Universidade Federal de Pernambuco
Centro de Informática

Introdução a Programação
1^a Prova Escrita

Ricardo Massa e Sérgio Soares

25 de Maio de 2010

Considere um sistema de controle de estoque que mantêm os produtos de um supermercado. Cada produto tem seu código e sua descrição, ambos do tipo `String`, além da quantidade do produto em estoque (número de unidades disponíveis em estoque).

1. (2 pontos) Defina a classe `Produto`, um construtor para a mesma e métodos `get` e `set` para todos os seus atributos.

```
//RESPOSTA
public class Produto {
    private String codigo;
    private String descricao;
    private int quantidade;
    public Produto(String codigo, String descricao, int quantidade) {
        this.codigo = codigo;
        this.descricao = descricao;
        this.quantidade = quantidade;
    }
    public String getCodigo() {
        return codigo;
    }
    public void setCodigo(String codigo) {
        this.codigo = codigo;
    }
    public String getDescricao() {
        return descricao;
    }
    public void setDescricao(String descricao) {
        this.descricao = descricao;
    }
    public int getQuantidade() {
        return quantidade;
    }
    public void setQuantidade(int quantidade) {
        this.quantidade = quantidade;
    }
}
```

2. Defina a classe `RepositorioProdutosArray` que armazena objetos da classe `Produto` em um array (0,5 ponto). Defina ainda o construtor da classe `RepositorioProdutosArray` (1 ponto) que deve receber o tamanho do array como parâmetro, e os seguintes métodos:

   - (0,5 ponto) `inserir` — Recebe um objeto `Produto` como parâmetro e insere o mesmo no array de produtos;
   - (1 ponto) `procurar` — Recebe um código (de `Produto`) como parâmetro, e retorna o produto com o referido código, caso o mesmo esteja no array. Se não houve um `Produto` com o código informado no array, o método deve retornar `null`;
   - (1 ponto) `remover` — Recebe um código (de `Produto`) como parâmetro, e remove o produto com o referido código do array;

- (1 ponto) `atualizar` — Recebe um objeto `Produto` como parâmetro, varre o array em busca de um `Produto` com o mesmo código do objeto recebido e, se encontrar, substitui o `Produto` do array pelo objeto recebido como parâmetro;
- (1 ponto) `existe` — Recebe um código (de `Produto`) como parâmetro, retorna `true` caso exista um `Produto` com o código fornecido no array e `false` se não houve;

```
//RESPOSTA
public class RepositorioProdutosArray {
    private Produto[] produtos;
    private int indice;
    public RepositorioProdutosArray(int tamanho) {
        produtos = new Produto[tamanho];
        indice = 0;
    }
    public void inserir(Produto produto) {
        produtos[indice] = produto;
        indice = indice + 1;
    }
    public Produto procurar(String numero) {
        Produto resposta = null;
        int i = this.getIndice(numero);
        if (i != this.indice) {
            resposta = this.produtos[i];
        }
        return resposta;
    }
    public void remover(String numero) {
        int i = this.getIndice(numero);
        if (i != this.indice) {
            this.indice = this.indice - 1;
            this.produtos[i] = this.produtos[this.indice];
            this.produtos[this.indice] = null;
        }
    }
    public void atualizar(Produto produto) {
        int i = this.getIndice(produto.getCodigo());
        if (i != this.indice) {
            this.produtos[i] = produto;
        }
    }
    public boolean existe(String codigo) {
        int i = this.getIndice(codigo);
        return (i != this.indice);
    }
    private int getIndice(String codigo) {
        boolean achou = false;
        int i = 0;
        while ((!achou) && (i < this.indice)) {
            if (produtos[i].getCodigo().equals(codigo)) {
                achou = true;
            } else {
                i = i + 1;
            }
        }
        return i;
    }
}
```

3. (2 pontos) Por que se considera um pre-requisito para a programação orientada a objetos a qualificação de atributos como `private`?

```
//RESPOSTA
Queremos impedir o acesso externo (por ouras classes) aos atributos de uma classe para:

1 - Impedir que outras classes acessem diretamente os atributos, o que faria as mesmas
serem dependentes das implementações destes atributos. Desta forma, sempre que os atributos
de uma classe C mudarem, os clientes desta classe (classes que usam C) sempre afetados.
Tornando os atributos privados, podemos mudar os mesmos sem que as classes clientes sejam
afetadas (desde que as assinaturas dos métodos get e set não sejam alteradas)

2 - Garantir um controle sobre a manipulação do atributo (que o mesmo será sempre
modificado consistentemente) através do método set.
```

QUESTÃO DESAFIO: (2 pontos) Implemente em Java a função recursiva Máximo Divisor Comum, conforme especificação a seguir:

$$mdc(x, y) = \begin{cases} x & \text{se } y = 0 \\ mdc(y, x\%y)) & \text{se } x \geq y \text{ e } y > 0 \end{cases}$$

```
//RESPOSTA
    public int mdc(int x, int y) {
        int resposta = 0; // 0 = erro
        if (y == 0) {
            resposta = x;
        } else if (x >= y && y > 0){
            resposta = mdc(y, x%y);
        }
        return resposta;
    }
```