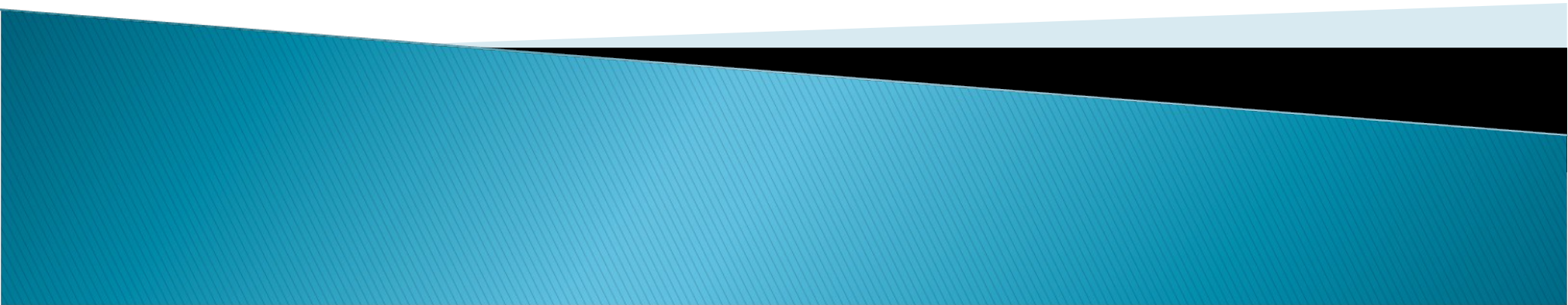
Arrays em Java

Profª. Juliana Pinheiro Campos
E-mail: jupcampos@gmail.com
ENG10082 – Programação II



Coleções

- Coleção: São objetos que podem armazenar um número arbitrário de outros objetos.

- Algumas aplicações precisam ser capazes de agrupar objetos:
  - Agenda eletrônica;
  - Biblioteca;
  - Registro de alunos em escolas e universidades;

Array (coleção de tamanho fixo)

- Arrays são estruturas de dados em Java que permitem o armazenamento de várias variáveis (uma coleção) de um mesmo tipo ou instâncias de uma mesma classe usando uma única referência e um índice de acesso.
- Cada valor do array pode ser acessado individualmente.
- O array inteiro pode ser processado como uma única entidade caso seja desejado.

Array (coleção de tamanho fixo)

- A declaração de arrays em Java é feita usando a notação de colchetes.

tipo_dos_elementos[] nome_do_array;

ou

tipo_dos_elementos nome_do_array[];

- Exemplos:

int notas[];

char[] letrasDoAlfabeto;

Array (coleção de tamanho fixo)

- As referências de arrays devem ser inicializadas (ter memória alocada para seus elementos) com a palavra-chave **new**, seguida do tipo de dado a ser alocado e do número de elementos a alocar, entre colchetes.

- Exemplos:

```
notas = new int[10];

char[] letrasDoAlfabeto = {'a','b'};   // só na declaração
```

- Os elementos do array são acessados usando o nome da referência seguido do índice do array (que varia de 0 a tamanho do array menos um).

Array (coleção de tamanho fixo)

- Quando arrays de tipos nativos são inicializados, os elementos automaticamente recebem valores default.

- Todo array unidimensional possui um **campo** length que indica o número de elementos no array. Ele se comporta como um campo final.

- A atribuição de uma referência de array a outra que foi declarada do mesmo tipo faz com que as duas apontem para a mesma posição de memória.

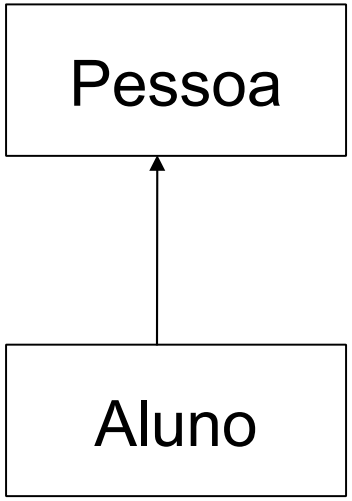
Array (coleção de tamanho fixo)

- Arrays de instâncias de classes podem ser declaradas de forma bem similar a arrays de valores de tipos nativos.

