

Aula 8 - Revisão

P. O. O.

Prof. Grace



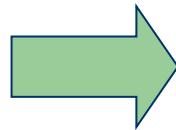
Listas de Exercícios

- POO_ListaEXE01_BasicoP1.pdf
- POO_ListaEXE02_ClassesP1.pdf
- Dia 15/04: Prova P1

Membros de uma Classe

Uma classe é composta por:

- Comportamentos ou operações
- Características ou atributos
(modelo)



- Métodos ou funções
- Dados ou variáveis
(implementação)

Atividade 12 – Classe Retângulo

- Modele e implemente uma classe que represente um Retângulo a partir de seus atributos: **base** e **altura**
- Ela deve ser capaz de instanciar objetos a partir de 2 parâmetros (base e altura)
- Caso base e altura não sejam informados, instancie um retângulo de base = 2 e altura = 1;
- Cada instância deve ser capaz de:
 - Devolver os valores de: Área; Perímetro; Base; Altura;
 - Informar se o objeto é um quadrado (boolean)
 - Imprimir todas as informações sobre o objeto instanciado
- Elabore o diagrama de Classes (jgrasp ou word ou qualquer editor).

Classe Retângulo

- Atributos
 - Base e altura (valores reais maiores que zero)
- Métodos
 - Construtor (base, altura);
 - Construtor (); // retângulo 2 x 1
 - Área: real
 - Perímetro: real
 - Configurar base(b)/ obter base: real
 - Configurar altura (h)/ obter altura: real
 - EhQuadrado(): booleano;

Solução

```
public class Retangulo {  
    private double base, altura;
```

```
    public Retangulo(){  
        setBase(2);  
        setAltura(1);  
    }
```

```
    public Retangulo(double b, double h){  
        if (b>0 && h>0) {  
            setBase(b);  
            setAltura(h);  
        }  
        else {  
            System.out.println("As dimensões devem ser maiores que zero!");  
            if (b<=0) setBase(2);  
            if (h<=0) setAltura(1);  
        }  
    }  
}
```

Atributos privados

Sobrecarga do método construtor

*Posso validar base e altura nos setters?
Porque validar aqui no construtor?*

Métodos set

```
public void setBase(double b)
{
    if (b>0)
        base = b;
    else
        System.out.println("A base deve ser maior que zero");
}

public void setAltura(double h)
{
    if (h>0)
        altura = h;
    else
        System.out.println("A altura deve ser maior que zero");
}
```

Setters

Não forçam para base ou altura = 1

Parâmetros X Atributos

- Posso usar parâmetro com mesmo identificador que atributo? Por exemplo:

```
public void setBase(double base)
{
    if (base>0)
        base = base;
    else
        System.out.println("A base deve ser maior que zero");
}
```

- Quem é o atributo e quem é o parâmetro?

Uso do operador this

- Usado para indicar que as informações (atributos ou métodos) do objeto instanciado estão sendo acessadas.
- No exemplo abaixo indica o acesso ao **atributo *base***, evitando confundir com o **parâmetro *base***;

```
public void setBase(double base)
{
    if (base>0)
        this.base = base;
    else
        System.out.println("A base deve ser maior que zero");
}
```

O mesmo pode ser feito no setAltura (altura)

Metodos get/ area/ perimetro

```
public double getBase()
{
    return base;
}
```

Getters

```
public double getAltura()
{
    return altura;
}
```

*Porque não usamos o this?
Poderíamos usar?*

```
public double area()
{
    return base * altura;
}
```

*Precisamos criar atributos
para armazenar Área e Perímetro?*

```
public double perimetro()
{
    return 2*(base + altura);
}
```

Atividade 11 – Média da Turma

- Altere o programa anterior, de modo que o mesmo calcule a média da turma.
- Durante a impressão das notas, além de exibir a média de cada aluno, deve ser informado se o aluno ficou, acima, abaixo ou na média da turma.
- Exemplo:
 - Supondo que a média da turma seja 8.2:
 - Aluno 1: 6.0 10.0 – Média: 8.0 – Abaixo da média
- Se necessário, altere as dimensões do *array*.

