Coleções

Array e ArrayList

DC – UFRPE

Programação II

Prof. Gustavo Callou

gustavo.callou@ufrpe.br

Agenda

- ArrayList

ArrayList

- Implementa a interface (métodos) de uma lista

  Adicionar, remover, recuperar, concatenar, etc

- Utiliza um array internamente
- Armazena elementos de apenas um tipo definido na declaração da lista

Exemplo: lista de st

Tipo da lista

```
ArrayList<String> nomes = new ArrayList();

nomes.add("João");
nomes.add("Pedro");

if(nomes.contains("Pedro"))
    nomes.remove("Pedro");

System.out.println(nomes.size());
```

Adiciona na lista

Se existe na lista remove

Imprime tamanho da lista

Percorrendo um ArrayList

```
for (int i = 0; i < nomes.size(); i++) {
  System.out.println(nomes.get(i));
}
```

OU

```
for (String nome : nomes) {
  System.out.println(nome);
}
```

Wrapper

- Wrapper são utilizados em OO para representar um tipo em contextos onde ele normalmente não poderia ser utilizado

- Exemplo: não consigo fazer uma lista de inteiros, mas consigo fazer uma lista de Integer (que empacota inteiros)

Wrapper

- Empacotando inteiros (wrapping)

```
Integer i1 = new Integer(1);
Integer i2 = new Integer(2);
```

- Desempacotando inteiros (unwrapping)

```
int x1 = i1.intValue();
int x2 = i2.intValue();
```

Wrapper

- Guardando inteiros em um ArrayList<Integer>
  1. Empacotar inteiro

```
Integer i = new Integer(1);
```

  2. Guardar/recuperar na lista

```
ArrayList<Integer> lista = new ArrayList();
lista.add(i);
Integer x1 = lista.get(0);
```

  3. Desempacotar inteiro

```
int y1 = x1.intValue();
```

Wrappers para tipos primitivos

Tipo primitivo	Classe Wrapper
boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double

Wrapper

```
ArrayList<Double> lista = new ArrayList();

double d = 0.56;

lista.add(d);
lista.add(1.44376);

double d2 = lista.get(0);
```

Garbage Collection (GC)

- Mecanismo da JVM que remove da memória objetos não utilizados
- O momento no qual o GC é executado não é controlado pelo usuário

Exemplo de objeto sem referência

```
Hora h = new Hora();

h = new Hora();

...
```

(1) h aponta para um objeto Hora

(2) h aponta para outro objeto Hora

(3) A partir deste ponto o GC pode liberar o espaço utilizado pelo 1° objeto Hora

Exercício 1

- Crie uma cópia da classe **LivroNotas** da Aula 5 com o nome **LivroNotasArray**
- Modifique **LivroNotasArray** para guarda as notas em um array
  - Tamanho do array é passado no construtor
  - Remova o atributo **somaNotas**
  - Manter métodos **adicionarNota** e **calcularMedia**
- Incluir os métodos:
  - **public double** getMaiorNota()
  - **public double** getMenorNota()

Exercício 2

- Crie uma cópia da classe **LivroNotasArray** do exercício anterior e renomeie para **LivroNotasArrayList**
- Modifique **LivroNotasArrayList** para trabalhar com ArrayList no lugar de um array