INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA PIAUÍ	**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ** **Curso: ADS** **Disciplina: Programação Orientada a Objetos** **Professor: Ely**

Exercício 06

1. As classes **Carro, Veiculo** e **CarroEletrico** são bem semelhantes. Refatore as classes para que os atributos duplicados não sejam mais necessários.

public class Veiculo {
String placa;
int ano;
}
	public class Carro {
	String placa;
	int ano;
	String modelo;
	}

public class CarroEletrico {
String placa;
int ano;
String modelo;
int autonomiaBateria
}	

2. Crie uma classe Calculadora com:
   a. Dois tributos privados chamados _op1 e _op2;
   b. Crie um construtor que inicializa os atributos;
   c. Crie um método chamado adicionar que retorna a soma dos dois atributos;
   d. Teste a classe.

3. Crie uma classe chamada CalculadoraCientifica que herda da classe Calculadora do exercício passado e:
   a. Implemente um método chamado exponenciar que retorne o _op1 elevado ao _op2;
   b. Teste a classe;
   c. Foi necessária alguma modificação em Calculadora para o acesso aos atributos?

4. Implemente na classe Banco o método renderJuros(numero: String): number, onde:
   a. É passado por parâmetro o número de uma poupança e feita uma consulta para ver se a conta existe. Note que a consulta não se altera sendo Conta ou Poupança;
   b. Caso a poupança seja encontrada, teste se realmente se trata de uma poupança com o operador instanceof, desconsidere a operação caso contrário;
   c. Caso seja, faça um cast e invoque o método renderJuros da própria instância encontrada;
   d. Teste o método da classe Banco passando tanto um número de poupança como de conta passados inseridos anteriormente.

5. Suponha duas classes Produto e ProdutoPerecivel. Produto tem atributos privados _id, _descricao, _quantidade e _valor. Já ProdutoPerecivel tem as mesmas características de Produto, porém possui a mais um atributo chamado _dataValidade (https://www.javatpoint.com/typescript-date-object).

   Produto possui dois métodos: repor e darBaixa, onde ambos somam e subtraem uma quatidade passada por parâmetro do atributo quantidade. Além disso, um produto perecível possui um método que diz se um produto está válido ou não comparando sua data de validade com a data atual.

   Dessa forma implemente:

   a. Usando herança, as duas classes Produto e ProdutoPerecivel;
   b. Uma classe chamada Estoque que possui um atributo privado que é um array de produtos (Produto ou ProdutoPerecivel);
   c. Métodos para Inserir, consultar, excluir produtos na classe estoque;
   d. Crie validações para não deixar serem incluídos produtos com mesmo id ou mesmo nome;
   e. Os métodos repor e darBaixa, onde após uma consulta são chamados os métodos da classe produto para finalmente alterar a quantidade;
   f. Um método que liste todos os produtos perecíveis vencidos.