

Linguagem de Programação

ADS – Análise e Desenvolvimento de Sistemas
Prof. Vagner Macedo

Aula 10
Classes OO

Discussão

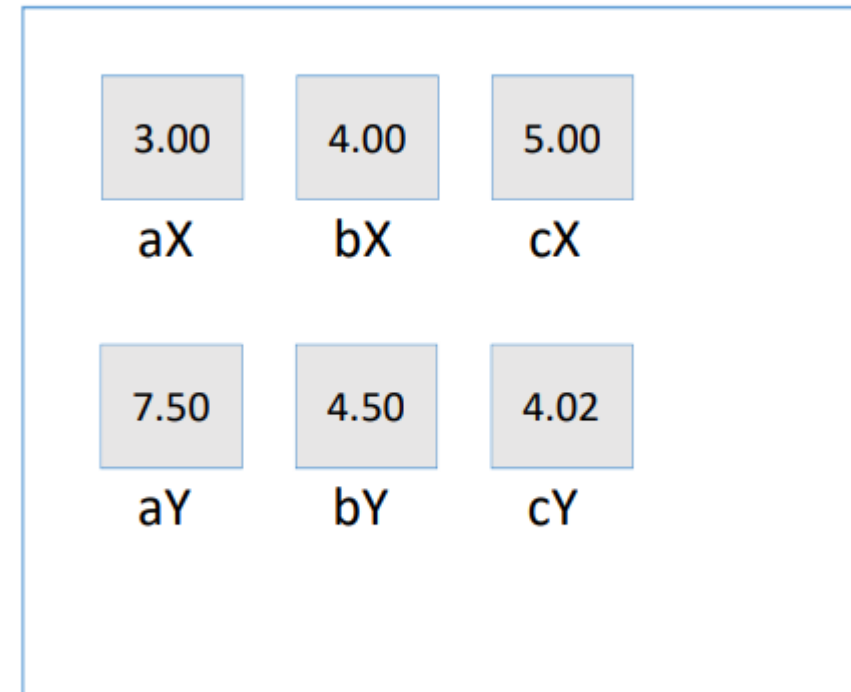
Triângulo é uma entidade com três atributos: a, b, c.

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:



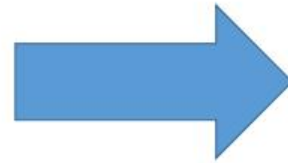
Classe

- 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.)

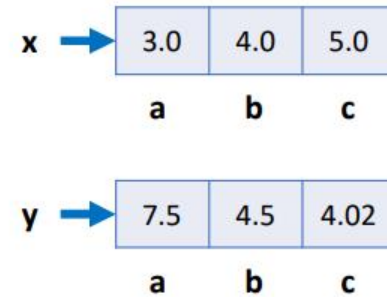
Classe Triangle

```
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();
```



Classe Program

```
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));
        (...)
```

