Programa Aspira

Paradigmas de Programação com TypeScript

Lista de Exercícios 1

1. Crie um projeto TypeScript organizado em diretórios src e public, contendo também um arquivo tsconfig.json configurado para compilar tendo como alvo o ECMAScript 2015. Dentro de src, crie uma pasta denominada entities (todas as classes criadas nos próximos exercícios deverão estar dentro desta pasta). As outras configurações do projeto em relação ao TypeScript e à extensão Live Server devem seguir o mesmo modelo utilizado como exemplo durante as aulas.
2. Crie um arquivo com uma classe denominada Person, a qual deve ser a exportação padrão deste arquivo, contendo as propriedades name do tipo String, birth do tipo Date e gender do tipo Gender. Implemente também o construtor da classe, o qual deve receber parâmetros equivalentes a cada propriedade a ser preenchida em uma instância desta classe. Crie também (no mesmo arquivo) um **enumerador** denominado Gender, com os membros Male = 'm' e Female = 'f'.
3. Crie um arquivo com uma classe denominada Document, a qual deve ser a exportação padrão deste arquivo, contendo as propriedades title do tipo String, subtitle do tipo String, publishedAt dos tipos Date ou Number, author do tipo Person. Implemente também o construtor da classe, o qual deve receber parâmetros equivalentes a cada propriedade, a serem preenchidas em uma instância desta classe. Não se esqueça de importar a classe Person neste arquivo, para que ela possa ser utilizada para tipagem da propriedade author.
4. Crie um arquivo com uma classe denominada Book, a qual deve ser a exportação padrão deste arquivo, contendo as propriedades isbn do tipo Number, edition do tipo Number e volume do tipo Number. Garanta que a classe Book herde as outras propriedades a partir da classe Document. Implemente também o construtor da classe, o qual deve receber parâmetros equivalentes a cada propriedade da classe Document e desta própria classe, a serem preenchidas em uma instância desta classe. Lembre-se que o construtor de uma classe filha deve chamar o construtor da classe pai para enviar os parâmetros corretos a serem preenchidos como propriedades herdadas da classe pai. Não se esqueça de importar a

classe Document neste arquivo, para que ela possa ser utilizada na herança.

5. Crie um arquivo com uma classe denominada Periodical, a qual deve ser a exportação padrão deste arquivo, contendo as propriedades issn do tipo Number, volume do tipo Number e issue do tipo Number. Garanta que a classe Periodical herde as outras propriedades a partir da classe Document. Implemente também o construtor da classe, o qual deve receber parâmetros equivalentes a cada propriedade da classe Document e desta própria classe, a serem preenchidas em uma instância desta classe. Lembre-se que o construtor de uma classe filha deve chamar o construtor da classe pai para enviar os parâmetros corretos a serem preenchidos como propriedades herdadas da classe pai. Não se esqueça de importar a classe Document neste arquivo, para que ela possa ser utilizada na herança.
6. Crie um arquivo tests.ts na pasta src, o qual deve importar as classes Person, Book e Periodical. Então, crie várias instâncias de pessoas, livros e periódicos, chamando corretamente seus construtores e preenchendo com diversos dados simulando possíveis dados reais que seriam preenchidos para estas entidades. Garanta que toda a tipagem correta esteja sendo atendida para todas as propriedades preenchidas. Por fim, faça console.log de cada objeto instanciado, para poder apreciar os resultados no console do navegador.
7. Crie uma página index.html na pasta public, a qual deve carregar o script compilado a partir do arquivo tests.ts, para poder visualizar no navegador os resultados da execução daquele arquivo.

new Date('2002-09-17T00:00:00')