

Coleções de Objetos

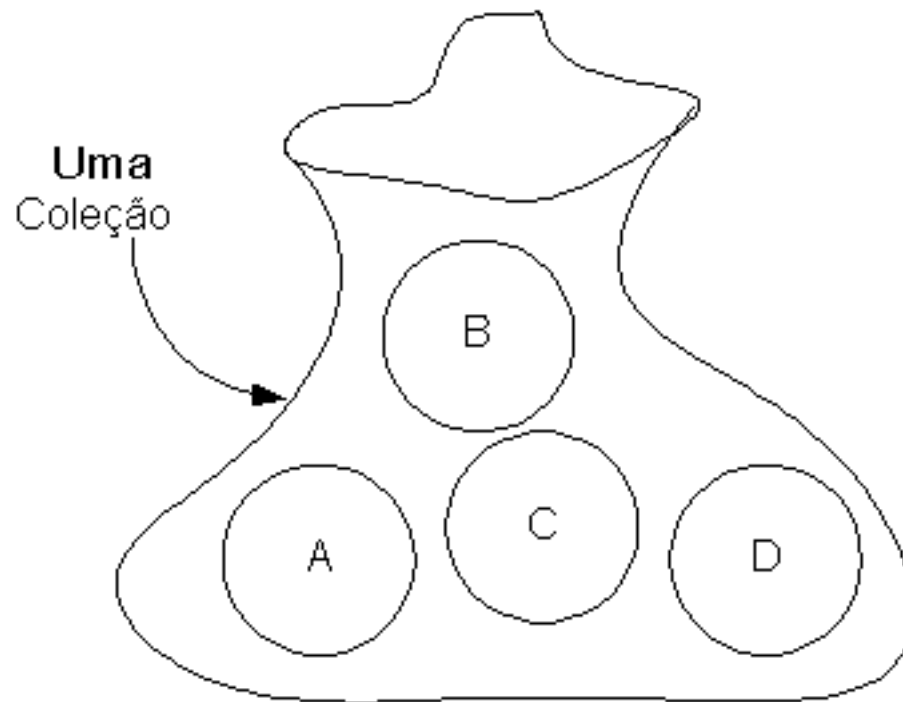
MATA55 – Programação O.O



Coleção de Objetos

- Em diversas situações desejamos lidar com vários objetos. Por exemplo
 - Alunos de um mesmo curso
 - Professores de um Departamento
 - Filmes de um provedor de mídias

Coleção de Objetos





Coleções de Objetos

- Composições podem envolver uma coleção de objetos
 - Uma turma é composta por um conjunto de alunos
 - Um histórico escolar de um determinado aluno possui uma lista de disciplinas ordenadas por semestres cursados



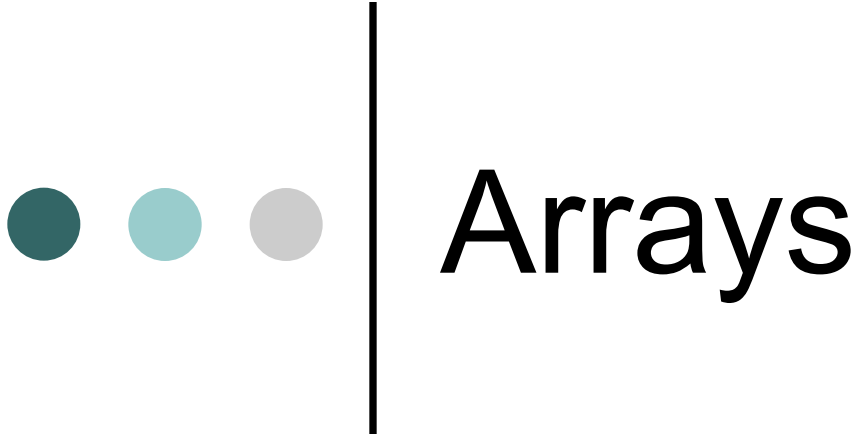
Coleção de Objetos

- Desejamos:
 - Adicionar um objeto dentro da coleção
 - Remover um objeto da coleção
 - Pesquisar (achar a referência a) um objeto particular da coleção, dada uma chave
 - Percorrer (varrer) os objetos da coleção

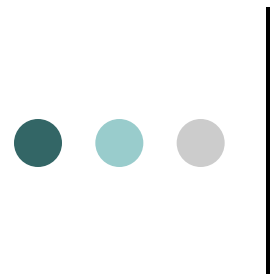


Coleção de Objetos

- Em Java podemos manipular coleções de diversas formas
 - Array
 - ArrayList
 - Vector
 - Etc....



Arrays

- 
- E se precisarmos de um número grande (ou indeterminado) de objetos como atributo de uma classe?
 - Ex.1: Classe Locadora e Classe DVD
 - Uma locadora tem uma coleção vasta de objetos da classe DVD.
 - Ex.2: Classe Biblioteca e Classe Livro
 - Uma biblioteca também tem uma coleção vasta de objetos da classe Livro.
 - Ex.3: Classe Faculdade e Classe Aluno
 - Etc.



