

Exercícios: Sobrecarga

Você trabalha em uma empresa especialista em máquinas de fazer café e sua equipe é a responsável por desenvolver uma classe para o novo modelo de cafeteiras que irão fabricar. No modelo anterior das cafeteiras, os usuários podiam selecionar se era para ser adicionado açúcar ou não, mas a nova versão da Super Cafeteira Brabo's Mega Power deve ser capaz de receber a quantidade (em gramas) de açúcar a ser colocada no café. Mesmo com essa nova possibilidade, o usuário não será obrigado a informar quanto de açúcar deseja. Deve-se adicionar 10 gramas de açúcar por padrão caso nenhum valor seja informado.

1. Para começar, crie a classe "MaquinaCafe" com um atributo chamado "acucarDisponivel", que será útil para saber a quantidade de açúcar disponível na máquina. Se o açúcar acabar, a máquina só permitirá cafezinhos sem açúcar, certo?

```
public class MaquinaCafe {  
  
    int acucarDisponivel;  
  
}
```

2. Agora crie um método chamado "fazerCafe" na mesma classe. Esse método deve verificar se a quantidade de açúcar disponível na máquina é suficiente e, claro, fazer o café.

```
void fazerCafe(int quantidadeAcucar) {  
    if (acucarDisponivel < quantidadeAcucar) {  
        System.out.println("Não há açúcar suficiente para fazer seu café.");  
    }  
    else {  
        acucarDisponivel -= quantidadeAcucar;  
        System.out.println("Fazendo cafezinho com " + quantidadeAcucar + " gramas de açúcar.");  
    }  
}
```

3. Como o usuário não será obrigado a informar a quantidade de açúcar a ser adicionado no café, crie outro método com o nome "fazerCafe", que não recebe nenhum parâmetro. Isso é uma sobrecarga de métodos! Veja como devem ficar os dois métodos que fazem cafés:

```
void fazerCafe() {  
    if (acucarDisponivel < 10) {  
        System.out.println("Não há açúcar suficiente para fazer seu café.");  
    }  
    else {  
        acucarDisponivel -= 10;  
        System.out.println("Fazendo cafezinho com 10 gramas de açúcar.");  
    }  
}
```

```

void fazerCafe(int quantidadeAcucar) {
    if (acucarDisponivel < quantidadeAcucar) {
        System.out.println("Não há açúcar suficiente para fazer seu café.");
    }
    else {
        acucarDisponivel -= quantidadeAcucar;
        System.out.println("Fazendo cafezinho com " + quantidadeAcucar +
            " gramas de açúcar.");
    }
}
}

```

4. Você deve ter percebido que os códigos dos dois métodos ficaram bem parecidos, certo? Sem pânico! Vamos melhorar fazendo um método chamar outro, aproveitando toda a lógica. Substitua os dois métodos por:

```

void fazerCafe() {
    fazerCafe(10);
}

void fazerCafe(int quantidadeAcucar) {
    if (acucarDisponivel < quantidadeAcucar) {
        System.out.println("Não há açúcar suficiente para fazer seu café.");
    }
    else {
        acucarDisponivel -= quantidadeAcucar;
        System.out.println("Fazendo cafezinho com " + quantidadeAcucar +
            " gramas de açúcar.");
    }
}
}

```

5. Crie uma classe agora para testar a máquina de café, afinal de contas, você deve evitar entregar código-fonte com bugs. Crie uma nova classe chamada "TesteMaquinaCafe" que instancia uma "MaquinaCafe" e atribui a quantidade de açúcar disponível.

```

class TesteMaquinaCafe {
    public static void main(String[] args) {
        MaquinaCafe maquina = new MaquinaCafe();
        maquina.acucarDisponivel = 30;
    }
}

```