

Aluno(a): Franciúne Almeida

Faça os exercícios abaixo e cole a solução (diagramas, códigos) nos locais indicados abaixo para cada exercício, salve e envie este arquivo via Moodle.

1) Escreva em Java uma classe Contador, que encapsule um valor usado para contagem de itens ou eventos. A classe deve oferecer métodos que devem:

- a) Zerar;
- b) Incrementar;
- c) Retornar o valor do contador.

MAIN:

```
import java.util.Scanner;
```

```
public class Main{
```

```
    public static void main(String[] args){
```

```
        Scanner input = new Scanner(System.in);
```

```
        Contador c = new Contador();
```

```
        int op;
```

```
        do{
```

```
            System.out.println("| 1- Zerar          |");
```

```
            System.out.println("| 2- Incrementar |");
```

```
            System.out.println("| 3- Imprimir    |");
```

```
            System.out.println("| 0- Sair        |");
```

```
            op = input.nextInt();
```

```
            switch(op){
```

```
                case 1:
```

```
                    c.Zera();
```

```
                    break;
```

```
                case 2:
```

```
                    c.Incrementa();
```

```
                    break;
```

```
                case 3:
```

```
                    c.Imprime();
```

```
                    break;
```

```

        case 0:
            System.exit(0);
            break;
        default:
            System.out.println("| OPÇÃO INVÁLIDA! |");
            break;
    }
} while (op!=0);
}
}

```

CONTADOR:

```

public class Contador{

    private int count;

    public void Zera(){
        count = 0;
    }

    public void Incrementa(){
        count++;
    }

    public void Imprime(){
        System.out.println("Contador:" + count);
    }
}

```

2) Escreva em Java uma classe Ponto2D que represente um ponto no plano cartesiano. Além dos atributos por você identificados, a classe deve oferecer os seguintes membros:

- a) Construtores sobrecarregados que permitam a inicialização do ponto:
 - i) Por default (sem parâmetros) na origem do espaço 2D;
 - ii) Num local indicado por dois parâmetros do tipo double (indicando o valor de abcissa e ordenada do ponto que está sendo criado).
- b) Métodos de acesso (getter/setter) dos atributos do ponto;
- c) Método de movimentação do ponto com os mesmos parâmetros indicados para o construtor com parâmetros;
- d) Método de representação do objeto como String.

MAIN:

```

import java.util.Scanner;

```

```

public class Main{

    public static void main(String[] args){

        Scanner input = new Scanner(System.in);
        int x, y;
        int op;
        String ponto;
        Ponto p = new Ponto();

        ponto:"");
        System.out.println("\n| Escolha a forma de inicialização do
        ponto:");
        System.out.println("| 1- Default");
        System.out.print("| 2- Personalizada\n| ");
        op = input.nextInt();

        switch(op){
            case 1:
                p = new Ponto();
                break;
            case 2:
                System.out.print("\n| Digite x\n| ");
                x = input.nextInt();
                System.out.print("| Digite y\n| ");
                y = input.nextInt();
                p = new Ponto(x , y);
                break;
        }

        do{
            System.out.println("\n| 1- Set");
            System.out.println("| 2- Get");
            System.out.println("| 3- Mover ponto");
            System.out.println("| 4- Mostrar coordenadas");
            System.out.print("| 0- Sair\n| ");
            op = input.nextInt();

            switch(op){
                case 1:
                    System.out.print("\n| Digite x\n| ");

```

```

        p.setX(input.nextInt());
        System.out.print("| Digite y\n| ");
        p.setY(input.nextInt());
        break;
    case 2:
        x = p.getX();
        y = p.getY();
        System.out.println("\n| x = " + x + "\n| y = " +
y);
        break;
    case 3:
        System.out.print("\n| Digite unidades para mover
em x\n| ");
        p.Move(input.nextInt(), 0);
        System.out.print("| Digite unidades para mover em
y\n| ");
        p.Move(0, input.nextInt());
        break;
    case 4:
        ponto = p.toString();
        System.out.println(ponto);
        break;
    case 0:
        System.exit(0);
        break;
    default:
        System.out.println("| OPÇÃO INVÁLIDA!");
        break;
    }

    } while (op!=0);
}

