

# Computação II – Orientação a Objetos

---

Fabio Mascarenhas - 2016.1

<http://www.dcc.ufrj.br/~fabiom/java>

# Introdução

---

- Esse não é um curso de Java!
- O objetivo é aprender os conceitos da programação orientada a objetos, e sua aplicação em programas reais
- A linguagem Java é só um veículo para isso
- Cuidado: programar em Java não é necessariamente programar de maneira OO

# Java não-OO

---

- Vamos nos concentrar nas partes OO de Java, mas ela também tem tudo que vocês já estão acostumados com C:
  - Variáveis locais e globais, funções, números, entrada e saída no console...
- Até a sintaxe é muito parecida
- Programas C simples podem ser reescritos em Java com pequenas alterações, mas **não são exemplos de programação OO!**

# De C para "Java com sabor de C"

```
#include <stdio.h>
```

bibliotecas

parâmetros

```
void geraConjunto(double vetor[], int tamanho, double inicial) {  
    for (int i = 0; i < tamanho; i++) {  
        vetor[i] = inicial;  
        inicial *= 2;  
    }  
}
```

→ função

} laços

string escrita em vetor

```
int main(int argc, char *argv[])
```

→ função

```
{  
    double vetor[5];  
    int i;
```

vetores

variáveis locais

```
puts("Este programa gera um vetor de numeros inteiros.\n");  
puts("Entre com o numero inicial do conjunto. ");  
scanf("%lf", &num);
```

SAÍDA  
ENTRADA

```
/* Geracao do conjunto */  
geraConjunto(vetor, 5, num);
```

chamada de função

```
/* Impressao do conjunto */  
for (i = 0; i < 5; i++)  
    printf("Elemento %d = %lf\n", i, vetor[i]);
```

} laços

```
return 0;  
}
```

leitura em vetor

# Faça você mesmo!

```
#include <stdio.h>
```

```
int main(int argc, char *argv[]) {
    int vetor[5], i;
    int trocou = 0;
    int fim = 5;
    int temp;
    printf("Entre com um vetor de %d elementos\n", 5);
    for (i = 0; i < 5; i++) {
        printf("Elemento %d ", i);
        scanf("%d", &vetor[i]);
    }
    do {
        trocou = 0;
        for (i=0; i < fim-1; i++) {
            if (vetor[i] > vetor[i+1]) {
                temp = vetor[i];
                vetor[i] = vetor[i+1];
                vetor[i+1] = temp;
                trocou = 1;
            }
        }
        fim--;
    } while (trocou);
    for (i=0; i < 5; i++) printf("%d\n", vetor[i]);
    return 0;
}
```

*Handwritten notes:*  
- A red arrow points from the word "ordenar" at the top left to the bubble sort algorithm.  
- Red circles with numbers 1, 2, and 3 are placed around the input loop.  
- A red circle with the number 1 is placed around the final printf statement.  
- A red note says:  $trocou \leq 1$   
- Another red note says:  $trocou \neq 0$

```
import java.util.*;
```

```
public class IniVetor {
    public static Scanner entrada = new Scanner(System.in);
    // public static -> função global ou variável global
    public static void geraConjunto(double[] vetor,
        int tamanho, double inicial) {
        for(int i = 0; i < tamanho; i++) {
            vetor[i] = inicial;
            inicial *= 2;
        }
    }
    public static void main(String[] args) {
        double[] vetor = new double[5]; // criação do vetor
        double num;
        int i;
        System.out.println
            ("Este programa gera um vetor de números inteiros.");
        System.out.println
            ("Entre com o número inicial do conjunto.");
        num = entrada.nextInt(); // chamada de método (00)
        geraConjunto(vetor, 5, num);
        for(i = 0; i < 5; i++)
            System.out.println("Elemento " + i +
                " = " + vetor[i]);
    }
}
```

*Handwritten notes:*  
- A red circle with the number 1 is placed around the first for loop in the main method.  
- A red circle with the number 2 is placed around the geraConjunto method call.