



Java e Orientação a Objetos

Capítulo: Classes, atributos, métodos, static

<https://devsuperior.com.br>

Prof. Dr. Nelio Alves

1

Resolvendo um problema sem
orientação a objetos

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

2

Problema exemplo

Fazer um programa para ler as medidas dos lados de dois triângulos X e Y (suponha medidas válidas). Em seguida, mostrar o valor das áreas dos dois triângulos e dizer qual dos dois triângulos possui a maior área.

A fórmula para calcular a área de um triângulo a partir das medidas de seus lados a, b e c é a seguinte (fórmula de Heron):

$$area = \sqrt{p(p-a)(p-b)(p-c)} \quad \text{onde} \quad p = \frac{a+b+c}{2}$$

Exemplo:

```
Enter the measures of triangle X:
3.00
4.00
5.00
Enter the measures of triangle Y:
7.50
4.50
4.02
Triangle X area: 6.0000
Triangle Y area: 7.5638
Larger area: Y
```

3

```
package application;

import java.util.Locale;
import java.util.Scanner;

public class Program {

    public static void main(String[] args) {
        Locale.setDefault(Locale.US);
        Scanner sc = new Scanner(System.in);

        double xA, xB, xC, yA, yB, yC;

        System.out.println("Enter the measures of triangle X: ");
        xA = sc.nextDouble();
        xB = sc.nextDouble();
        xC = sc.nextDouble();
        System.out.println("Enter the measures of triangle Y: ");
        yA = sc.nextDouble();
        yB = sc.nextDouble();
        yC = sc.nextDouble();

        double p = (xA + xB + xC) / 2.0;
        double areaX = Math.sqrt(p * (p - xA) * (p - xB) * (p - xC));

        p = (yA + yB + yC) / 2.0;
        double areaY = Math.sqrt(p * (p - yA) * (p - yB) * (p - yC));

        System.out.printf("Triangle X area: %.4f\n", areaX);
        System.out.printf("Triangle Y area: %.4f\n", areaY);

        if (areaX > areaY) {
            System.out.println("Larger area: X");
        }
        else {
            System.out.println("Larger area: Y");
        }

        sc.close();
    }
}
```

4

Criando uma classe com três atributos para representar melhor o triângulo

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

5

Discussão

Triângulo é uma entidade com três atributos: a, b, c.

Estamos usando três variáveis distintas para representar cada triângulo:

double aX, bX, cX, aY, bY, cY;

Para melhorar isso, vamos usar uma CLASSE para representar um triângulo.

Memória:

3.00	4.00	5.00
aX	bX	cX
7.50	4.50	4.02
aY	bY	cY

6

Classe

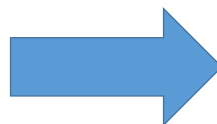
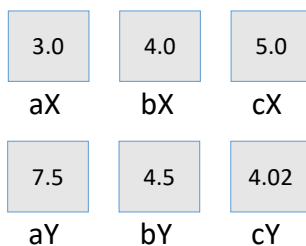
- É um tipo estruturado que pode conter (membros):
 - Atributos (dados / campos)
 - Métodos (funções / operações)
- A classe também pode prover muitos outros recursos, tais como:
 - Construtores
 - Sobrecarga
 - Encapsulamento
 - Herança
 - Polimorfismo
- Exemplos:
 - Entidades: Produto, Cliente, Triangulo
 - Serviços: ProdutoService, ClienteService, EmailService, StorageService
 - Controladores: ProdutoController, ClienteController
 - Utilitários: Calculadora, Compactador
 - Outros (views, repositórios, gerenciadores, etc.)

7

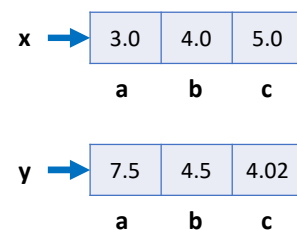
```
package entities;
```

```
public class Triangle {  
  
    public double a;  
    public double b;  
    public double c;  
}
```

```
double aX, bX, cX, aY, bY, cY;
```



```
Triangle x, y;  
x = new Triangle();  
y = new Triangle();
```



