

Universidade Federal de Pernambuco
Centro de Informática

Introdução a Programação
2ª Prova Escrita

Ricardo Massa e Sérgio Soares

29 de Junho de 2010

1. (2,5 pontos) Levando em conta aspectos de qualidade de software e de programação orientada a objetos, defina no pacote `peessoas` uma classe básica para representar uma pessoa com nome, cpf e endereço (rua, numero, complemento, bairro, cep, cidade e estado).

```
//RESPOSTA
package peessoas;
public class Pessoa {
    private String nome;
    private String cpf;
    private Endereco endereco;
    public Pessoa(String nome, String cpf, Endereco endereco) {
        this.nome = nome;
        this.cpf = cpf;
        this.endereco = endereco;
    }
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public String getCpf() {
        return cpf;
    }
    public void setCpf(String cpf) {
        this.cpf = cpf;
    }
    public Endereco getEndereco() {
        return endereco;
    }
    public void setEndereco(Endereco endereco) {
        this.endereco = endereco;
    }
}

//PARTE 2 DA RESPOSTA (menos um ponto para quem não declarou essa classe)
package peessoas;
public class Endereco {
    private String rua;
    private String numero; // nao faz mal quem declarou como int
    private String complemento;
    private String bairro;
    private String cep;
    private String cidade;
    private String estado;
}
```

```

public Endereco(String rua, String numero, String complemento,
                 String bairro, String cep, String cidade, String estado) {
    this.rua = rua;
    this.numero = numero;
    this.complemento = complemento;
    this.bairro = bairro;
    this.cep = cep;
    this.cidade = cidade;
    this.estado = estado;
}
public String getRua() {
    return rua;
}
public void setRua(String rua) {
    this.rua = rua;
}
public String getNumero() {
    return numero;
}
public void setNumero(String numero) {
    this.numero = numero;
}
public String getComplemento() {
    return complemento;
}
public void setComplemento(String complemento) {
    this.complemento = complemento;
}
public String getBairro() {
    return bairro;
}
public void setBairro(String bairro) {
    this.bairro = bairro;
}
public String getCep() {
    return cep;
}
public void setCep(String cep) {
    this.cep = cep;
}
public String getCidade() {
    return cidade;
}
public void setCidade(String cidade) {
    this.cidade = cidade;
}
public String getEstado() {
    return estado;
}
public void setEstado(String estado) {
    this.estado = estado;
}
}
}

```

2. (3 pontos) Defina uma Coleção de Negócio (CadastroPessoas) para objetos da classe básica Pessoa, definida na questão anterior, que utiliza a Interface Negócio-Dados abaixo. A coleção de negócio deve ter os métodos cadastrar (que não permite cadastrar duas pessoas com o mesmo cpf), atualizar, remover e procurar.

```
package dados;
public interface RepositorioPessoas {
    public void inserir(Pessoa pessoa);
    public void atualizar(Pessoa pessoa) throws PessoaNaoEncontradaException;
    public Pessoa procurar(Pessoa pessoa) throws PessoaNaoEncontradaException;
    public void remover(Pessoa pessoa) throws PessoaNaoEncontradaException;
    public boolean existe(Pessoa pessoa);
}
```

Considere que a classe PessoaNaoEncontradaException já existe no pacote dados. Caso precise de mais alguma exceção, defina a mesma na prova.

```
//RESPOSTA parte 1
package pessoas;
import dados.PessoaNaoEncontradaException;
import dados.RepositorioPessoas;
public class CadastroPessoas {
    private RepositorioPessoas pessoas;
    public CadastroPessoas(RepositorioPessoas pessoas) {
        this.pessoas = pessoas;
    }
    public void cadastrar(Pessoa pessoa) throws PessoaJaCadastradaException {
        if (pessoas.equals(pessoa)) {
            throw new PessoaJaCadastradaException();
        } else {
            pessoas.inserir(pessoa);
        }
    }
    public void atualizar(Pessoa pessoa) throws PessoaNaoEncontradaException {
        pessoas.atualizar(pessoa);
    }
    public void remover(Pessoa pessoa) throws PessoaNaoEncontradaException {
        pessoas.remover(pessoa);
    }
    public Pessoa procurar(Pessoa pessoa) throws PessoaNaoEncontradaException {
        return pessoas.procurar(pessoa);
    }
}
```

```
//RESPOSTA parte 2
package pessoas; //essa é uma exceção de negócio, deve ser neste pacote
public class PessoaJaCadastradaException extends Exception {
    public PessoaJaCadastradaException() {
        super("Pessoa ja cadastrada");
    }
}
```

3. (2 pontos) Qual a vantagem de termos no projeto uma interface como a da questão anterior?

```
//RESPOSTA
A interface torna a class CadastroPessoas independente do meio de armazenamento de dados. Como a classe CadastroPessoas é uma classe de negócio (Coleção de Negócio) as regras de negócio ficam desacopladas (separadas) das regras de armazenamento de dados. Logo, os dois interesses (negócio e dados) podem evoluir independentemente.
```

4. (2,5 pontos) Considerando que na aplicação em questão existem clientes e funcionários, defina duas novas classes para representar estes tipos, utilizando conceitos de programação orientada a objetos e aspectos de qualidade de software. Além de nome e cpf, clientes têm limite de crédito, e funcionários têm salário.

```
//RESPOSTA parte 1
package pessoas;
public class Cliente extends Pessoa {
    private double credito;
    public Cliente(String nome, String cpf, Endereco endereco, double credito) {
        super(nome, cpf, endereco);
        this.credito = credito;
    }
    public double getCredito() {
        return credito;
    }
    public void setCredito(double credito) {
        this.credito = credito;
    }
}

//RESPOSTA parte 2
package pessoas;
public class Funcionario extends Pessoa {
    private double salario;
    public Funcionario(String nome, String cpf, Endereco endereco, double salario) {
        super(nome, cpf, endereco);
        this.salario = salario;
    }
    public double getSalario() {
        return salario;
    }
    public void setSalario(double salario) {
        this.salario = salario;
    }
}
```

5. (1 ponto) DESAFIO: Considere a definição das seguintes classes.

```
public class A {
    public void m() {
        System.out.println("A");
    }
}

public class B extends A {
    public void m() {
        System.out.println("B");
    }
}

public class Programa {
    public void x() {
        A a = new B();
        a.m();
        ((B)a).m();
        ((A)a).m();
    }
}
```

O que será impresso na saída padrão ao executar o método x da classe Programa? Explique por que. Quanto mais completa a sua resposta, melhor.

```
//RESPOSTA
Ao executar o método x da classe Programa, sairá impresso no console a letra B em três linhas. Isto acontece pois apesar do cast, o objeto criado no método x é do tipo B. Como em Java temos o conceito de ligação dinâmica, no momento da execução vale o objeto que foi criado, não os tipos das variáveis ou do cast. Logo, é sempre o método m do objeto B que será executado no programa apresentado.
```