora tenemos las clases que heredan de Tarta, usando extends: TartaChocolate, TartaFresa, TartaQueso, TartaLimon y TartaZanahoria.

Cada clase concreta:

- llama al constructor de la clase base con super(sabor, imagen)
- implementa hornear()
- define ingredientesEspecificos()

Por ejemplo, TartaChocolate devuelve Cacao y Mantequilla como ingredientes específicos. Y en hornear() indicamos cómo hornear la tarta y usamos this.nombre para mostrar su nombre.

3 Estado inicial

El array tartas puede empezar vacío y no veríamos nada en la web. Podemos añadir más usando el botón Añadir.

Hemos incluido sólo la de Limón para que veamos lo que muestra, y entendamos mejor cada parte del código. El resto que añadamos será una actualización dinámica.

Conseguimos ver la Tarta de Limón gracias al render,

Desde el código usando **new TartaChocolate()**, o similares dentro del array, podemos añadir más a la web sin clicar al botón Añadir.

4 Referencias al DOM

Cogemos los elementos HTML donde vamos a mostrar la lista de tartas y los resultados con getElementById.

5 Función renderLista()

Esta función genera el HTML de cada tarta y lo mete en la página.

Cada tarta se muestra como una tarjeta (<article>).

Añadimos un button por cada tarta para poder hornearla individualmente.

El data-index nos ayuda a saber cuál tarta hornear.

Cuando pulsamos el botón, llamamos a `tartas[idx].hornear()` y mostramos el resultado en la lista de resultados.

6 Función `pushResultado(texto)`

Crea un `` con el mensaje de horneado y lo añade al inicio de la lista de resultados.

7 Botones de acción

Botón Añadir tarta:

- Lee el tipo de tarta seleccionado del `<select>`.
- Crea una instancia con `crearTarta(tipo)`.
- La añade al array `tartas`.
- Vuelve a renderizar la lista.

Botón Hornear todas:

- Recorre todas las tartas y llama a `hornear()` de cada una, mostrando el resultado.
-

8 Función `crearTarta(tipo)`

Esta función devuelve una nueva instancia de la tarta según el tipo: chocolate, fresa, queso, limón o zanahoria.

Si no coincide, devuelve chocolate por defecto.

9 Primer render

Al cargar la web llamamos a `renderLista()`.

Como el array está vacío, no se muestra ninguna tarta hasta que añadamos alguna.