8

```

package application;

import java.util.Locale;
import java.util.Scanner;

import entities.Triangle;

public class Program {

    public static void main(String[] args) {

        Locale.setDefault(Locale.US);
        Scanner sc = new Scanner(System.in);

        Triangle x, y;
        x = new Triangle();
        y = new Triangle();

        System.out.println("Enter the measures of triangle X: ");
        x.a = sc.nextDouble();
        x.b = sc.nextDouble();
        x.c = sc.nextDouble();
        System.out.println("Enter the measures of triangle Y: ");
        y.a = sc.nextDouble();
        y.b = sc.nextDouble();
        y.c = sc.nextDouble();

        double p = (x.a + x.b + x.c) / 2.0;
        double areaX = Math.sqrt(p * (p - x.a) * (p - x.b) * (p - x.c));

        p = (y.a + y.b + y.c) / 2.0;
        double areaY = Math.sqrt(p * (p - y.a) * (p - y.b) * (p - y.c));
        (...)
    }
}

```

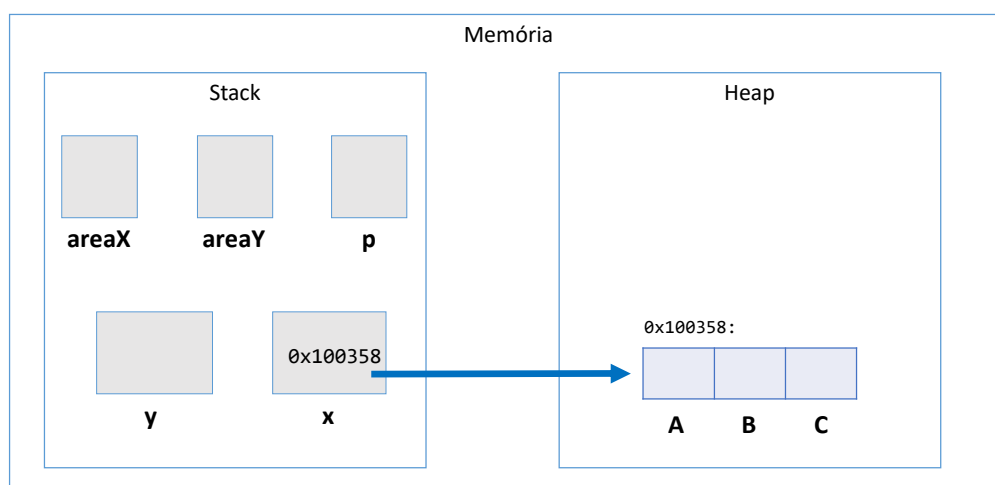
9

Instanciação

(alocação dinâmica de memória)

```
double areaX, areaY, p;
Triangle x, y;
```

```
x = new Triangle();
```

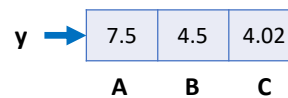
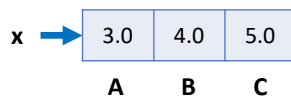


10

Classes, objetos, atributos

- Classe: é a definição do tipo
- Objetos: são instâncias da classe

```
package course;  
  
public class Triangle {  
  
    public double a;  
    public double b;  
    public double c;  
  
}
```



11

Criando um método para
obtermos os benefícios de
reaproveitamento e delegação

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

12

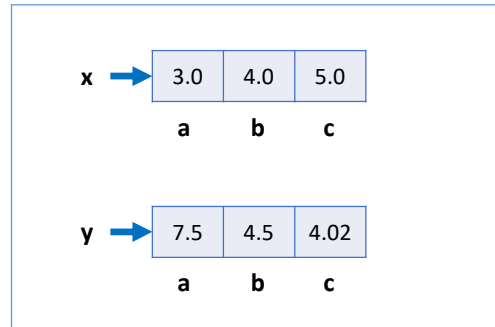
Discussão

Com o uso de CLASSE, agora nós temos uma variável composta do tipo "Triangle" para representar cada triângulo:

```
Triangle x, y;
x = new Triangle();
y = new Triangle();
```

Agora vamos melhorar nossa CLASSE, acrescentando nela um MÉTODO para calcular a área.