Sugestão

- Incluir + 1 coluna para armazenar média por aluno (notas [5][3])
- No 1º loop, já grava P1, P2 e calcula Média do aluno
- Criar uma variável para “acumular” a soma de todas as médias calculadas durante a entrada de dados
- Após a entrada de todas as notas, calcular a média da Turma ($\text{mediaTurma} = \text{somaMedias} / \text{numAlunos}$)
- Durante o loop de impressão, comparar a média de cada aluno com a Média da Turma.

Atividade 09 - Classe Data

- Altere a classe Data incluindo o método `bissexto()` que informa se o ano é bissexto ou não (devolve booleano);
- Utilize este método para validar a quantidade de dias de fevereiro dentro do método `checkDia()`;

Ativ. 09 - Ano Bissexto

```
private boolean bissexto(int anoTeste){
    if ((anoTeste%4==0 && anoTeste%100!=0) || (anoTeste%400 ==0))
        return true;
    return false;
}

private int checkDia (int diaTeste){

    int diasMes[]={0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
    if (bissexto(ano))
        diasMes[2] = 29;
    if (diaTeste > 0 && diaTeste <= diasMes[mes])
        return diaTeste;
    System.out.println ("Dia " + diaTeste + " inválido. Configurado dia = 1.");
    return 1;
}
```

Outra solução

```
public boolean bissexto(){
    if ((ano%4==0 && ano%100!=0) || (ano%400 ==0))
        return true;
    return false;
}

private int checkDia (int diaTeste){

    int diasMes[]={0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
    if (bissexto())
        diasMes[2] = 29;
    if (diaTeste > 0 && diaTeste <= diasMes[mes])
        return diaTeste;
    System.out.println ("Dia " + diaTeste + " inválido. Configurado dia = 1.");
    return 1;
}
```

Listas de Exercícios 2

- 1. Considere um objeto que represente um número inteiro positivo. Este objeto deve ter a capacidade de:
 - (a) Informar se o mesmo é ou não um número par.
 - (b) Informar se o mesmo é ou não um número primo.
 - (c) Fornecer o seu fatorial.

Codifique uma classe Java que represente este objeto e um programa para testá-la.

Classe “numero”

- Atributos: - num: int
- Métodos: + setNum (int n) //validar se é positivo
+ getNum: int
+ ehPar: boolean
+ ehPrimo: boolean
+ fat: int

Lista de Exercícios

2. Considere um objeto que tenha a capacidade de totalizar valores, isto é, um objeto que, a cada vez que receber uma mensagem para totalizar determinado valor, some este valor com o total que o mesmo contém até então.

Este objeto conterà sempre uma soma (um total), a qual **inicialmente será zero**. A cada vez que for solicitado a totalizar determinado valor, o objeto deverá somá-lo com seu total, obtendo um novo total.

Lista de Exercícios

- O objeto também deve ter a capacidade de, quando solicitado, informar qual o montante totalizado, bem como, qual a média dos valores já totalizados.

Acumulador
soma: real contagem: inteiro
totalize(valor) informeTotal() informeContagem() fornecaMedia()

Qual é o objetivo do atributo contagem???

Lista de Exercícios

```
public class acumulador{  
    private float soma = 0;  
    private int contagem = 0;  
  
    void totalize(float valor){  
        soma = soma + valor;  
        contagem++;  
    }  
    float informeTotal(){  
        return soma;  
    }  
}
```

Acumulador
soma: real contagem: inteiro
totalize(valor) informeTotal() informeContagem() fornecaMedia()

Lista de Exercícios

...

```
int informeContagem(){  
    return contagem;
```

```
float fornecaMedia(){  
    return soma/contagem;  
}
```

```
}
```

Acumulador
soma: real contagem: inteiro
totalize(valor) informeTotal() informeContagem() fornecaMedia()