

Avance 3

Consideraciones Proyecto semana 3

Duración: 2 horas

(Colocar instrucciones y recomendaciones del Avance 3 para el proyecto final, incluye hasta el tema 6))

Avance del proyecto semanal.

En este avance vas a crear las clases de la aplicación.

- En el archivo llamado Dato.js crea una clase padre para el manejo de datos que se llame Dato
- Agrega un constructor al cual le vas a pasar dos atributos: descripción y valor
- Dentro del cuerpo del constructor define los atributos `_descripción` y `_valor` y asígnale el valor de los parámetros que se están recibiendo
- Agrega los métodos **get** y **set** para el atributo descripción:
 - En el **get** retorna este valor del atributo **descripción**.
 - En el **set**, haz que se reciba el parámetro **descripción** y modifica el atributo de **descripción** con el valor que se está recibiendo.
- Crea los métodos set y get para el atributo valor de la misma manera que se realizó para el atributo **descripción** con sus respectivos valores.
- Recuerda que es importante encapsular los datos

Define ahora las clases hijas:

- Utilizando el archivo **Ingreso.js**, define la clase **Ingreso**:
 - Define la clase ingreso, que extiende de la clase padre **Dato**.
 - Define la variable estática `contadorIngresos` e iguálala a 0.
 - Crea el constructor de la clase en el que recibas los parámetros **descripción y valor**.
 - En el cuerpo del constructor, invoca al constructor de la clase padre, la cual debe recibir los parámetros **descripción y valor**.
 - Define el atributo `_id` y para asignarle un valor, utiliza la variable estática, para hacerlo, realiza un preincremento a la variable estática de la clase **Ingresos**.
 - Crea el método **get id**, el cual va a regresar el valor de `this._id`, no agregues el método set porque este valor no deberá ser modificado.

Avance 3

- Utilizando el archivo **Egreso.js**, crea la clase Egreso, que es hija de la clase Dato:
 - Define una variable estática contadorEgresos inicializada en 0.
 - Crea el constructor que recibe los valores de **descripcion y valor**.
 - Inicializa el objeto de la clase **padre**, el cual recibe el valor de descripcion y valor.
 - Define el atributo **_id** y, para asignarle un valor, utiliza la variable estática. Para hacerlo, realiza un preincremento a la variable estática de la clase Egresos para asegurar que inicie en 1.
 - Crea el método **get id**, el cual va a regresar el valor de this._id, no agregues el método set, pues este valor no deberá ser modificado.
- Crea los arreglos que van a manejar los ingresos y los egresos de la aplicación (elimina los arreglos que se habían creado para poder realizar la prueba de las funciones). Para ello, define la variable ingresos y egresos fuera de toda función (al principio del archivo de ser posible) en el archivo app.js de la siguiente manera:
 - El objeto **ingresos** es un arreglo que instancia a la clase Ingreso y recibe como parámetros, a manera de ejemplo, los valores ('Salario', 20000) y ('Venta auto', 50000).
 - La constante **egresos** es un arreglo que crea dos objetos del tipo Egreso y recibe como parámetros, a manera de ejemplo, los valores ('Renta', 4000) y ('Ropa', 800).
- Modifica la clase **totalIngresos** y recorre el arreglo **ingresos** con un ciclo for of. Asigna a la variable **totalIngresos** la suma del valor del elemento **ingreso**.
- Modifica la clase totalEgresos y recorre el arreglo egresos con un ciclo for of. Asigna a la variable **totalEgreso** la suma del valor del elemento **ingreso**.
- Prueba nuevamente y valida que la salida de la instrucción **cargarCabecero()** sea la siguiente:

```

> cargarCabecero();
$2,300.00 app.js:48
36.11 % app.js:49
$3,600.00 app.js:50
$1,300.00 app.js:51
< undefined
> |
  
```

Avance 3

Consideraciones

Maestro:

- Conforme a la planeación de tu clase, de ser posible, refuerza el concepto de clases y resalta la diferencia entre las funciones y las clases.

Alumno:

- Asegúrate de comprender la diferencia entre clase y función y sigue los pequeños tips de nomenclatura indicados en la literatura. El test en la consola te puede ayudar para ver los resultados.

La obra presentada es propiedad de ENSEÑANZA E INVESTIGACIÓN SUPERIOR A.C. (UNIVERSIDAD TECMILENIO), protegida por la Ley Federal de Derecho de Autor; la alteración o deformación de una obra, así como su reproducción, exhibición o ejecución pública sin el consentimiento de su autor y titular de los derechos correspondientes es constitutivo de un delito tipificado en la Ley Federal de Derechos de Autor, así como en las Leyes Internacionales de Derecho de Autor.

El uso de imágenes, fragmentos de videos, fragmentos de eventos culturales, programas y demás material que sea objeto de protección de los derechos de autor, es exclusivamente para fines educativos e informativos, y cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por UNIVERSIDAD TECMILENIO.

Queda prohibido copiar, reproducir, distribuir, publicar, transmitir, difundir, o en cualquier modo explotar cualquier parte de esta obra sin la autorización previa por escrito de UNIVERSIDAD TECMILENIO. Sin embargo, usted podrá bajar material a su computadora personal para uso exclusivamente personal o educativo y no comercial limitado a una copia por página. No se podrá remover o alterar de la copia ninguna leyenda de Derechos de Autor o la que manifieste la autoría del material.