```

PONTO:

```

public class Ponto{

    private int x, y;

    public Ponto(int x, int y){
        this.x = x;

```

```

        this.y = y;
    }

    public Ponto() {
        this.x = 0;
        this.y = 0;
    }

    public void setX(int x) {
        this.x = x;
    }

    public void setY(int y) {
        this.y = y;
    }

    public int getX() {
        return this.x;
    }

    public int getY() {
        return this.y;
    }

    public void Move(int mx, int my) {
        this.x += mx;
        this.y += my;
    }

    public String toString() {
        return ("\n| x = " + this.x + "\n| y = " + this.y);
    }
}

```

3) Escreva em Java uma classe que represente um círculo no plano cartesiano. Forneça os seguintes membros de classe:

- a) Um construtor que receba o raio e um ponto (o centro do círculo);
- b) Um construtor que receba o raio e posicione o círculo na origem do espaço cartesiano;
- c) Métodos de acesso ao atributo raio do círculo;
- d) Métodos inflar e desinflar, que, respectivamente, aumentam e diminuem o raio do círculo de um dado valor;
- e) Método mover, que move o círculo para um local indicado por dois parâmetros do tipo double (indicando o valor de abcissa e ordenada do ponto para onde o círculo se move);

f) Método que retorna a área do círculo.

CIRCULO:

```
public class Circulo{

    private Centro c = new Centro();
    private int r;
    private double pi = 3.141592653589793;
    private double area;

    public Circulo(int r){
        this.r = r;
    }

    public Circulo(int r, int x, int y){
        this.r = r;
        c.DefineCentro(x, y);
    }

    public void setR(int r){
        this.r = r;
    }

    public int getR(){
        return this.r;
    }

    public void Infla(int rr){
        r += rr;
    }

    public void Desinfla(int rr){
        r -= rr;
    }

    public void MoveCentro(int x, int y){
        c.DefineCentro(x, y);
    }

    public double CalculaArea(){
        area = pi * (Math.pow(r,2));
    }
}
```

```
        return area;
    }
}
```

CENTRO:

```
public class Centro{

    private int x, y;

    public void DefineCentro(int x, int y){
        this.x = x;
        this.y = y;
    }

}
```

MAIN:

```
import java.util.Scanner;

public class Main{

    public static void main(String[] args){

        Scanner input = new Scanner(System.in);
        double area;
        int op;
        int r, x, y;

        System.out.print("R: ");
        r = input.nextInt();
        System.out.print("Centro - X: ");
        x = input.nextInt();
        System.out.print("Centro - Y: ");
        y = input.nextInt();

        Circulo a = new Circulo (r, x, y);

        a.MoveCentro(3,5);
    }
}
```

```

        System.out.println("1 - Inflar raio");
        System.out.println("2 - Desinflar raio");
        System.out.println("3 - Manter raio");
        op = input.nextInt();
        switch(op) {
            case 1:
                a.Infla(input.nextInt());
                break;
            case 2:
                a.Desinfla(input.nextInt());
                break;
            case 3:
                break;
            default:
                System.out.println("Opção inválida!");
                break;
        }

        area = a.CalculaArea();
        System.out.println("Área:" + area + " rad²");
    }
}

```

4) Escreva uma classe que represente um país. Um país é representado através dos atributos: código ISO 3166-1 (ex.: BRA), nome (ex.: Brasil), população (ex.: 193.946.886) e a sua dimensão em Km2 (ex.: 8.515.767,049). Escreva a classe em Java e forneça os seus membros a seguir:

- a) Construtor que inicialize o código ISO, o nome e a dimensão do país;
- b) Métodos de acesso (getter/setter) para as propriedades código ISO, nome, população e dimensão do país;
- c) Um método que retorne a densidade populacional do país

PAISES:

```

public class Paises{

    private String ISO;
    private String nome;
    private double populacao;
    private double area;

    public Paises(String ISO, String nome, double populacao, double area){
        this.ISO = ISO;
        this.nome = nome;
        this.populacao = populacao;
    }
}