- A principal diferença é que a inicialização do array deve ser seguida da inicialização dos elementos do array, que deve ser feita da mesma maneira que fazemos com instâncias de classes comuns.

- Outra diferença é que elementos de arrays de instâncias de classe podem conter instâncias de qualquer classe que seja herdeira da classe usada para declaração do array (polimorfismo).

Array (coleção de tamanho fixo)




```
Pessoa[] colecao = new Pessoa[5];
colecao[0] = new Pessoa(“Jose da Silva, 22”);
colecao[1] = new Aluno(“Maria Jose”, 19,
“Ciência da Computação”);

System.out.println(colecao[1]);
System.out.println(colecao[2]);    //imprime null
```

Array (coleção de tamanho fixo)

- Exemplo: Bloco de notas pessoal

- Modelaremos uma aplicação de bloco de notas pessoal que tem os seguintes recursos básicos:
  - Permite que seja armazenado um nº x de notas (lembretes);
  - Permite mostrar as notas individualmente e mostrar também todas as notas;
  - Informa quantas notas estão atualmente armazenadas;
  - Remover uma nota.

Array (coleção de tamanho fixo)

- Arrays multidimensionais: são aqueles arrays onde a posição de cada elemento será indicada por dois ou mais índices. Ex: Matriz.

- Declaração de matriz:

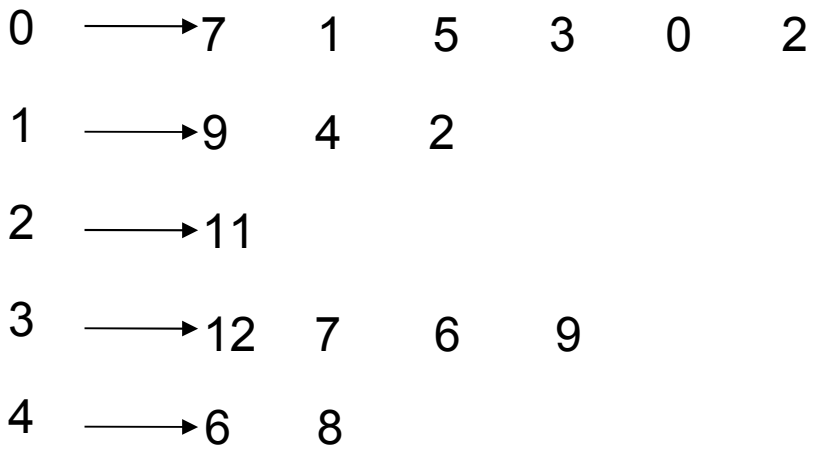
```
private int matriz[][];

private int[][] matriz = new int[4][3];          // com inicialização
```

- Como inicializar/ imprimir um array multidimensionais?

Array (coleção de tamanho fixo)

- Arrays irregulares: uma característica interessante dos arrays multidimensionais implementados em Java é que eles não precisam ter o mesmo número de valores para cada dimensão. É possível declararmos arrays que sejam irregulares.



Array (coleção de tamanho fixo)

- Arrays irregulares é considerado como um array unidimensional onde cada elemento é outro array unidimensional que contém valores de determinado tipo.

- Pode ser declarado da mesma forma que um array multidimensional, exceto que somente o tamanho da primeira das dimensões precisa ser especificado. As outras dimensões serão alocadas em passos subsequentes.

```
public int [][] trianguloDePascal = new int[numLinhas][];

trianguloDePascal[0] = new int[6];

trianguloDePascal[1] = new int[3];
```

Coleção de tamanho variável (flexível)

- Número de itens armazenados na coleção varia: seu tamanho é modificado de acordo com a necessidade.
- Os itens são adicionados e excluídos com freqüência.
- O ideal seria que não fosse necessário conhecer o número de itens agrupados.

A classe ArrayList

- ArrayList é um exemplo de coleção de tamanho flexível.
- É uma lista de objetos: coleções onde elementos repetidos podem ocorrer e onde os elementos têm posições definidas.
- É uma classe (do pacote java.util) de coleção que pode armazenar um conjunto de objetos (somente objetos).

