



Instituto Federal de Educação, Ciência e Tecnologia do Piauí
Campus Teresina Zona Sul
Licenciatura em Informática
Disciplina: Programação Orientada a Objetos

Exercício Revisão

Questão 1

Crie uma classe **Pessoa**, que contenha **nome** e **idade** como atributos encapsulados e três construtores, o primeiro não recebe nenhum valor e cria uma pessoa com o nome “Fulano” e idade 0(zero), o segundo recebe o nome da pessoa e cria uma pessoa com idade 0(zero) e o último recebe o nome e idade da pessoa. Após isso, crie uma classe **Aplicativo** que contenha três objetos da classe Pessoa, cada um criado com um construtor diferente. Por fim, exiba o nome e a idade de cada pessoa criada.

Questão 2

Altere a classe pessoa da questão anterior, colocando nela o método **exibir**. Esse método deve retornar uma *String* com o nome e a idade da pessoa, por exemplo, “João 25”. Com isso, no momento de exibir as informações da pessoa deve-se usar o método exibir, como no código abaixo.

```
System.out.println(fulano.exibir());
```

Questão 3

Na programação, toda a evolução ocorreu para o reaproveitamento de código. Inicialmente, criaram-se os métodos, depois as classes e objetos, por fim, os serviços. Sabendo disso, altere a classe **Pessoa** para que o código da sobrecarga de construtores seja reaproveitado. Para isso, você deve ter em mente que a palavra **this** funciona como um executor do construtor da classe em questão, como no exemplo abaixo, na qual o construtor da classe conta que recebe dois parâmetros é executado nos outros construtores, através da palavra **this**. Mais explicações sobre o **this** no encadeamento de construtores [aqui](#).

```
public class Conta {  
    private int numero;  
    private double saldo;  
  
    public Conta(){  
        this(0,0);  
    }  
  
    public Conta(int numero){  
        this(numero, 0);  
    }  
}
```

```

public Conta(double saldo, int numero){
    this.saldo = saldo;
    this.numero = numero;
}
...
}

```

Questão 4

Crie uma classe **Lampada**, que simula uma lâmpada em um sistema. Nesta classe, os objetos do tipo **Lampada** deve conter um estado que indique a situação da lâmpada (acesa ou apagada). Por fim, crie uma classe **Aplicativo** que contenha um objeto da classe **Lampada**, coloque o seu estado e exiba o valor atual desse estado. Observe o diagrama de classe abaixo.

Lampada
- estado: boolean
+ Lampada() + acender(): void + apagar(): void + getEstado():boolean

Questão 5

Crie uma classe **Aluno** que é formada por 2 atributos, um para armazenar o nome e o outro para armazenar a média. Além disso, o aluno deve ter um construtor que recebe os valores de seus atributos e um método que exibe a situação do aluno (aprovado, reprovado, prova final). Após isso, crie uma classe **Aplicativo** que contenha um objeto da classe Aluno, coloque o nome do aluno e sua média e exiba a situação do aluno. Observe o diagrama abaixo para construir a classe Aluno.

Aluno
- nome: String - media: double
+ Aluno(nome,media) + setMedia(media): void + getMedia(): double + getSituacao(): String

Questão 6

(Obs: na próxima versão, fazer como uma conta que a empresa deposita dinheiro e saca e o sistema controlar a emissão e consumo de carbono, como já ocorre)

Uma das novidades no combate do aquecimento global foi a criação do Banco de Carbono. Em resumo, esse banco funciona da seguinte forma, empresas que retiram/consomem carbono da

atmosfera, como plantações de mata nativa e florestas, podem vender carbono para empresas que emitem carbono, como fábricas automotiva e frotas de automóveis. Sabendo disso, crie duas classes, uma para as empresas Consumidoras e outra para as Emissoras.

As empresas consumidoras são caracterizadas pela taxa mensal de consumo de carbono, preço atual que ela vende o carbono, crédito de carbono (total consumido) e o saldo em conta (dinheiro na conta da empresa). Além disso, elas devem possuir um método consumir carbono (consumirCarbono), que simula o consumo mensal de carbono, e outro vender crédito de carbono (venderCarbono), que simula a venda de créditos de carbono. Dessa forma, o método de consumir carbono deve aumentar o total de carbono consumido pela empresa (seu crédito de carbono). Já, o método de vender crédito de carbono deve decrementar o crédito de carbono da empresa e aumenta o seu saldo.

As empresas emissoras, são caracterizadas pela taxa mensal de emissão de carbono, lucro mensal da empresa, total de carbono emitido e saldo na conta da empresa (dinheiro na conta da empresa). Seus métodos são o produzir, que simula o ganho mensal da empresa, e comprar crédito de carbono, que simula uma compra de créditos de carbono. Seguindo esta lógica, o método **produzir** deve aumentando o total de carbono emitido pela empresa e o seu saldo em conta. Por sua vez, o método comprar crédito de carbono deve decrementar o total de carbono emitido pela empresa e decrementar o seu saldo em conta (dinheiro em conta).

Emissora
- taxaDeEmissao: double - lucroMensal: double - carbonosEmitidos: double - saldo: double
+ Emissora(taxaDeEmissao, lucroMensal) + produzir(): void + comprarCreditoCarbono(quantidade, valor): void

Consumidora
- taxaDeConsumoMensal: double - precoDoCreditoDeCarbono: double - creditosDeCarbono: double - saldo: double
+ Consumidora(taxaDeConsumoMensal, precoDoCreditoDeCarbono) + consumir(): void + venderCreditoDeCarbono(quantidade): void + getPrecoCreditoCarbono(): double

Para simplificar, imagine que o taxa de consumo de carbono de cada empresa já é em créditos de carbono. Assim, se uma empresa tem uma taxa de consumo de 30, que dizer que ela gera 30 créditos de carbono por mês.

Agora simule uma transação de venda de carbono. Primeiro você deve criar os objetos simulando empresas consumidoras e emissoras. Depois, você deve fazer a Emissora produzir e a Consumidora consumir. Por fim, a empresa Emissora deve comprar carbono da empresa Consumidora.

Questão 7

Altere a questão anterior para que, no método de comprar crédito de carbono, o método receba a quantidade de crédito que será comprado e empresa consumidora que irá vender o crédito. O novo diagrama da classe **Emissora** pode ser visto abaixo.

Emissora
- taxaDeEmissao: double - lucroMensal: double - carbonosEmitidos: double - saldo: double
+ Emissora(taxaDeEmissao, lucroMensal) + produzir(): void + comprarCarbono(quantidade, Consumidora): void