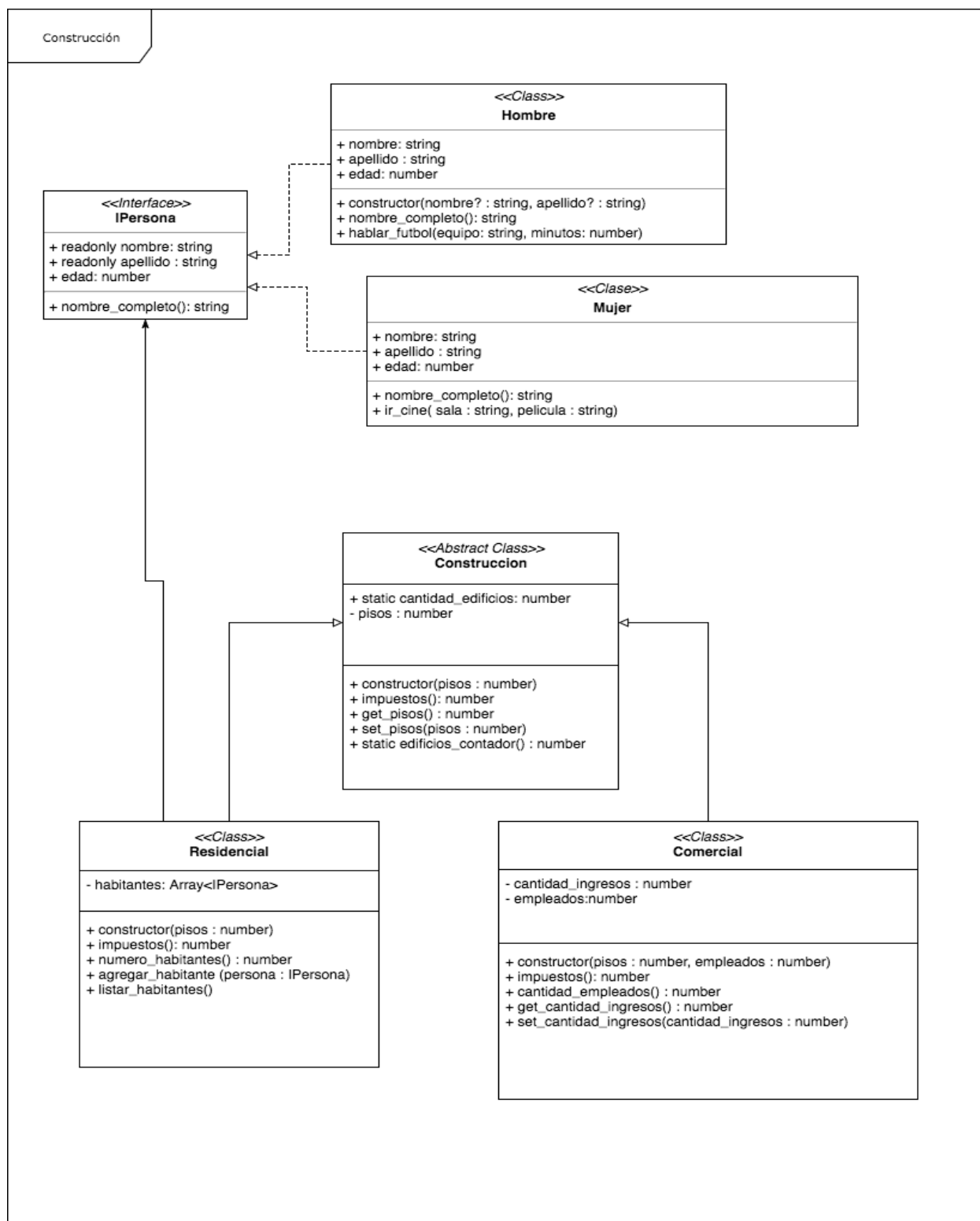




Taller de Programación Orientada a Objetos con TypeScript

Analice y desarrolle el siguiente diagrama de clases en el lenguaje TypeScript:





UPBtec



Universidad
Pontificia
Bolivariana

Taller de Programación Orientada a Objetos con TypeScript

Consideraciones generales

Tenga en cuenta lo siguiente:

