



INSTITUTO FEDERAL
SANTA CATARINA

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA
CAMPUS GASPAR

CURSO TÉCNICO EM INFORMÁTICA

TURMA: 16

ALUNO: MAIARA JUNCHES SEEMANN

DATA: 15/08/2023

LISTA DE REVISÃO - PRÁTICAS PARA DISPOSITIVOS MÓVEIS

Questão 1.

R: Abstração:

A abstração oculta os detalhes de implementação, criando “coisas” abstratas para serem reutilizadas de diversas maneiras diferentes, não podendo portanto instanciar, mas sim herdar. Para isso, o termo *abstract* deve estar presente na declaração da classe, do atributo, do método, etc.

Encapsulamento:

O encapsulamento de classes, objetos, métodos, etc. serve para torná-los passíveis, ou não, de serem reutilizados de formas objetivas. Para isso são usados os encapsuladores como *public* e *private*.

Herança:

A herança permite o compartilhamento de atributos e métodos entre classes, podendo assim uma classe herdar os elementos da outra e reutilizá-los. Para isso, usa-se o termo *extends* entre os nomes das classes na declaração da classe.

Polimorfismo:

O polimorfismo possibilita que objetos de classes diferentes respondam a mesma mensagem de maneiras diferentes, como por exemplo utilizar o mesmo método da classe mãe e realizá-lo de formas diferentes.

Questão 2.

```
package questao2;
public class Pessoa {
    private String nome;
    private String email;
    private String telefone;
    private Endereco endereco;
    public Pessoa(String nome, String email, String telefone, Endereco endereco)
    {

        this.nome = nome;
        this.email = email;
        this.telefone = telefone;
        this.endereco = endereco;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }
}
```

```

        public String getEmail() {
            return email;
        }
        public void setEmail(String email) {
            this.email = email;
        }
        public String getTelefone() {
            return telefone;
        }
        public void setTelefone(String telefone) {
            this.telefone = telefone;
        }
        public Endereco getEndereco() {
            return endereco;
        }
        public void setEndereco(Endereco endereco) {
            this.endereco = endereco;
        }
    }
}
package questao2;
public class Endereco {
    private String cidade;
    private String rua;
    private String numero;
    public Endereco(String cidade, String rua, String numero) {

        this.cidade = cidade;
        this.rua = rua;
        this.numero = numero;
    }

    public String getCidade() {
        return cidade;
    }
    public void setCidade(String cidade) {
        this.cidade = cidade;
    }
    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;
    }
}
}
package questao2;

import java.util.ArrayList;

```

```

import questao2.Pessoa;

public class ListaPessoa {

    private ArrayList<Pessoa> listaPessoa;

    public ListaPessoa() {
        listaPessoa = new ArrayList<>();
    }
    public void adicionarPessoa(Pessoa pessoa) {
        listaPessoa.add(pessoa);
    }
    public ArrayList<Pessoa> getListaPessoa() {
        return listaPessoa;
    }
}

```

e)R:

Encapsulamento:

Abaixo é declarada a classe pessoa com encapsulamento público, ou seja, ela é visível e passível de uso nas outras classes. Além disso, nela é declarado um atributo nome, também publico.

```

public class Pessoa;
public String Nome;

```

Agora, na classe Main, é declarado o objeto pessoa do tipo Pessoa, da classe pessoa, o que só foi possível pois esta classe é pública.

Questão 3.

```

package questao3;
public abstract class Forma {

    public abstract double area();
    public abstract double perimetro();
}

package questao3;
public class Retangulo extends Forma {

    private double altura;
    private double largura;
    public Retangulo(double altura, double largura) {

        this.altura = altura;
        this.largura = largura;
    }
    public double getAltura() {
        return altura;
    }
    public void setAltura(double altura) {
        this.altura = altura;
    }
}

```

```

    }
    public double getLargura() {
        return largura;
    }
    public void setLargura(double largura) {
        this.largura = largura;
    }
    @Override
    public double area() {
        return altura * largura;
    }
    @Override
    public double perimetro() {
        return 2*(altura+largura);
    }
}

package questao3;
import questao3.Retangulo;
public class Quadrado extends Retangulo{

    public Quadrado (double lado) {
        super (lado, lado);
    }
    public double getLado() {

        return getAltura();
    }
    public void setLado(double lado) {
        setAltura(lado);

        setLargura(lado);
    }
}

package questao3;
public class Circulo extends Forma {
    private double raio;
    public Circulo(double raio) {
        this.raio = raio;
    }
    public double getRaio() {
        return raio;
    }
    public void setRaio(double raio) {
        this.raio = raio;
    }
    @Override
    public double area() {

        return Math . PI * raio * raio;
    }
    @Override
    public double perimetro() {

```

```

return 2 * Math . PI * raio;
}
}
package questao3;
public class MainForma {

    public static void main(String[] args) {

        double alturaRetangulo = 5.0;
        double larguraRetangulo = 8.0;
        Retangulo retangulo = new Retangulo(alturaRetangulo,
larguraRetangulo);

        double ladoQuadrado = 4.0;
        Quadrado quadrado = new Quadrado(ladoQuadrado);

        double raioCirculo = 3.0;
        Circulo circulo = new Circulo(raioCirculo);

        double areaRetangulo = retangulo.area();
        double perimetroRetangulo = retangulo.perimetro();
        double areaQuadrado = quadrado.area();
        double perimetroQuadrado = quadrado.perimetro();
        double areaCirculo = circulo.area();
        double perimetroCirculo = circulo.perimetro();

        System.out.println("Retângulo:\n");
        System.out.println("Área:" + areaRetangulo);
        System.out.println("Perímetro:" + perimetroRetangulo);
        System.out.println("\nQuadrado:\n");
        System.out.println("Área:" + areaQuadrado);
        System.out.println("Perímetro:" + perimetroQuadrado);
        System.out.println("\nCírculo:\n");
        System.out.println("Área:" + areaCirculo);
        System.out.println("Perímetro:" + perimetroCirculo);
    }
}

```

e) R: Abstração:

Na classe Forma são declarados os métodos abstratos area e perimetro, os quais, por serem abstratos, estão passíveis a serem utilizados de diversas maneiras em outras classes que os herdarem.

Polimorfismo:

Agora, na classe Retangulo, que é filha de Forma, os métodos da classe Forma são utilizados para calcular a área e o perímetro de retângulos, com sua devida fórmula. Já na classe Circulo, que também é filha de Forma, os métodos da classe Forma são também utilizados com as devidas fórmulas para o cálculo de área e perímetro de círculos.

Ou seja, os métodos abstratos de Forma foram utilizados por classes diferentes de diferentes formas, graças ao polimorfismo.