```



```

        this.area = area;
    }

    public void setISO (String ISO){
        this.ISO = ISO;
    }

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

    public void setPopulacao (double populacao){
        this.populacao = populacao;
    }

    public void setArea (double area){
        this.area = area;
    }

    public String getISO{
        return this.ISO;
    }

    public String getNome{
        return this.nome;
    }

    public double getPopulacao{
        return this.populacao;
    }

    public double getArea{
        return this.area;
    }

    public double CalculaDensidade{ //pessoas/km2
        return (densidade = populacao / area);
    }

}

```

```

MAIN:
import java.util.Scanner;

public class Main{

    public static void main(String[] args){

        Scanner input = new Scanner(System.in);
        Países Brasil = new Países ("BRA", "Brasil", 207700000, 8516000);
        double densidade;

        densidade = Brasil.CalculaDensidade();
        System.out.println("Densidade demográfica: " + densidade + "
        pessoas/km²");

    }
}

```

5) Escreva a classe ConversaoDeUnidadesDeArea com métodos estáticos para conversão das unidades de área segundo a lista abaixo.

1 metro quadrado = 10.76 pés quadrados
1 pé quadrado = 929 centímetros quadrados
1 milha quadrada = 640 acres
1 acre = 43.560 pés quadrados

```

MAIN:
import java.util.Scanner;

public class Main{

    public static void main(String[] args){

        Scanner input = new Scanner(System.in);
        double in;
        int op;

        System.out.print("| Valor: ");
        in = input.nextInt();

        do{
            System.out.println("\n| 1- Converter m² para pés²      |");
            System.out.println("| 2- Converter pés² para cm²      |");
            System.out.println("| 3- Converter milha² para acres  |");
            System.out.println("| 4- Converter acres para pés²    |");
            System.out.println("| 0- Sair                        |");

```

```

        op = input.nextInt();

        switch(op){
            case 1:
                System.out.println("|          Resultado:          "          +
Conversor.mtoft(in) + " pés²");
                break;
            case 2:
                System.out.println("|          Resultado:          "          +
Conversor.fttocm(in) + " cm²");
                break;
            case 3:
                System.out.println("|          Resultado:          "          +
Conversor.mltoac(in) + " acres");
                break;
            case 4:
                System.out.println("|          Resultado:          "          +
Conversor.actoft(in) + " pés²");
                break;
            case 0:
                System.exit(0);
                break;
            default:
                System.out.println("| OPÇÃO INVÁLIDA! |");
                break;
        }

    } while(op!=0);
}

```

CONVERSOR:

```

public class Conversor{
    public static double out;

    public static double mtoft(double in){
        out = in * 10.67;
        return out;
    }

    public static double fttocm(double in){
        out = in * 929;
        return out;
    }
}

```

```

    }

    public static double mltoac(double in){
        out = in * 640;
        return out;
    }

    public static double actoft(double in){
        out = in * 43560;
        return out;
    }
}

```

6) Crie a classe Acionista, que possui os atributos cliente (*String*), numeroCotas (*int*) e valorCota (atributo de classe do tipo *float*). Crie um construtor que receba o cliente o o número de cotas, e um construtor estático que receba como parâmetro o valor da cota e inicialize o mesmo. Crie o métodos de acesso (get e set) para todos os atributos e o método toString().

MAIN:

```

import java.util.Scanner;

public class Main{

    public static void main(String[] args){

        Scanner input = new Scanner(System.in);

        Acionista Milton = new Acionista ("Milton", 100);

        Acionista.setValor(300);

        System.out.println(Milton.toString());
    }
}

```

ACIONISTA:

```

public class Acionista{
    private String cliente;
    private int cotas;
    private static float valor;

    public Acionista(String cliente, int cotas){
        this.cliente = cliente;
        this.cotas = cotas;
    }

    public void setCliente(String cliente){
        this.cliente = cliente;
    }

    public void setCotas(int cotas){
        this.cotas = cotas;
    }
}

```

```
public static void setValor(float val){
    valor = val;
}

public String getCliente(){
    return this.cliente;
}

public int getCotas(){
    return this.cotas;
}

public float getValor(){
    return this.valor;
}

public String toString() {
    return ("| Total: " + valor * cotas);
}
}
```