Prog. Orientada a Objetos

CFL
Programador full-stack

Ejercicios para Practicar - Parte 1

Evaluación del Módulo

- Va a consistir en tres partes
- Pensar en Objetos
  - A partir de una serie de requerimientos, implementar un sistema en TypeScript → prestar atención a encarar bien la solución, no hace falta preocuparse por detalles de implementación
- Documentación
  - A partir de un código en TypeScript, plantear el diagrama de clases → prestar atención a las relaciones entre las clases
- Preguntas Teóricas
  - Van a ser básicas pero con la idea de que puedan desarrollar y demostrar que conocen la teoría

Prog. Orientada a Objetos

CFL
Programador
full-stack

Ejercicios

Parte 1 - Diseño de Sistemas

- Implementar un sistema de control de stock para un kiosco. Dicho sistema debe tener un listado de los elementos disponibles para vender, y un listado de los elementos vendidos
  - Tener en cuenta que cada ítem o elemento tiene un costo asociado
  - El sistema debe poder cargar de un arreglo los items a vender
  - El sistema debe poder imprimir en consola los ítems vendidos

Parte 2 - Documentación

- A partir del código TypeScript, plantear el diagrama de clases asociado
- Tener en cuenta las relaciones entre clases
  - Herencia
  - Implementación
  - Composición
- Tener en cuenta los modificadores de acceso en las variables y métodos
  - + → public
  - - → private
  - # → protected
- Partir del código disponible en el repo

Parte 3 - Preguntas Teóricas

- Responder las preguntas en un TXT y subirlo a GitHub
  - ¿Cuál es el beneficio de usar un lenguaje con tipos (Typescript)?
  - ¿A qué se le llama variable interna? ¿Por qué internas?
  - Explicar la diferencia entre composición y herencia
  - Explicar el mecanismo que provee TypeScript para manejar casos en donde los parámetros que le llegan a un método son inválidos