

Programação Orientada a Objetos - POOS3

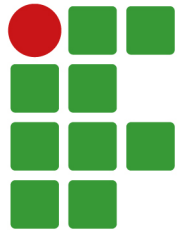
1

Tecnologia em Análise e Desenvolvimento de Sistemas

Aula 4

Utilização de arrays em Java

2º semestre de 2018



INSTITUTO FEDERAL

São Paulo


Câmpus Araraquara

Arrays

- Um **array** é um conjunto de variáveis do mesmo tipo, referenciado por um nome comum e acessados por meio de sua posição (índice) .

Exemplo

- Implementar um programa que leia 10 idades e após a leitura informe, para cada idade lida, se a mesma está acima, abaixo ou na média.



Vamos criar 10 variáveis inteiras :-D

Sim, 10 variáveis inteiras organizadas em um array!!!



```

public class Pessoa {
    public static final int MAXIMO = 10;
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int idades[] = new int[Pessoa.MAXIMO], i;
        double media = 0;
        System.out.println("Informe 10 idades: ");
        for(i=0; i<Pessoa.MAXIMO; i++){
            idades[i] = input.nextInt();
            media += idades[i];
        }
        media /= Pessoa.MAXIMO;
        for(i=0; i<Pessoa.MAXIMO; i++){
            if(idades[i] > media)
                System.out.println("Idade " + idades[i] + " está acima da média");
            else
                if (idades[i] < media)
                    System.out.println("Idade " + idades[i] +
                        " está abaixo da média");
                else
                    System.out.println("Idade " + idades[i] +
                        " está exatamente na média");
        }
    }
}

```

Para declararmos um array definimos o tipo de dado armazenado e a quantidade de posições.

Podemos ter array de qualquer tipo de dado, inclusive arrays de objetos.

Para termos acesso a cada uma das variáveis declaradas dentro do array utilizamos o índice deste array.



Orientado a Objetos, programa não está.

```
public class Pessoa {  
    private int idade;  
  
    public Pessoa(int idade) {  
        setIdade(idade);  
    }  
  
    public int getIdade() {  
        return idade;  
    }  
  
    public void setIdade(int idade) {  
        this.idade = idade >= 0 ? idade : 0 ;  
    }  
}
```

```

public class Main {
    public static final int MAXIMO = 10;
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        Pessoa pessoas[] = new Pessoa(Main.MAXIMO);
        Pessoa p;
        int i;        double media=0;

        System.out.println("Informe 10 idades : ");
        for(i=0; i<MAXIMO; i++){
            //p = new Pessoa(input.nextInt());
            //pessoas[i] = p;
            pessoas[i] = new Pessoa(input.nextInt());
            media += pessoas[i].getIdade();
        }
        media /= MAXIMO;
        for(i=0; i < MAXIMO; i++){
            if(pessoas[i].getIdade() > media)
                System.out.println("Idade " + pessoas[i].getIdade() + " está acima da média");
            else
                if (pessoas[i].getIdade() < media)
                    System.out.println("Idade " + pessoas[i].getIdade() + " está abaixo da média");
                else
                    System.out.println("Idade " + pessoas[i].getIdade() +
                        " está exatamente na média");
        }
    }
}

```

Como pretendo armazenar objetos define-se a classe como tipo de dados do array.

Diferente de um tipo primitivo, é necessário que um novo objeto seja instanciado para a posição do array. Pode-se instanciar um objeto e atribuí-lo ao array ou mesmo instanciar o objeto dentro do array.

Como cada posição do array possui um objeto, é possível chamar os métodos do objeto que está no array.

Atenção



7

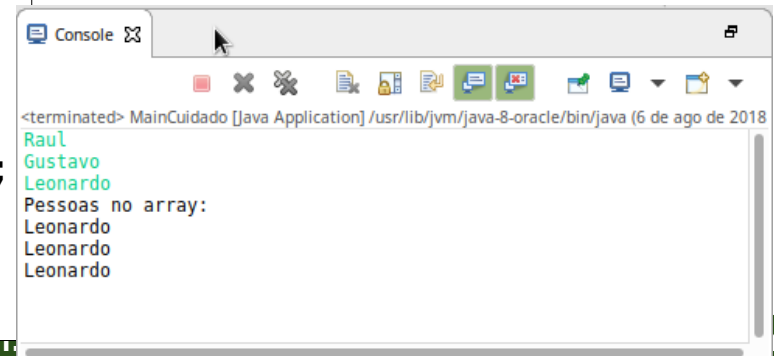
```
public class MainCuidado {  
    public static final int MAXIMO = 10;  
  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
        Person pessoas[] = new Person[MAXIMO];  
        Person p, p2;  
        int i;  
        p = new Person();  
        p2 = new Person();  
        for(i=0; i<MAXIMO; i++){  
            p.setNome(input.nextLine());  
            pessoas[i] = p;  
        }  
        System.out.println("Pessoas no array:");  
        for(i=0; i<MAXIMO; i++){  
            System.out.println(pessoas[i].getNome());  
        }  
    }  
}
```

Memória



Pessoa : p
p.nome = "Pio"

Pessoa : p2
p.nome = ""



Outro exemplo

- Pode-se utilizar array dentro de uma classe.
- Analisemos um exemplo onde uma loja que possui até 4 vendedores quer saber qual o vendedor que teve maior valor em vendas acumuladas no período de um ano.

Classe Vendedor

```
package model;

public class Vendedor {
    public static final int MESES = 12;

    private String nome;
    private double vendasMes[];

    public Vendedor(String nome) {
        this.nome = nome;
        vendasMes = new double[MESES];
    }

    public void setVendasMes(double valor, int mes) {
        vendasMes[mes] = valor;
    }

    //...
}
```

Classe Loja

```
package model;

public class Loja {
    private final int TOTAL_VENDEDORES = 4;
    private String razaoSocial;
    private Vendedor[] vendedores;
    private int vendedoresCadastrados;

    public Loja(String razaoSocial) {
        this.razaoSocial = razaoSocial;
        vendedores = new Vendedor[TOTAL_VENDEDORES];
        vendedoresCadastrados = 0;
    }
}
```



- **Leitura Obrigatória**

- Winder, R.; Roberts, G. Desenvolvendo Software em Java. 3 ed. Rio de Janeiro: LTC, 2009.
 - Capítulos 4.1 e 4.2

- **Videoaulas**

- <https://www.youtube.com/watch?v=Kc2RXH7dW-M>
- <https://www.youtube.com/watch?v=HxRb5KLofcl>
- <https://www.youtube.com/watch?v=UN6ZJBSTR0M>
- <https://www.youtube.com/watch?v=Es42H9OrDyl>