Universidade Federal de Campina Grande
Centro de Engenharia Eletrica e Informática
Departamento de Sistemas e Computação
Graduação em Ciência da Computação

Exercício de revisão de Java 02

Objetivo: Relembrar os conceitos de Java relacionados a Interfaces e Arrays. É importante que voce resolva com atenção pois diversos conceitos e práticas de programação serão essenciais no decorrer da disciplina. Dessa forma, se voce tiver qualquer dúvida, procure o professor ou monitor. Utilize o horário da aula para resolver os exercícios.

1. Para que serve o recurso de interface em Java? Quais situações/cenários requerem seu uso?
2. Baseando-se nos itens abaixo e usando os recursos de orientação a objeto que voce conhece, implemente entidades da forma que voce julgar mais adequado.
   - Como sabemos, cada forma geométrica tem sua área calculada através de uma fórmula específica. Considerando apenas algumas delas temos:
     a. Triângulo: (base*altura)/2
     b. Retangulo: (base*altura)
     c. Quadrado: $lado^2$ (um tipo especial de retangulo)
     d. Círculo: $\pi * r^2$ (Em Java π é definido pela constante Math.PI)
   - Imagine uma classe com um método main e que faz uso de uma forma geometrica. O objetivo dessa classe é mostrar a area da forma geometrica (com System.out.println mesmo). Tente usar recursos de forma que essa classe não conheça os detalhes de como cada forma geometrica calcula sua área.
3. Reutilize as classes de produtos implementadas no exercício anterior. Implemente a classe de repositórios em array para Produto. Os métodos são indicados abaixo. Métodos com * têm um significado especial: ou eles executam com sucesso ou eles retornam um erro (RuntimeException). A classe deve armazenar e gerenciar produtos usando uma estrutura array internamente.

RepositorioProdutosArray

`int procurarIndice(int codigo)`
`boolean existe(int codigo)`
`void inserir(Produto produto)`
`void atualizar(Produto produto)*`
`void remover(int codigo)*`
`Produto procurar(int codigo)*`

   - procurarIndice(int codigo) = recebe o codigo do produto e devolve o indice desse produto no array ou -1 caso ele nao se encontre no array. Esse método é util apenas na implementacao com arrys por questoes de localizacao. Outras classes que utilizam outras estruturas internas podem nao precisar desse método. Dessa forma, que modificador de visibilidade o método deve ter?
   - existe(int codigo) = recebe o codigo e diz se tem produto com esse codigo armazenado.
   - inserir(Produto produto) = insere um novo produto (sem se preocupar com duplicatas).
   - atualizar(Produto produto) = atualiza um produto armazenado. Note que, para localizacao o código do produto será utilizado.

- remover(int codigo) = remove produto com determinado codigo, se existir, ou entao retorna um erro, caso contrário.
- procurar(int codigo) = retorna um produto com determinado codigo ou entao um erro, caso o produto nao esteja armazenado.

4. Agora é hora de generalizar a coleção implementada. Para isso utilize o recurso de extrair interface a partir de uma classe no Eclipse. Use o nome RepositorioProdutos.java na interface que voce vai estrair do RepositorioProdutosArray.
5. Em seguida construa uma classe (RepositorioProdutoArrayList.java) que implementa a interface extraída e usa ArrayList como estrutura interna para armazenar os produtos. Se tiver alguma duvida, consulte a documentação da classe ArrayList.java.