

Programação Orientada a Objetos: Lista de exercícios do 2º bimestre/maio_2019.

Para todos os exercícios apresentar impresso o código fonte e a simulação da aplicação funcionando (em grupos de até 4 alunos). Entrega no dia da prova 2.

Considerar que em uma empresa o setor de RH constrói a folha de pagamentos mensal utilizando as informações:

- Nome do funcionário;
- Identidade funcional (valor alfa-numérico);
- Função dentro da empresa;
- Fator do plano de carreira, que é multiplicado ao salário, sendo:

Fator A – empregados que estão no nível mais alto do plano na empresa;

Fator B – empregados que estão em um nível intermediário;

Fator C – empregados iniciantes na empresa.

- Número de dias trabalhados no mês;
- Horas extras.

Exercício 1. Construir uma aplicação que utilize os conceitos de orientação a objetos relacionados à abstração, encapsulamento, métodos construtores, herança e polimorfismo:

- Construir a classe que represente a abstração da aplicação;
- Construir a(s) classe(s) que manipulam os objetos;
- Construir a classe que implemente a interface do usuário com a aplicação.
- A aplicação deve retornar a folha de pagamento consolidada para um conjunto de empregados da empresa.

Exercício 2

```
package geometricas;

public class Figuras_geometricas {

    private double lado_a;
    private double lado_b;
    private static double lado_c;
    private double area;

    Figuras_geometricas ( ){
        lado_a = 2.0;
        lado_b = 3.0;
        lado_c = 1.0;}

    public void setLado_a(double lado_a)
    {this.lado_a = lado_a;}

    public void setLado_b(double lado_b)
    {this.lado_b = lado_b;}

    public double getLado_a() {
        return lado_a;
    }

    public double getLado_b() {
        return lado_b;
    }

    public double getArea (double lado_a, double lado_b )
    {this.lado_a = lado_a; this.lado_b = lado_b;
    area =lado_a*lado_b; return area; }

    public double getArea ( )
    { area = lado_c*lado_c; return area;}

    public void resultado ( )
    {System.out.println ("resultado = " + getArea(getLado_a(), getLado_b()));}

}
```

```
package geometricas;

public class Figuras_1 extends Figuras_geometricas {
    Figuras_geometricas e = new Figuras_geometricas( );
    public static void main (String args[]){
        Figuras_geometricas x = new Figuras_geometricas( );
        System.out.println("Area = " + x.getArea(x.getLado_a(),x.getLado_b())); }

    public void resultado ( )
    {System.out.println ("resultado = " + e.getArea( ));}

}
```

```

package geometricas;

public class Figuras_2 extends Figuras_geometricas{
    Figuras_geometricas v = new Figuras_geometricas( );
    public static void main (String args[]){
        Figuras_geometricas y = new Figuras_geometricas( );
        System.out.println("lado a = " + y.getLado_a());
    }
    public void resultado ( )
        {System.out.println ("resultado = " + v.getArea( ));}
    }

package geometricas;

public class Calcula_areas {
    public static void main (String args[]){
        Figuras_1 a = new Figuras_1( );
        Figuras_2 b = new Figuras_2( );
        Figuras_geometricas u = new Figuras_geometricas ( );
        u.setLado_a(4.0);
        u.setLado_b(5.0);
        a.resultado( );
        b.resultado( );
        u.resultado( );
    }
}

```

Dadas as classes Figuras_Geometricas, Figuras_1, Figuras_2 e Calcula_areas:

Questão 1: identificar os atributos.

Questão 2: identificar o método relacionado ao conceito de sobrecarga de métodos.

Questão 3: Esquematizar as classes conforme um diagrama de classes.

Questão 4: mostrar o resultado, no console, da execução da classe Figuras_1.

Questão 5: mostrar o resultado, no console, da execução da classe Figuras_2.

Questão 6: mostrar o resultado, no console, da execução da classe Calcula_areas.

Exercício 3

Exercícios de 01 a 15 dos arquivos “Exemplos de Classes Abstratas e Interfaces I, II e III.