1. Tenga en cuenta los modificadores de acceso que se presentan para cada una de las propiedades y métodos, donde menos (-) significa privado, numeral (#) significa protegido y más (+) significa publico.
2. Tenga en cuenta los tipos para cada propiedad.
3. Tenga en cuenta los tipos de datos para los parámetros de entrada.
4. Tenga en cuenta los tipos de datos devueltos por los métodos. Recuerde que los métodos que no tienen marcado el tipo de retorno, por defecto son void.
5. Antes del nombre de cada ítem del diagrama se encuentra su marcado entre los siguientes símbolos "<<tipo de ítem>>" lo cual le permite identificar cómo deberá ser creado.
6. Las clases Hombre y Mujer implementan la interface IPersona.
7. Las clases Residencial y Comercial se heredan de la clase Construcccion.
8. Entre la clase Residencial y la interfaz IPersona, se encuentra una relación, puesto que en la clase Residencial se creará un array que contendrá objetos de tipo IPersona.
9. Recuerde que cuando encuentre el símbolo de cierre de pregunta (?) en los parámetros de un método, significa que estos son opcionales y deberá validar que estos sean diferentes de undefined para poder ser asignados.
10. Algunos métodos o propiedades están marcadas como **static**. Tenga en cuenta dicha característica para que su desarrollo esté correcto.

Consideraciones para la clase Hombre

1. El método nombre_completo retorna el nombre y apellido de la persona.
2. El método hablar_futbol mostrará por consola el siguiente mensaje:
 - a. Hablemos sobre el: <<equipo>> <<minutos>> minutos, pues está jugando como los dioses.Recuerde que <<equipo>> y <<minutos>> son parámetros que llegan al método.

Consideraciones para la clase Mujer

1. El método nombre_completo retorna el nombre y apellido de la persona.
2. El método ir_cine mostrará por consola el siguiente mensaje:
 - a. Viendo <<pelicula>> en <<sala>>.Recuerde que <<pelicula>> y <<sala>> son parámetros que llegan al método.

Consideraciones para la clase Construcccion



UPBtec



Universidad
Pontificia
Bolivariana

Taller de Programación Orientada a Objetos con TypeScript

1. En el método **constructor**, además de asignar el valor de pisos que llega por parámetro, se debe sumar 1 al valor que ya tenga el parámetro estático **cantidad_edificios**. Esto permitirá que cada vez que se cree una instancia de los hijos de esta clase, se cree un contador de edificios que pertenecen a esta construcción.
2. El método **impuestos** retorna el valor por defecto de la multiplicación de **pisos** por el valor de 3000000.
3. El método estático de **edificios_contador** retorna el valor del atributo estático **cantidad_edificios**.

Consideraciones para la clase Residencial

1. El atributo **habitantes** deberá ser inicializado, por favor investigue cómo hacerlo.
2. El método **impuestos** deberá ser sobrescrito para retornar el resultado de la multiplicación de **pisos** por el valor de 1500000.
3. El método **numero_habitantes** retornara el tamaño de la propiedad **habitantes**.
4. El método **agregar_habitante** deberá agregar en la última posición de la propiedad **habitantes** la persona que llega por parámetro.
5. El método de **listar_habitantes** deberá recorrer la propiedad habitantes y por cada persona mostrar el nombre completo en consola. Además deberá validar el tipo de instancia de cada objeto recorrido y dependiendo de la instancia llamar al método que lo caracteriza. Por ejemplo:
 - a. Si el objeto es de tipo **Hombre**, llamar al método **hablar_futbol** y pasar algunos parámetros por defecto.
 - b. Si el objeto es de tipo **Mujer**, llamar al método **ir_cine** y pasar algunos parámetros por defecto.

Consideraciones para la clase Comercial

1. En el método **constructor**, se debe asignar el valor de **pisos** y **empleados** a sus propiedades correspondientes.
2. El método **impuestos** no se debe sobrescribir, pues se utilizara la misma funcionalidad de la clase padre.
3. El método **cantidad_empleados** retorna el valor de la propiedad **empleados**.