Arrays

- Um vetor (**array**) é uma seqüência de objetos ou valores de tipos primitivos, todos do mesmo tipo e combinados sob um único identificador.
- Vetores são estáticos.
 - O seu tamanho é definido no momento da sua criação.
- Em Java, vetores são objetos.
 - Na prática, eles herdam de Object.
- Arrays possuem um atributo público que informa o seu tamanho: **length**.
- Arrays em Java iniciam na posição (índice) 0.



Arrays

- Na programação OO, podemos ter **arrays de tipos primitivos** e **arrays de objetos**.
- Declaração: `tipo[] identificador`;
 - Exemplos:
 - `int[] vet;`
 - `Button[] b;`

Declaração de um vetor de tipo primitivo.

Declaração de um vetor de objeto.

Arrays

- o **Unidimensionais**

- o **Construção:** identificador = new tipo[tamanho]

- Exemplo: `vet = new int[12];`
`b = new Button[10];`

- o **Inicialização:**

```
int[] vet = {1,2,3,4};
```

```
String[] Mes = {"JAN", "FEV", "MAR", "ABR", "MAI",  
"JUN", "JUL", "AGO", "SET", "OUT", "NOV", "DEZ"};
```

Observe que um vetor pode ser declarado, construído e inicializado ao mesmo tempo.



Arrays

○ Exemplos – Tipos primitivos

- `Int[] idades = new int[6];`
- `boolean[] historicoDeValidacoes;`
`historicoDeValidacoes = new`
`boolean[50];`
- `String[] diasSemana = {"domingo",`
`"segunda", "terça", "quarta",`
`"quinta", "sexta", "sábado"};`
- `float[] registrosDeTemperatura;`
`registrosDeTemperatura = new`
`float[qtdRegistros];`



Arrays

- Exemplos – Arrays de objetos:

- `Carro[] carrosDanificados = new Carro[50];`
- `Endereco[] endrs = new Endereco[5];`
- `Ponto[] pontos = new Ponto[qtdPontos];`



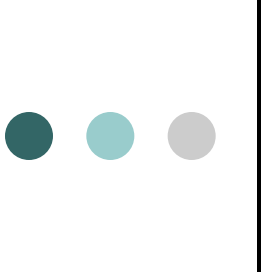
Arrays

- Índice deve ser inteiro.
 - Erros se os limites forem ultrapassados
 - Tamanho não pode ser modificado
- Tipos dos elementos não podem ser modificados
- Podemos declarar assim também:
 - `Int idades[];`
 - Mas não é tão comum



Arrays

- Exemplo array de String representando dias da semana



Esse exemplo mostra os meses e a quantidade de dias que cada mês possui.

```
class DiasDosMeses
{
    public static void main(String[] arg)
    {
        int[] maxDiasMes = new int[12];
        String[] nomeMes = {"JAN", "FEV", "MAR", "ABR", "MAI", "JUN",
                           "JUL", "AGO", "SET", "OUT", "NOV", "DEZ"};
        for(int i=0; i < maxDiasMes.length; i++)
        {
            if(((i+1 < 8) && ((i+1)%2==1)) || ((i+1 >= 8) && ((i+1)%2==0)))
                maxDiasMes[i] = 31;
            else
                maxDiasMes[i] = 30;
        }
        maxDiasMes[1] = 28;
        for(int i=0; i<12; i++)
            System.out.println(nomeMes[i]+":"+ maxDiasMes[i]);
    }
}
```




Matrizes

- Os arrays multidimensionais funcionam de forma análoga aos arrays unidimensionais.
- Cada dimensão é representada por um par de colchetes [].
- A propriedade length, quando associada a matriz, retorna o número de linhas
- A propriedade length, quando associada a uma linha, retorna o número de colunas



Matrizes

Exemplo.

```
int[][] x = new int[3][5];
```

```
int y=x.length;    y terá o valor 3
```

```
int w=x[0].length;    w terá o valor 5
```



Matrizes

```
class ManipulaMatriz
{
    public static void main(String args[])
    {
        int[][] Mat = new int[5][2];
        for( int i=0; i < Mat.length; i++)
        {
            for( int j=0; j < Mat[0].length; j++)
            {
                Mat[i][j]= (i*2)+j;
                System.out.print(" Mat[ "+ i +" ] " + "[ " + j + " ] = " + Mat[i][j]);
            }
            System.out.println(" ");
        }
    }
}
```

Declaração e construção de uma matriz.



Arrays

○ Boas Práticas

● Encapsulamento

- Declaração do array como campo da classe
- Inicialização no construtor
 - Outros Métodos terão acesso ao array já inicializado.



Composição X Arrays

- Como irá chamar os métodos de f e c a partir de locadora que é composta por um array de clientes e filmes?

```
public class Locadora
{
    // instance variables - replace the example below with
    your own
    private String nomeLocadora;
    private String enderecoLocadora;
    private Cliente[] c;
    private Filme[] f;
```