A classe ArrayList

- Métodos da classe:
- add: adiciona um novo objeto no final da lista;
- size: retorna quantos objetos estão armazenados na lista;
- get: recupera um objeto da lista. Recebe como argumento a posição do objeto que se quer recuperar;
- remove: remove um objeto da lista;
- E muitos outros!

A classe ArrayList

- Recursos da classe:
  - Aumenta sua capacidade interna conforme necessário;
  - Mantém a contagem do número de itens armazenados atualmente;
  - Mantém a ordem de itens que você insere nela.
- ArrayList tem um trabalho complexo. Essa é a grande vantagem de utilizar classes de bibliotecas. Detalhes como tudo isso é feito é ocultado (encapsulado).

A classe ArrayList

- Desvantagens em relação as classes de coleção de tamanho fixo:

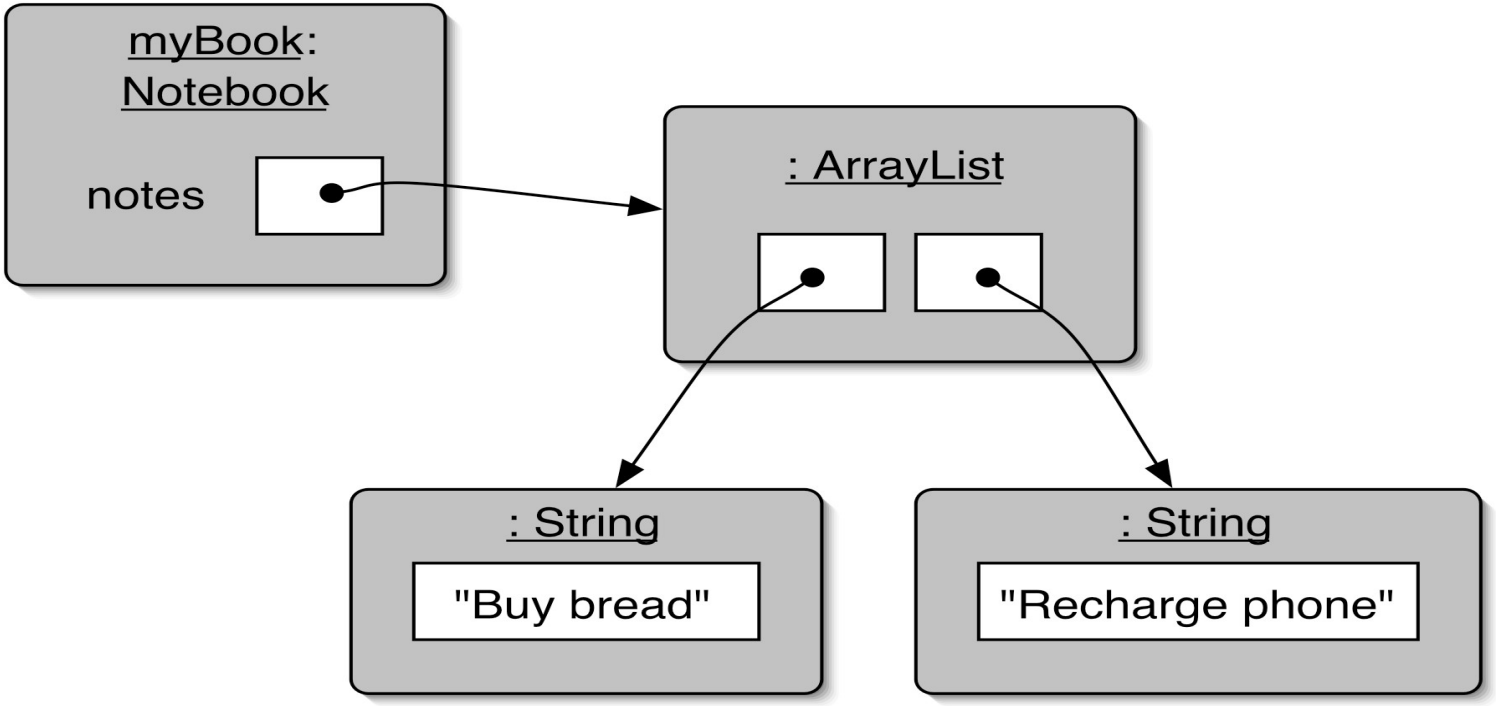
  - O acesso aos itens mantidos em um array é mais eficiente do que o acesso aos itens em uma coleção de tamanho flexível.