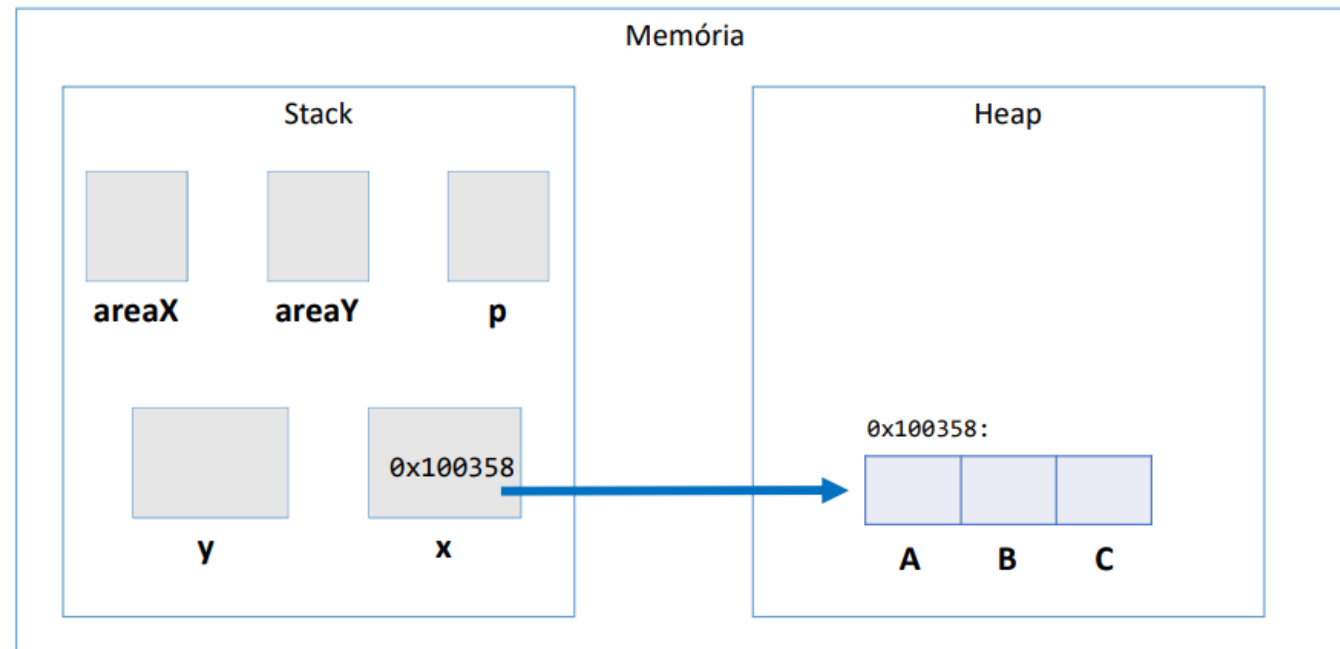
Objeto

Instanciação

(alocação dinâmica de memória)

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

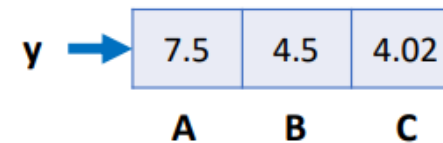
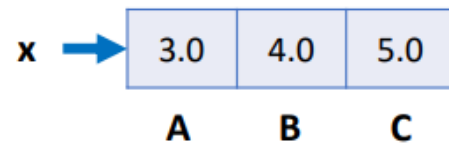
```
x = new Triangle();
```



Classes, objetos e 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;  
  
}
```



Criando um método para obtermos os benefícios de reaproveitamento e delegação

```
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();
        (...)
```


Criando um método para obtermos os benefícios de reaproveitamento e delegação

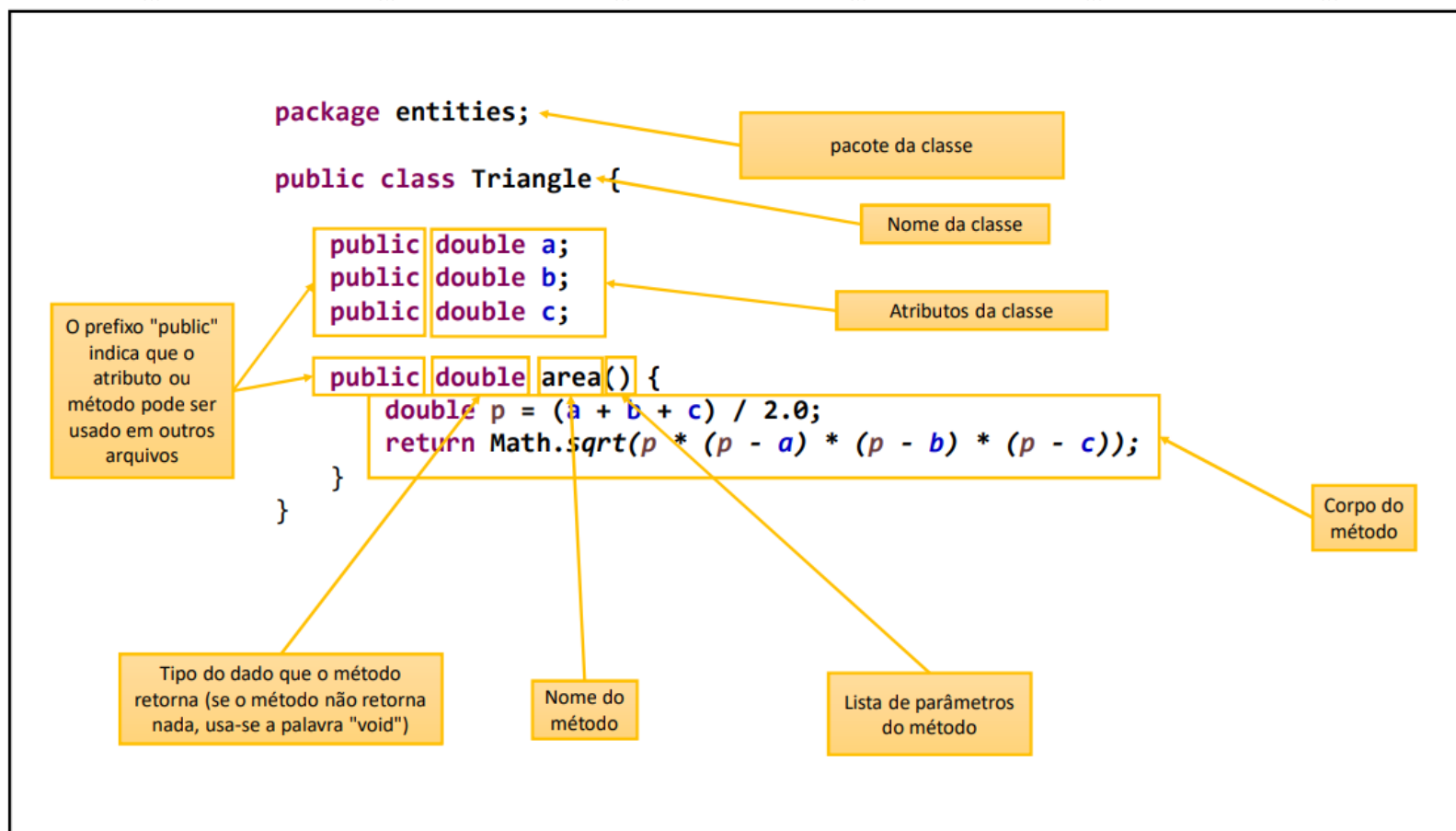
```
package entities;

