

# Programação Orientada a Objetos

Aula 2 – Professor Diogo Orlando Nunes de Almeida

# Acessando um compilador online:

- Nesse inicio da disciplina iremos usar um compilador online:
- [https://www.onlinegdb.com/online\\_java\\_compiler](https://www.onlinegdb.com/online_java_compiler)
- Futuramente e se necessário vamos para o Eclipse.

# O que é um objeto

- Os objetos são criados a partir de classes. A classe descreve o tipo do objeto; os objetos representam instanciações de classe (KOLLING, BARNES, 2008, p.2).
- Objetos ou, mais precisamente, as classes de onde os objetos vêm são essencialmente componentes reutilizáveis de software. (DEITEL, DEITEL, 2017, p.8).

# Declarar contra instanciar

- Declarar uma variável é inserir diretamente na memória uma variável.
- Instanciar uma variável é criar um objeto variável que estará conectada a uma classe.

# Declarar contra instanciar



```
a = 7.5  
a.is_integer()
```



False

- As variáveis em Python são instâncias de objetos.
- É declarado a variável “a” onde podemos acessar o método `is_integer()`
- `is_integer()` retorna se a variável é um inteiro ou não.

# Declarar contra instanciar

The screenshot shows a Java code editor with a file named Main.java. The code attempts to call a method is\_integer() on an integer variable 'a'. The IDE displays a compilation error message: "Compilation failed due to following error(s)." followed by the specific error: "Main.java:7: error: int cannot be dereferenced a.is\_integer(); ^". A status bar at the bottom indicates "1 error".

```
Main.java  :  
1 public class Main  
2 {  
3     public static void main(String[] args) {  
4         int a = 5;  
5         a.is_integer();  
6         System.out.println("Hello World");  
7     }  
8 }  
input  
Compilation failed due to following error(s).  
Main.java:7: error: int cannot be dereferenced  
        a.is_integer();  
          ^  
1 error
```

- Ao tentar chamar um método em Java de um inteiro, ocorre um erro afirmando que int não pode ser deixar de ser referenciada.
- Cópia e referência é um conteúdo que será vista em outra disciplina, mas são conceitos de variáveis que são declaradas na memória.

# Declarar contra instanciar

```
1 public class Main
2 {
3     public static void main(String[] args) {
4         String a = "texto teste";
5         System.out.println(a.length());
6     }
7 }
8 
```

...Program finished with exit code 0  
Press ENTER to exit console. □

- Já o tipo String em Java é instanciado, pois podemos acessar métodos da função sem nenhum problema.

# Tipos primitivos em Java

| Nome do tipo     | Descrição                               | Mínimo               | Máximo                 |
|------------------|---|----------------------|------------------------|
| Números inteiros |   |                      |                        |
| byte             | Inteiro de 1 byte (8 bits)              | -128                 | 127                    |
| short            | Inteiro curto (16 bits)                 | -32768               | 32767                  |
| int              | Inteiro (32 bits)                       | -2147483648          | 2147483647             |
| long             | Inteiro longo (64 bits)                 | -9223372036854775808 | 9223372036854775807    |
| Números reais    |   |                      |                        |
| float            | Ponto flutuante de precisão simples.    | 1.4e-45              | 3.4028235e38           |
| double           | Ponto flutuante de precisão duplo.      | 4.9e-324             | 1.7976931348623157e308 |
| char             | Um único caractere (16 bits)            |                      |                        |
| boolean          | Um valor booleano (verdadeiro ou falso) |                      |                        |

# Classes

- Classes são compostas de:
  - Atributos: são as propriedades de uma classe.
  - Métodos: são as funções que vão manipular esses atributos.

# Classes

```
Main.java :  
1 class Triangulo  
2 {  
3     int ponto1_x;  
4     int ponto1_y;  
5     int ponto2_x;  
6     int ponto2_y;  
7     int ponto3_x;  
8     int ponto3_y;  
9  
10    public void imprimePontos(){  
11        System.out.println(ponto1_x + " " + ponto1_y);  
12        System.out.println(ponto2_x + " " + ponto2_y);  
13        System.out.println(ponto3_x + " " + ponto3_y);  
14    }  
15}  
16  
17 public class Main  
18 {  
19     public static void main(String[] args) {  
20         Triangulo t1 = new Triangulo();  
21  
22         t1.ponto1_x = 50;  
23         t1.ponto1_y = 100;  
24  
25         t1.imprimePontos();  
26     }  
27 }
```

The screenshot shows a Java code editor with the file 'Main.java' open. The code defines a class 'Triangulo' with three integer attributes for points and a method to print them. It also defines a class 'Main' with a static 'main' method that creates an instance of 'Triangulo', sets its point coordinates, and calls its printing method. The output window at the bottom shows the printed coordinates: 50 100, 0 0, and 0 0.

```
50 100  
0 0  
0 0
```

- Foi criado a classe triângulo com seus três pontos bidimensionais como atributos.
- Foi criado o método para imprimir o par de pontos em cada linha desse triangulo.