  - Os arrays são capazes de armazenar objetos ou valores de tipo primitivo. As coleções de tamanho flexível podem armazenar **somente objetos**.

A classe ArrayList

- Exemplo: Bloco de notas pessoal

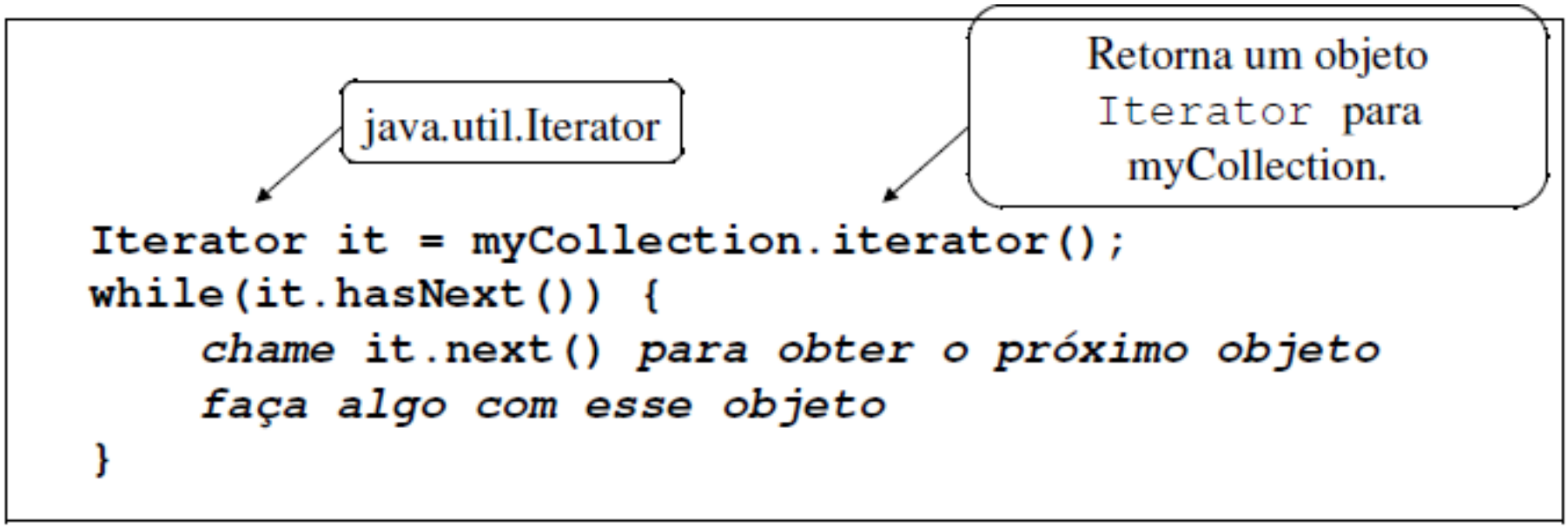
- Altere o bloco de notas do exemplo anterior para que ele utiliza um ArrayList ao invés de uma lista de tamanho fixo.



Iterando por uma coleção

- **Iterador** é um objeto que fornece funcionalidade para iterar por todos os elementos de uma coleção.
- A classe **Iterator** é definida no pacote java.util.
- O método iterator retorna um objeto do tipo Iterator para uma coleção.
- Métodos para iterar por uma coleção:
  - hasNext(): verifica se não há mais objetos na coleção;
  - next(): obtém o próximo objeto;

Iterando por uma coleção



Altere o método mostrarNotas para que ele utilize um Iterator.

Acesso por índice x Iteradores

- Podemos iterar por um ArrayList de duas formas:
  - utilizando uma estrutura de repetição e o método get com um índice; ou
  - utilizando um objeto Iterator.
- Para algumas coleções é impossível ou muito ineficiente acessar elementos individuais fornecendo um índice.
- A utilização do método get é uma solução particular para ArrayList.
- A solução utilizando um iterador, está disponível para todas as coleções na biblioteca de classes Java.