public class Triangle {

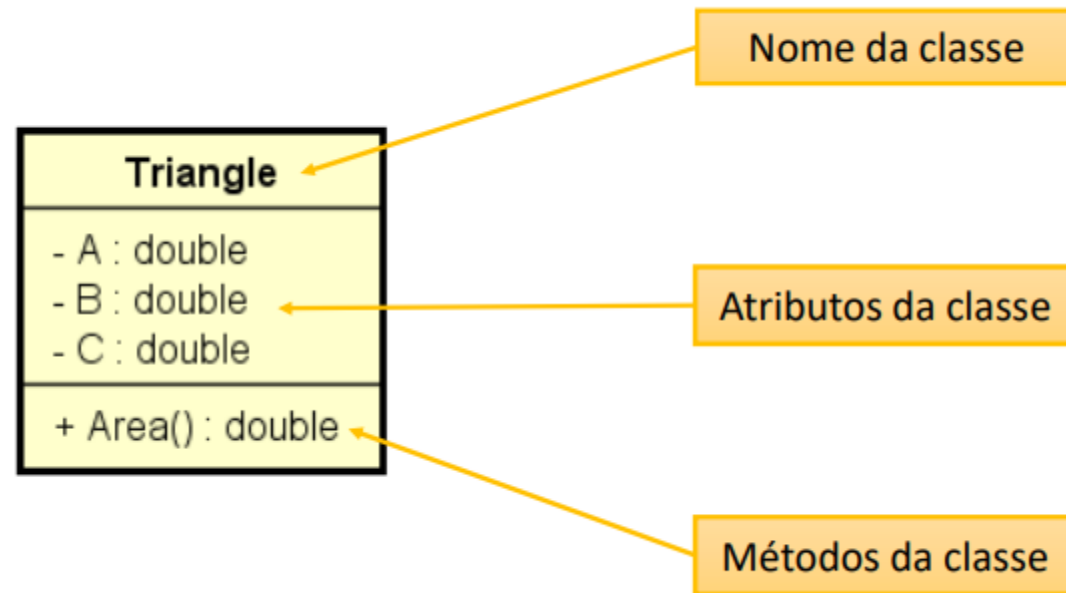
    public double a;
    public double b;
    public double c;

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

Detalhes da classe



Projeto da Classe (UML)



Obrigado!