# Classes

```
1 class Poligono
2 {
3     int[] ponto_x;
4     int[] ponto_y;
5     public int soma_x(){
6         int soma = 0;
7         for (int i = 0; i < ponto_x.length; i++){
8             soma += ponto_x[i];
9         }
10        return soma;
11    }
12    public int soma_y(){
13        int soma = 0;
14        for (int i = 0; i < ponto_y.length; i++){
15            soma += ponto_y[i];
16        }
17        return soma;
18    }
19 }
```

- Usando a classe polígono, será explicado alguns conceitos importantes de JAVA para a disciplina.

# Classes

```
1 class Poligono
2 {
3     int[] ponto_x;
4     int[] ponto_y;
5     public int soma_x(){
6         int soma = 0;
7         for (int i = 0; i < ponto_x.length; i++){
8             soma += ponto_x[i];
9         }
10        return soma;
11    }
12    public int soma_y(){
13        int soma = 0;
14        for (int i = 0; i < ponto_y.length; i++){
15            soma += ponto_y[i];
16        }
17        return soma;
18    }
19 }
```

- Para criar um vetor em JAVA, ele deve ser criado com um único tipo, usando o tipo e abre e fecha colchetes.
- Existe outras formas de criar um vetor de diversos tipos, como arrays de Objects, Listas e HashMap que não serão vistos nesta disciplina, mas tenha em mente que a complexidade é extrema devido ao casting de tipos.

# Classes

```
1 class Poligono
2 {
3     int[] ponto_x;
4     int[] ponto_y;
5     public int soma_x(){
6         int soma = 0;
7         for (int i = 0; i < ponto_x.length; i++){
8             soma += ponto_x[i];
9         }
10        return soma;
11    }
12    public int soma_y(){
13        int soma = 0;
14        for (int i = 0; i < ponto_y.length; i++){
15            soma += ponto_y[i];
16        }
17        return soma;
18    }
19 }
```

- O for em JAVA se faz de forma bem parecida com o C:
  - Cria uma variável iteradora int i, que vai mostrar qual é o índice onde o for vai iniciar.
  - A condição de parada.
  - A condição de incremento.

# Classes

```
1 class Poligono
2 {
3     int[] ponto_x;
4     int[] ponto_y;
5     public int soma_x(){
6         int soma = 0;
7         for (int i = 0; i < ponto_x.length; i++){
8             soma += ponto_x[i];
9         }
10        return soma;
11    }
12    public int soma_y(){
13        int soma = 0;
14        for (int i = 0; i < ponto_y.length; i++){
15            soma += ponto_y[i];
16        }
17        return soma;
18    }
19 }
```

- Para criar uma função ou método com retorno em JAVA, deve-se mostrar no inicio da função ou método o que ela vai retornar.
- Se eu crio a função int soma\_x() ela vai retornar int, se eu crio a função String palavras\_com\_s() ela vai retornar uma string.

# Classes

```
21 public class Main {  
22     public static void main(String[] args) {  
23         Poligono p1 = new Poligono();  
24  
25         p1.ponto_x = new int[] {30, 40, 120};  
26         p1.ponto_y = new int[] {100, 75, 150};  
27  
28         System.out.println(p1.soma_x());  
29         System.out.println(p1.soma_y());  
30     }  
31 }
```

- Usando a variável do tipo Poligono da classe recém criada, instanciamos um objeto Poligono chamado p1 usando a palavra new.
- Para enviar ao atributo da classe uma lista de inteiros deve-se usar a palavra new int[] {...}; sendo que entre chaves terá os vetores.

# Exercícios

- Crie uma classe que ao receber os atributos da cinemática (exceto cinemática circular) terão os métodos que irão fazer as contas.
- Crie uma classe que ao receber um vetor, vai retornar a média, o máximo, mínimo, somatório e produtório.

# Referências

- KÖLLING, M.; BARNES, D. J. **Programação orientada a objetos com Java**: uma introdução prática usando o Bluej. 4. ed. São Paulo, SP: Pearson, 2008. E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 23 fev. 2025.
- DEITEL, P. J.; DEITEL, H. M. **Java: como programar**. 10. ed. São Paulo, SP: Pearson, 2017. E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 23 fev. 2025.