MATA55 - PROGRAMAÇÃO ORIENTADA A OBJETOS

Aula 6: Criando classes e objetos em Java

Prof. Felipe Fernandes

31 Agosto de 2022

1. Escreva em Java uma classe **RestauranteCaseiro** que implemente o modelo descrito na Figura 1. Para isso, crie também uma classe **MesaDeRestaurante** que represente uma mesa de restaurante conforme mostrado na mesma figura. Algumas sugestões sobre a criação dessas classes são:

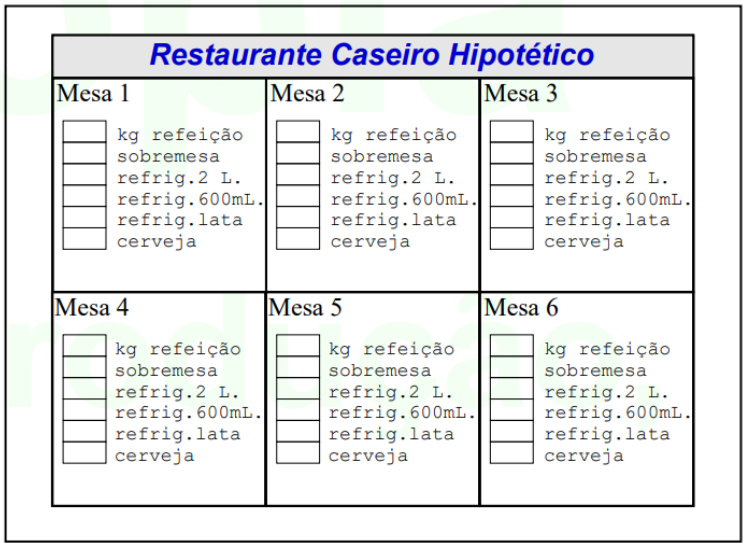

Figure 1

   (a) A classe **MesaDeRestaurante** deve ter atributos para representar a quantidade de cada pedido feito, um método **adicionaAoPedido** que incrementa a quantidade de pedidos feitos, o método **zeraPedidos** que cancela todos os pedidos feitos, isto é, faz com que a quantidade de pedidos seja zero para cada item, e o método **calculaTotal**,

que calcula o total a ser pago por aquela mesa. Como modelar cada item da comanda separadamente?

(b) A classe **RestauranteCaseiro** deve ter várias atributos que são instâncias da classe **MesaDeRestaurante**, para representar suas mesas separadamente.

(c) A classe **RestauranteCaseiro** também deve ter um método **adicionaAoPedido** que adicionará uma quantidade a um item de uma mesa. Esse método deverá chamar o método **adicionaAoPedido** da mesa à qual o pedido está sendo adicionado. A solução deste exercício requer a criação de um número predeterminado e imutável de instâncias de **MesaDeRestaurante** em **RestauranteCaseiro**. Comente sobre as vantagens e desvantagens de criar classes desta forma.

2. Imagine uma lâmpada que possa ter três estados distintos: apagada, acesa e meia luz. Faça o que se pede:

(a) Implemente a classe da figura 2.

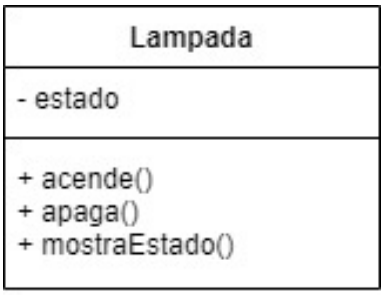

Figure 2

(b) Modifique a classe **Lampada** para que ela possa representar uma lâmpada onde a luminosidade pode ser ajustada com qualquer valor entre 0% (apagada) e 100% (acesa).

(c) Explique como modificar a classe **Lampada** para acrescentar o estado que represente a lâmpada **queimada**.

3. Em um banco hipotético, uma conta corrente está vinculada a uma pessoa física. Deve-se conhecer o nome, a data de nascimento e o CPF da pessoa física. Para cada conta corrente, deve-se conhecer o seu número, o seu titular e o seu saldo. Precisa-se ainda saber se a conta corrente é especial ou comum. Escreva um programa que permita abrir uma conta, depositar e sacar valores. Se a conta corrente for comum, a abertura da conta necessita apenas dos dados do titular. Se a conta for especial, então a abertura da conta necessita dos dados do titular e do saldo inicial. É requerida ainda a possibilidade de mostrar os dados da conta na tela.

4. Crie um algoritmo que simule o funcionamento de um caixa de supermercado. O caixa fica aberto até o fim do expediente e pode processar a compra de vários clientes. Cada cliente pode comprar vários itens. Ao ler cada item deve ser exibida uma mensagem para o operador do caixa perguntando se há mais itens a serem processados. Ao final, exiba quanto a compra custou ao cliente. E então solicite do operador do caixa a informação se deseja fechar o caixa. Encerre o algoritmo quando o usuário informar que deseja fechar o caixa.

