FullStack

Exercícios da Semana

[M1S4] Ex 1 – PadariaTechMegaPlus

O seu Manoel gostou do último aplicativo e agora quer inovar. Ele quer que você construa a aplicação utilizando orientação a objetos.

Construa uma classe chamada **CaixaRegistradora**. Defina um atributo que armazene vários produtos no estoque contendo os seguintes campos:

[{

codigoBarra:number;

preco:number;

nome:string;

}]

Defina um método no qual o seu Manoel consiga adicionar novos produtos no estoque. Esse método recebe os mesmos campos do produto, e todos eles são obrigatórios.

Defina um método que inicie o atendimento ao cliente, você deve passar o nome do cliente para iniciar.

Defina um método que receba *codigoBarra:number*; e *quantidade:number*; para o seu Manoel conseguir adicionar itens na caixa registradora. **Importante**: A aplicação só poderá adicionar itens na caixa, se o código de barra dele existir.

Defina um método que você consiga verificar o valor total da conta do cliente.

Defina um método **fecharConta**, no qual você passa o dinheiro e ele calcula o troco e zera a caixa registradora.

[M1S4] Ex 2 – Banco Poupancito

Um banco contratou você para criar uma classe que opere as contas do banco utilizando orientação a objetos. Crie uma classe chamada **Conta**.

A classe conta recebe dois atributos: saldo e senha(privado), e métodos depósito e retirada. O método depósito adiciona valor ao saldo (o usuário deve passar a senha, e ela deve ser igual a senha determinada no atributo), e o depósito retira valor do saldo.

Por fim, instancie um objeto chamado contaCorrente e teste as operações.

[M1S4] Ex 3 – Banco Poupancito 2

Agora o banco quer criar uma nova classe chamada **ContaPoupanca**. A conta poupança tem todos os métodos e atributos da classe **Conta** (herança), porém essa nova classe deve ter um novo método chamado **atualizaJuros**. Esse método deve pegar o valor do saldo e adicionar um ganho baseado em um índice de 0.7% todas as vezes que for chamado.

[M1S4] Ex 4 – Banco Poupancito 3

Agora o banco inventou de criar um novo tipo de poupança, chamada **poupancaPremium**, que contém uma taxa de juros melhor. Crie uma classe que herde atributos e métodos da classe poupança criada anteriormente e modifique o método **atualizaJuros** (polimorfismo), aumentando a taxa para 1.2% toda a vez que o método for chamado.

[M1S4] Ex 5 – Data Person

Crie uma classe chamada **Person** que receba atributos nome, idade e altura, e um método **apresentar** que imprima: 'Olá me chamo ${nome}, tenho ${idade} anos e tenho ${altura} de altura'.

Crie uma classe que herde atributos e métodos da classe **Person**, adicione o método **profissao**, e reescreva o método apresentar para imprimir 'Olá me chamo ${nome}, tenho ${idade} anos, tenho ${altura} de altura e sou ${profissao}'.

[M1S4] Ex 6 – PadariaTechMegaPlusBlaster

O seu Manoel agora quer mais uma inovação. Ele quer poder guardar itens no estoque e que o item seja removido automaticamente do estoque quando alguém fizer alguma compra. Para isso você pode colocar um novo item em cada objeto chamado **quantidade**. Para salvar os dados, você pode utilizar *localStorage*.