Memória:



13

```
package application;

import java.util.Locale;
import java.util.Scanner;

import entities.Triangle;

public class Program {

    public static void main(String[] args) {

        Locale.setDefault(Locale.US);
        Scanner sc = new Scanner(System.in);

        Triangle x, y;
        x = new Triangle();
        y = new Triangle();

        System.out.println("Enter the measures of triangle X: ");
        x.a = sc.nextDouble();
        x.b = sc.nextDouble();
        x.c = sc.nextDouble();
        System.out.println("Enter the measures of triangle Y: ");
        y.a = sc.nextDouble();
        y.b = sc.nextDouble();
        y.c = sc.nextDouble();

        double areaX = x.area();
        double areaY = y.area();
        (...)
```

14

```

package entities;

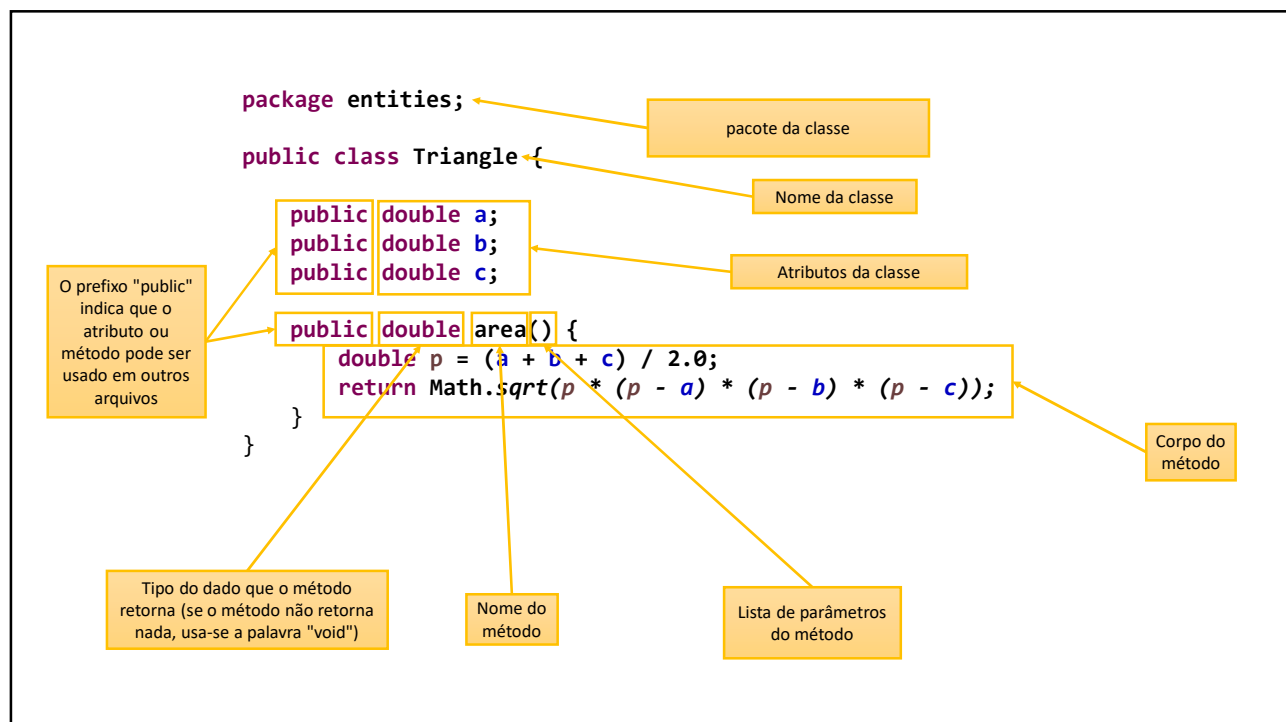
public class Triangle {

    public double a;
    public double b;
    public double c;

    public double area() {
        double p = (a + b + c) / 2.0;
        return Math.sqrt(p * (p - a) * (p - b) * (p - c));
    }
}

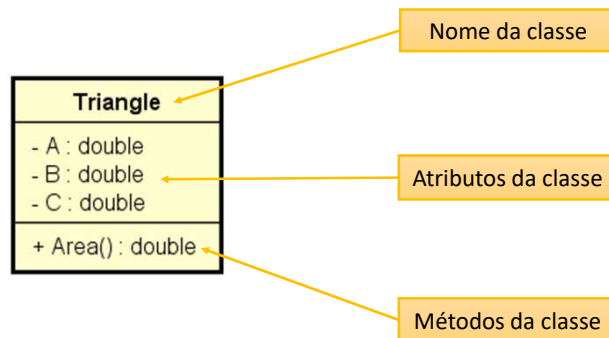
```

15



16

Projeto da classe (UML)



17

Discussão

Quais são os benefícios de se calcular a área de um triângulo por meio de um MÉTODO dentro da CLASSE Triangle?

1) Reaproveitamento de código: nós eliminamos o código repetido (cálculo das áreas dos triângulos x e y) no programa principal.