5. Escreva em Java a classe **NumeroComplexo** que represente um número complexo. A classe deverá ter os seguintes métodos:

   - **inicializaNúmero**, que recebe dois valores como argumentos para inicializar os campos da classe (parte real e imaginária);
   - **imprimeNúmero**, que deve imprimir o número complexo encapsulado usando a notação $a + bi$ onde a é a parte real e b a imaginária;
   - **isIgual**, que recebe outra instância da classe **NumeroComplexo** e retorna *true* se os valores dos campos encapsulados forem iguais aos da instância passada como argumento;
   - **soma**, que recebe outra instância da classe **NumeroComplexo** e soma este número complexo com o encapsulado usando a fórmula $(a + bi) + (c + di) = (a + c) + (b + d)i$;
   - **subtrai**, que recebe outra instância da classe **NumeroComplexo** e subtrai o argumento do número complexo encapsulado usando a fórmula $(a + bi) - (c + di) = (a - c) + (b - d)i$;
   - **multiplica**, que recebe outra instância da classe **NumeroComplexo** e multiplica este número complexo com o encapsulado usando a fórmula $(a + bi) * (c + di) = (ac - bd) + (ad + bc)i$;

6. Crie uma classe denominada Elevador para armazenar as informações de um elevador dentro de um prédio. A classe deve armazenar o andar atual (térreo = 0), total de andares no prédio (desconsiderando o térreo), capacidade do elevador e quantas pessoas estão presentes nele. A classe deve também disponibilizar os seguintes métodos:

   - **Inicializa** : que deve receber como parâmetros a capacidade do elevador e o total de andares no prédio (os elevadores sempre começam no térreo e vazio);
   - **Entra** : para acrescentar uma pessoa no elevador (só deve acrescentar se ainda houver espaço);
   - **Sai** : para remover uma pessoa do elevador (só deve remover se houver alguém dentro dele);
   - **Sobe** : para subir um andar (não deve subir se já estiver no último andar);

- **Desce** : para descer um andar (não deve descer se já estiver no térreo);

7. Suponha que temos um conjunto de livros. Cada livro tem título, autor, editora, ISBN e ano. Deseja-se organizar esse conjunto de objetos em estruturas do tipo lista. Uma lista consiste em uma sequência de $n+2$ células, numeradas de 1 a $n+2$, onde cada célula contém (ou aponta para) um livro. Há duas células especiais, que não contêm nenhum livro: a célula cabeça (posição 1) e a célula cauda (posição $n+2$). Toda célula i da lista, $i = 1, ..., n+1$, possui uma referência para a célula $i+1$. Além disso, toda célula $i = 2, ..., n+2$ possui uma referência para a célula $i-1$. A célula cauda aponta para a célula cabeça, e a cabeça aponta para a cauda. Implemente esta lista utilizando orientação a objetos em Java. Implemente os seguintes métodos:

   (a) Adicionar um livro na posição j (o tamanho da lista cresce em uma unidade).

   (b) Remover um livro na posição j (o tamanho da lista diminui em uma unidade).

   (c) Retornar o j-ésimo elemento da lista.

   (d) Percorrer a lista da direita para a esquerda.

   (e) Percorrer a lista da esquerda para a direita.

   (f) Retornar a quantidade de elementos da lista, sem utilizar a estrutura de laço.

8. Encontre e explique o(s) erro(s) das classes abaixo.

```
1  class Registro De Eleitor
2    {
3    /**
4     * Declaração dos atributos desta classe
5     */
6    int tituloDeEleitor; // número do título do eleitor
7    String nome; // nome do eleitor
8    short zonaEleitoral; // número da zona eleitoral
9    } // fim da classe
```

Figure 3

(a)

```
1  class DoisValores
2    {
3    /**
4     * Declaração dos atributos desta classe
5     */
6    int valor1,valor2;
7    /**
8     * Declaração dos métodos desta classe
9     */
10   int maior()
11     {
12     if (valor1 > valor2)
13       return true;
14     else return false;
15     }
16   void menor()
17     {
18     if (valor1 < valor2)
19       return valor1;
20     else return valor2;
21     }
22   } // fim da classe
```

Figure 4

(b)

```
1  class NumeroComplexo
2    {
3    /**
4     * Declaração dos atributos desta classe
5     */
6    float real,imaginário;
7    /**
8     * Declaração dos métodos desta classe
9     */
10   float valor()
11     {
12     return real,imaginário;
13     }
14   } // fim da classe
```

Figure 5

(c)

```
class Amplitude
  {
  /**
   * Declaração dos atributos desta classe
   */
  double val1,val2,val3;
  /**
   * Declaração dos métodos desta classe
   */
  double amplitude()
    {
    double amplitude2()
      {
      return val1-val2;
      }
    return amplitude2()-val3;
    }
  } // fim da classe
```

Figure 6

(d)

```
class Registro De Eleitor
  {
  /**
   * Declaração dos atributos desta classe
   */
  int tituloDeEleitor; // número do título do eleitor
  String nome; // nome do eleitor
  short zonaEleitoral; // número da zona eleitoral
  } // fim da classe
```

Figure 7

(e)

9. Quais dos identificadores abaixo podem ser usados como nomes de classes, atributos, métodos e variáveis em Java? Quais não podem, e por quê?

   (a) contador
   (b) 1contador
   (c) contador de linhas
   (d) Contador
   (e) count

10. Qual o tipo de dado ou classe mais adequados para representar:

   (a) O número de municípios de um estado do Brasil.
   (b) O nome de um estado do Brasil.
   (c) A população de um estado do Brasil.
   (d) A área do Brasil em quilômetros quadrados.
   (e) A população total do mundo.
   (f) O CEP de um endereço no Brasil.
   (g) O nome de uma rua em um endereço no Brasil.
   (h) A altura de uma pessoa em metros.
   (i) O peso de uma pessoa em quilos.
   (j) A temperatura corporal de uma pessoa.
   (k) O sexo de uma pessoa.
   (l) A altura de uma pessoa em milímetros.