

Instituto Tecnológico “Sudamericano”

Programación Orientada a Objetos Nave Espacial

Nombre: Ariel Samaniego

Carrera: Desarrollo en Software

Ciclo: Segundo Ciclo

Programación Orientada a Objetos Nave Espacial

En el desarrollo del proyecto se enfocó principalmente en la forma de conseguir el resultado deseado con la nave espacial y sus funcionalidades.

El ejercicio consiste en desarrollar un simulador de exploración espacial en consola utilizando TypeScript, donde el jugador gestiona una nave espacial explorando planetas y enfrentando eventos aleatorios. Se modelan los datos con interfaces para la nave y clases para planetas y eventos, empleando tipos primitivos y enums para categorizar propiedades. Las funcionalidades incluyen exploración, recolección de recursos y manejo de eventos, utilizando tipos de unión y genéricos para flexibilizar las funciones. Se aborda la captura de entradas del usuario y la simulación del tiempo y la distancia durante el viaje, con manejo de errores. La organización del código se realiza mediante módulos y espacios de nombres para mejorar la mantenibilidad y escalabilidad del proyecto. Para ejecutar el simulador, se requiere Node.js y TypeScript para compilar y ejecutar el código resultante.

Como resultado me dio los siguientes valores:

```
Planeta { nombre: 'Planeta 1', tipoRecurso: 0, peligro: false }
{
  nombre: 'Tormenta de asteroides',
  afectaSalud: true,
  afectaCarga: false
}
{ salud: 100, capacidadCarga: 50, velocidad: 10 }
```

El código se divide en dos partes que es index.ts y Funcionalidades.ts las cuales se complementa para dar el resultado deseado en el trabajo.

Funcionalidades.ts

```
Funciones.ts > NaveEspacial > capacidadCarga
1  export enum TipoRecurso {
2    Agua,
3    Mineral,
4    Vegetal
5  }
6
7  export class Planeta {
8    constructor(public nombre: string, public tipoRecurso: TipoRecurso, public peligro: boolean) {}
9  }
10
11 export interface Evento {
12   nombre: string;
13   afectaSalud: boolean;
14   afectaCarga: boolean;
15 }
16
17 export interface NaveEspacial {
18   salud: number;
19   capacidadCarga: number;
20   velocidad: number;
21 }
22
```

Index.ts

```
1 import { Planeta, TipoRecurso } from './Funciones';
2 import { Evento, NaveEspacial } from './Funciones';
3
4 function imprimirInfoPlaneta(planeta: Planeta): void {
5     console.log('Planeta { nombre: `${planeta.nombre}`, tipoRecurso: `${planeta.tipoRecurso}`, peligro: `${planeta.peligro} }');
6 }
7
8 function imprimirInfoEvento(evento: Evento): void {
9     console.log(evento);
10 }
11
12 function imprimirInfoNave(nave: NaveEspacial): void {
13     console.log(nave);
14 }
15
16 const planeta = new Planeta('Planeta 1', TipoRecurso.Agua, false);
17 const evento: Evento = {
18     nombre: 'Tormenta de asteroides',
19     afectaSalud: true,
20     afectaCarga: false
21 };
22 const nave: NaveEspacial = {
23     salud: 100,
24     capacidadCarga: 50,
25     velocidad: 10
26 };
27
28 imprimirInfoPlaneta(planeta);
29 imprimirInfoEvento(evento);
30 imprimirInfoNave(nave);
31
```

En conclusión, estas imágenes demuestran como el código pudo ejecutarse sin problemas y dando el resultado de la nave espacial reaccionando a los planetas.