

Linguagem de Programação II

Prof. Dr. Gustavo Leitão

Repetição

```
public class Main {  
    public static void main(String[] args){  
        for (int i = 0; i < args.length ; i++){  
            System.out.println(args[i]);  
        }  
    }  
}
```

Repetição

```
public class Main {  
    public static void main(String[] args){  
        int a = 1;  
        while (a % 2 != 0){  
            a = (int) (Math.random() * 1000);  
        }  
  
        System.out.println(a);  
    }  
}
```

```
public class Main {  
    public static void main(String[] args){  
        int a;  
        do{  
            a = (int) (Math.random() * 1000);  
        }while (a % 2 != 0);  
  
        System.out.println(a);  
    }  
}
```


String

Tipo de dado não primitivo que permite armazenar textos

```
public class Main {  
    public static void main(String[] args){  
        String nome = "Hello LP2";  
    }  
}
```

Atributos

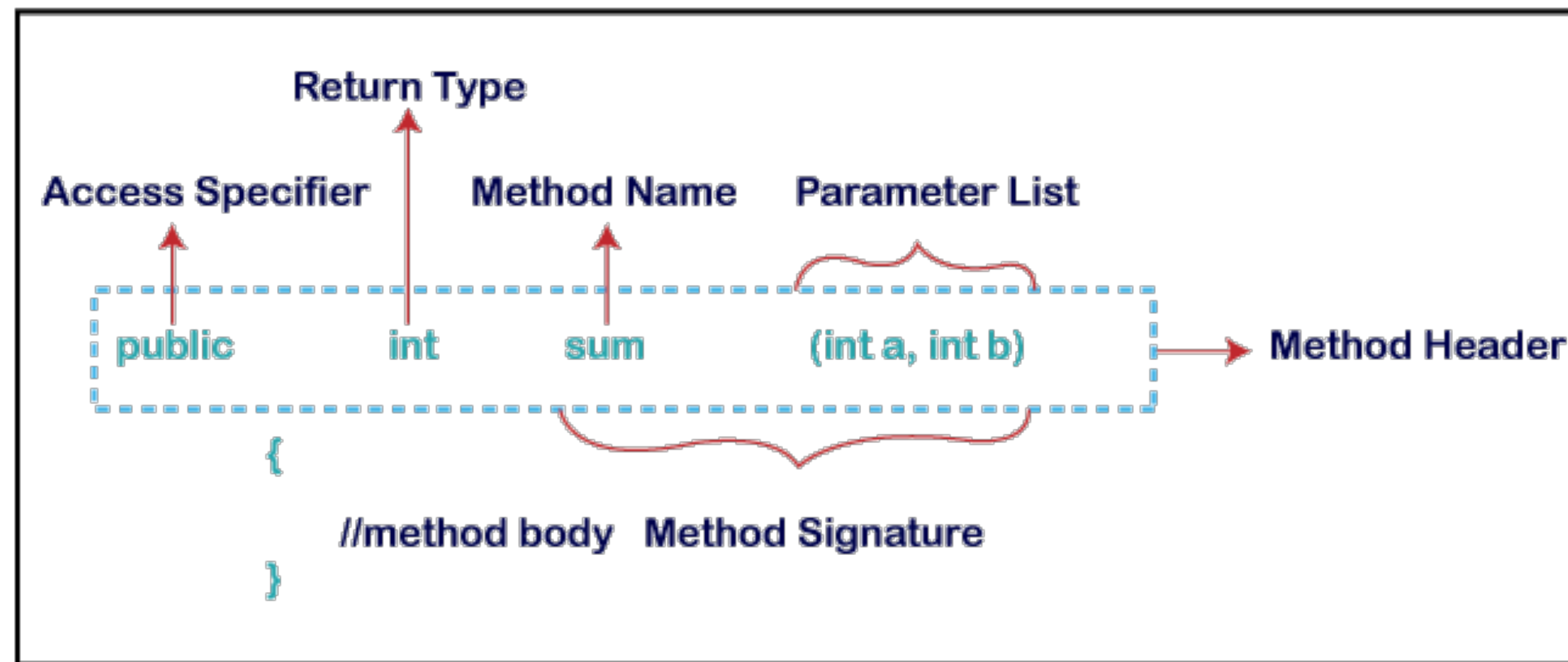
São propriedades semelhantes que os objetos de uma classe possuem.

```
package geo;  
  
public class Circle {  
     private double raio;  
  
}
```

Métodos

Um método é um bloco de código ou coleção de instruções ou um conjunto de códigos agrupados para executar uma determinada tarefa ou operação. É usado para alcançar a reutilização do código. Nós escrevemos um método uma vez e o usamos várias vezes.