2) Delegação de responsabilidades: quem deve ser responsável por saber como calcular a área de um triângulo é o próprio triângulo. A lógica do cálculo da área não deve estar em outro lugar.

18

Começando a resolver um segundo problema exemplo

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

19

Outro exemplo

Fazer um programa para ler os dados de um produto em estoque (nome, preço e quantidade no estoque). Em seguida:

- Mostrar os dados do produto (nome, preço, quantidade no estoque, valor total no estoque)
- Realizar uma entrada no estoque e mostrar novamente os dados do produto
- Realizar uma saída no estoque e mostrar novamente os dados do produto

Para resolver este problema, você deve criar uma CLASSE conforme projeto ao lado:

(veja exemplo na próxima página)

Product
- Name : string - Price : double - Quantity : int
+ TotalValueInStock() : double + AddProducts(quantity : int) : void + RemoveProducts(quantity : int) : void

20

Example:

Enter product data:

Name: **TV**

Price: **900.00**

Quantity in stock: **10**

Product data: TV, \$ 900.00, 10 units, Total: \$ 9000.00

Enter the number of products to be added in stock: **5**

Updated data: TV, \$ 900.00, 15 units, Total: \$ 13500.00

Enter the number of products to be removed from stock: **3**

Updated data: TV, \$ 900.00, 12 units, Total: \$ 10800.00

21

Example:

Enter product data:

Name: **TV**

Price: **900.00**

Quantity in stock: **10**

Product data: TV, \$ 900.00, 10 units, Total: \$ 9000.00

Enter the number of products to be added in stock: **5**

Updated data: TV, \$ 900.00, 15 units, Total: \$ 13500.00

Enter the number of products to be removed from stock: **3**

Updated data: TV, \$ 900.00, 12 units, Total: \$ 10800.00

Product
- Name : string
- Price : double
- Quantity : int
+ TotalValueInStock() : double
+ AddProducts(quantity : int) : void
+ RemoveProducts(quantity : int) : void

22

Object e toString

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

23

Discussão

- Toda classe em Java é uma subclasse da classe Object
- Object possui os seguintes métodos:
 - getClass- retorna o tipo do objeto
 - equals - compara se o objeto é igual a outro
 - hashCode - retorna um código hash do objeto
 - toString - converte o objeto para string

24

```
package entities;

public class Product {

    public String name;
    public double price;
    public int quantity;

    public double totalValueInStock() {
        return price * quantity;
    }

    public void addProducts(int quantity) {
        this.quantity += quantity;
    }

    public void removeProducts(int quantity) {
        this.quantity -= quantity;
    }

    public String toString() {
        return name
            + ", $ "
            + String.format("%.2f", price)
            + ", "
            + quantity
            + " units, Total: $ "
            + String.format("%.2f", totalValueInStock());
    }
}
```

25

Finalizando o programa

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

26

```

package application;

import java.util.Locale;
import java.util.Scanner;

import entities.Product;

public class Program {
    public static void main(String[] args) {
        Locale.setDefault(Locale.US);
        Scanner sc = new Scanner(System.in);

        Product product = new Product();
        System.out.println("Enter product data: ");
        System.out.print("Name: ");
        product.name = sc.nextLine();
        System.out.print("Price: ");
        product.price = sc.nextDouble();
        System.out.print("Quantity in stock: ");
        product.quantity = sc.nextInt();

        System.out.println();
        System.out.println("Product data: " + product);

        System.out.println();
        System.out.print("Enter the number of products to be added in stock: ");
        int quantity = sc.nextInt();
        product.addProducts(quantity);

        System.out.println();
        System.out.println("Updated data: " + product);

        System.out.println();
        System.out.print("Enter the number of products to be removed from stock: ");
        quantity = sc.nextInt();
        product.removeProducts(quantity);

        System.out.println();
        System.out.println("Updated data: " + product);

        sc.close();
    }
}

```

27

Membros estáticos - PARTE 1

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves


28

Product
- Name : string - Price : double - Quantity : int
+ TotalValueInStock() : double + AddProducts(quantity : int) : void + RemoveProducts(quantity : int) : void

membros
=
atributos e métodos

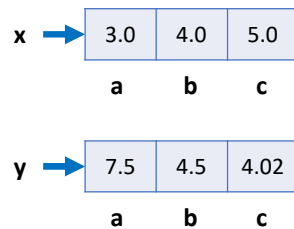
29

Membros estáticos

- Também chamados membros de classe
 - Em oposição a membros e instância
- São membros que fazem sentido independentemente de objetos. Não precisam de objeto para serem chamados. São chamados a partir do próprio nome da classe.
- Aplicações comuns:
 - Classes utilitárias  **Math.sqrt(double)**
 - Declaração de constantes
- Uma classe que possui somente membros estáticos, pode ser uma classe estática também. Esta classe não poderá ser instanciada.

