Instituto Federal da Paraíba
Unidade Acadêmica de Informática
Curso Superior de Tecnologia em Sistemas para Internet
Disciplina: Programação Orientada a Objetos
Professor: Frederico C. G. Pereira

Prática 04 – Objetos em Java

por Frederico C. G. Pereira

Objetivos

- Praticar a implementação de objetos
- Praticar o encapsulamento

I. Transformando programas imperativos em orientados a objetos

Dado o código abaixo, que apesar de estar escrito numa linguagem orientada a objetos, não utiliza os conceitos da OO corretamente, transforme-o para que fique orientado a objetos.

1) Você recebeu um programa escrito de forma totalmente procedural que lê uma temperatura em graus Celsius, converte para Fahrenheit e imprime o resultado. O programa consiste apenas de um método main com variáveis locais e operações matemáticas.

```java
package br.edu.ifpb.poo;

public class ConversorApp {
    public static void main(String[] args) {
        double celsius = 25.0;
        double fahrenheit;

        fahrenheit = (celsius * 9 / 5) + 32;

        System.out.println(celsius + " °C equivalem a " + fahrenheit + " °F");
    }
}
```

- Reescreva o programa para utilizar uma abstração em Java, responsável pela conversão entre Celsius e Fahrenheit.
- Crie também a classe ConversorApp, que contém o método main e utiliza um objeto da classe criada acima.
- Ao final, o programa deve manter a funcionalidade, mas com a lógica de conversão encapsulada no objeto ConversorTemperatura.

2) Um programa procedural foi escrito para ler a base e a altura de um retângulo, calcular a área e o perímetro e imprimir os resultados. Todo o cálculo foi feito dentro do método main usando variáveis locais.

```
package br.edu.ifpb.poo;

public class RetanguloApp {
    public static void main(String[] args) {
        double base = 5.0;
        double altura = 3.0;

        double area = base * altura;
        double perimetro = 2 * (base + altura);

        System.out.println("Retangulo: Ret(Base= " + base + ", Altura= " + altura + ")");
        System.out.println("> Área: " + area);
        System.out.println("> Perímetro: " + perimetro);
    }
}
```

- Reescreva o programa para utilizar uma abstração chamada Retangulo, que deve armazenar base e altura e oferecer métodos para calcular a área e o perímetro.
- Crie também a classe RetanguloApp, que contém o método main e utiliza um objeto da classe Retangulo.
- O programa final deve imprimir os mesmos resultados, mas com os cálculos encapsulados dentro da classe Retangulo.

3) Dado o programa abaixo, reescreva-o, de modo que ele se torne orientado a objetos. Que entidade você identifica como sendo importante de se tornar uma classe? Analise o código antes de iniciar a codificação.

```
package br.edu.ifpb.poo;

public class SubnetApp {
    public static void main(String[] args) {
        int[] ip = {192, 168, 10, 15};
        int[] mask = {255, 255, 255, 0};
        int[] network = new int[4];

        for (int i = 0; i < 4; i++) {
            network[i] = ip[i] & mask[i];
        }

        System.out.println("IP: " + ip[0] + "." + ip[1] + "." + ip[2] + "." + ip[3]);
        System.out.println("Máscara: " + mask[0] + "." + mask[1] + "." + mask[2] + "." + mask[3]);
        System.out.println("Endereço de rede: " +
                network[0] + "." + network[1] + "." + network[2] + "." + network[3]);
    }
}
```

4) Para cada uma das entidades acima, implemente o método toString() de modo a permitir a conversão implícita do seu objeto para String quando ele for concatenado com um String Java. Imprima seu objeto concatenado com algum String no main() para testar a implementação de toString().