Method Declaration



Métodos

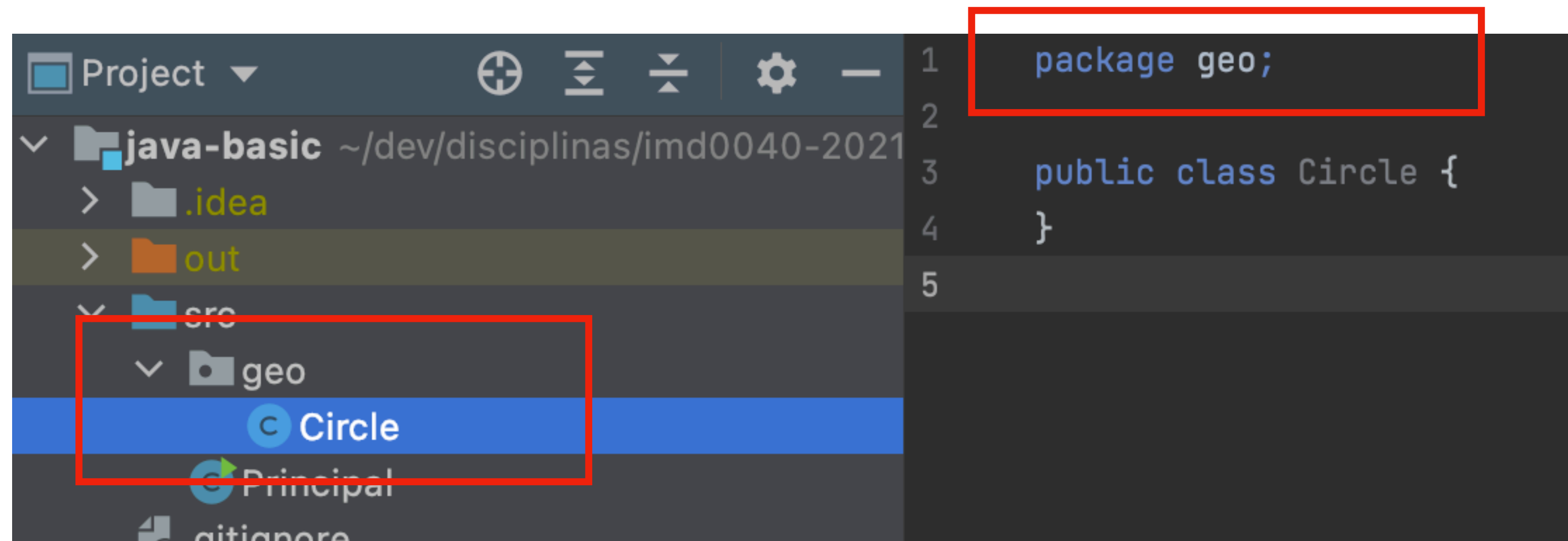
Um método é um bloco de código ou coleção de instruções ou um conjunto de códigos agrupados para executar uma determinada tarefa ou operação. É usado para alcançar a reutilização do código. Nós escrevemos um método uma vez e o usamos várias vezes.

- **Público:** O método é acessível por todas as classes quando usamos o especificador público em nosso aplicativo.
- **Privado:** O método está acessível apenas nas classes em que está definido.
- **Protegido:** O método é acessível no mesmo pacote ou subclasses em um pacote diferente.
- **Padrão:** quando não usamos nenhum especificador de acesso na declaração do método, o Java usa o especificador de acesso padrão por padrão. É visível apenas do mesmo pacote.

```
public class Circle {  
    public double area(int raio){  
        return 3.14 * raio * raio;  
    }  
}
```

Pacotes

Um pacote ou **package** na tecnologia Java nada mais é do que um conjunto de classes localizadas na mesma estrutura hierárquica de diretórios.



Construtor

O construtor é definido como um método cujo nome deve ser o mesmo nome da classe e sem indicação do tipo de retorno -- nem mesmo void. O construtor é unicamente invocado no momento da criação do objeto através do operador **new**.

```
1  package geo;
2
3  public class Circle {
4
5      Circle(){
6          |
7      }
8
9      public double area(int raio){
10         return 3.14 * raio * raio;
11     }
12
13 }
14
```

Criando Instâncias (Objetos)

```
var a = new Circle();
```

O Operador new é o responsável por criar um objeto a partir de uma classe

String

Comparação de Strings

```
public class Main {  
    public static void main(String[] args) {  
        String a = new String(original: "oi");  
        String b = new String(original: "oi");  
        System.out.println(a == b);  
        System.out.println(a.equals(b));  
    }  
}
```

<https://docs.oracle.com/javase/7/docs/api/java/lang/String.html>

Entrada teclado

```
public class Main {  
    public static void main(String[] args){  
        var scanner = new Scanner(System.in);  
        System.out.print("Digite um número: ");  
        String input = scanner.next();  
        System.out.println("Você digitou "+input);  
    }  
}
```

Atributos

Class Definition

Circle
-radius:double=1.0 -color:String="red"
+getRadius():double +getColor():String +getArea():double

Instances

<u>c1:Circle</u>
-radius=2.0 -color="blue"
+getRadius() +getColor() +getArea()

<u>c2:Circle</u>
-radius=2.0 -color="red"
+getRadius() +getColor() +getArea()

<u>c3:Circle</u>
-radius=1.0 -color="red"
+getRadius() +getColor() +getArea()