30

```
Triangle x, y;
x = new Triangle();
y = new Triangle();
```



x.area() → **6.0**

y.area() → **7.5638**

31

Problema exemplo

Fazer um programa para ler um valor numérico qualquer, e daí mostrar quanto seria o valor de uma circunferência e do volume de uma esfera para um raio daquele valor. Informar também o valor de PI com duas casas decimais.

Exemplo:

```
Enter radius: 3.0
Circumference: 18.85
Volume: 113.10
PI value: 3.14
```

32

Checklist

- Versão 1: métodos na própria classe do programa
 - Nota: dentro de um método estático você não pode chamar membros de instância da mesma classe.
- Versão 2: classe Calculator com membros de instância
- Versão 3: classe Calculator com método estático

33

```
package application;

import java.util.Locale;
import java.util.Scanner;

public class Program {

    public static final double PI = 3.14159;

    public static void main(String[] args) {

        Locale.setDefault(Locale.US);
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter radius: ");
        double radius = sc.nextDouble();

        double c = circumference(radius);

        double v = volume(radius);

        System.out.printf("Circumference: %.2f\n", c);
        System.out.printf("Volume: %.2f\n", v);
        System.out.printf("PI value: %.2f\n", PI);

        sc.close();
    }

    public static double circumference(double radius) {
        return 2.0 * PI * radius;
    }

    public static double volume(double radius) {
        return 4.0 * PI * radius * radius * radius / 3.0;
    }
}
```

VERSÃO 1

34

```
package util;

public class Calculator {

    public final double PI = 3.14159;

    public double circumference(double radius) {
        return 2.0 * PI * radius;
    }

    public double volume(double radius) {
        return 4.0 * PI * radius * radius * radius / 3.0;
    }
}
```

VERSÃO 2

```
Calculator calc = new Calculator();

System.out.print("Enter radius: ");
double radius = sc.nextDouble();

double c = calc.circumference(radius);

double v = calc.volume(radius);

System.out.printf("Circumference: %.2f\n", c);
System.out.printf("Volume: %.2f\n", v);
System.out.printf("PI value: %.2f\n", calc.PI);
```

35

Membros estáticos - PARTE 2

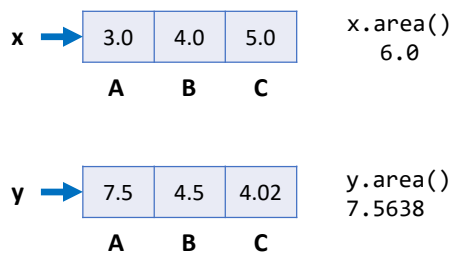
<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

36

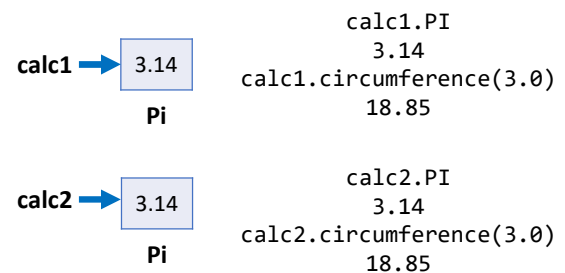
Discussão

- No problema dos triângulos, cada triângulo possui sua área.
- Area() é uma operação concernente ao objeto: cada triângulo possui sua área.



- Já no caso da calculadora, os valores dos cálculos não mudam para calculadoras diferentes, ou seja, são cálculos estáticos. O valor de Pi também é estático.

```
Calculator calc1 = new Calculator();
Calculator calc2 = new Calculator();
```



37

```
package util;

public class Calculator {

    public static final double PI = 3.14159;

    public static double circumference(double radius) {
        return 2.0 * PI * radius;
    }

    public static double volume(double radius) {
        return 4.0 * PI * radius * radius * radius / 3.0;
    }
}
```

```
System.out.print("Enter radius: ");
double radius = sc.nextDouble();

double c = Calculator.circumference(radius);

double v = Calculator.volume(radius);

System.out.printf("Circumference: %.2f\n", c);
System.out.printf("Volume: %.2f\n", v);
System.out.printf("PI value: %.2f\n", Calculator.PI);
```

VERSÃO 3

38