

Arrays em Java

Introdução

Arrays são estruturas e dados em Java que permitem o armazenamento de várias variáveis de um mesmo tipo ou instâncias de uma mesma classe usando uma única referência e um índice de acesso.

Por exemplo, um array pode armazenar os dados de alunos de turma de 30 alunos.

Arrays unidimensionais

A declaração de arrays em Java é feita usando a notação dos colchetes: uma variável declarada como sendo de um tipo específico seguida de um par de colchetes será uma referência a um array de um elementos daquele tipo. Exemplos:

```
int [] posição;  
char [] vogais;  
double[] alturaTurma;
```

Não basta declarar as referências a arrays, estas devem ser inicializadas. O trecho de código abaixo mostra várias maneiras de inicializar arrays.

```
int[] posição = new int [50];  
char[] vogais = {'a','e','i','o','u'};  
double[] alturaTurma;  
double[] alturaTurma2 = alturaTurma;
```

O índice para acesso de um array deverá ser um valor entre zero e o tamanho do array menos um. O índice para acessar um elemento de um array deve ser sempre inteiro e positivo. Na inicialização do array devemos especificar o seu tamanho entre colchetes. Esse tamanho não precisa ser uma constante definida em tempo de execução. O tamanho mínimo aceitável para um array é zero: como os índices são contados a partir de zero, um array declarado de tamanho zero terá um único elemento.

Um array não pode ter seu tamanho modificado, embora possamos usar a referência para apontar para outro array.

Arrays que deverão representar valores constantes podem ser inicializados diretamente com os valores dos elementos do array, que devem ser especificados como uma lista de valores do tipo dos elementos do array, separados por vírgulas e cercados por chaves.

Quando existe um campo que é um array, este deve ser inicializado no construtor da classe.

O campo estático **length** do tipo final, fornece o número de elementos do array.

Quando temos array composto de elementos de tipos nativos, estes devem ser do mesmo tipo ou compatível. Quando se trata de array de instâncias de classes, este pode ser constituído de objetos pertencentes a classes da hierarquia.

Um exemplo de uso de vetores com elementos de tipo nativo segue na classe Array DeFloats.

```
public class ArrayDeFloats {  
    private float [] array;  
  
    ArrayDeFloats(int numero) {  
        array = new float[numero];  
    }  
  
    public int tamanho() {
```

```

    return array.length;
}

public void modifica(int posicao, float valor) {
    if ((posicao >= 0) && (posicao < array.length) )
        array[posicao] = valor;
}

public float valor(int posicao) {
    if ((posicao >= 0) && (posicao < array.length) )
        return array[posicao];
    else return Float.NaN;
}

public float[] paraArray() {
    return array;
}

public float menorValor() {
    float menorAtéAgora = array[0];
    for (int i = 0; i < array.length; i++)
        if (array[i] < menorAtéAgora)
            menorAtéAgora = array[i];
    return menorAtéAgora;
}

public float maiorValor() {
    float maiorAtéAgora = array[0];
    for (int i = 0; i < array.length; i++)
        if (array[i] > maiorAtéAgora)
            maiorAtéAgora = array[i];
    return maiorAtéAgora;
}

public String toString() {
    String resultado = "O array tem "+array.length+" elementos:\n";
    for (int i = 0; i < array.length; i++)
        resultado += array[i]+ " ";
    return resultado;
}
}

```

```

public class AppArrayDeFloats {
    public static void main (String args[]) {
        ArrayDeFloats af = new ArrayDeFloats(25);
        for (int i = 1; i < 40; i++)
            af.modifica(i,(float)1/i);
        System.out.println(af);
        System.out.println("Maior valor = "+af.maiorValor());
        System.out.println("Menor valor = "+af.menorValor());

        System.out.println("Na posição 0 = "+af.valor(0));
        System.out.println("Na posição 24 = "+af.valor(24));
        System.out.println("Na posição 25 = "+af.valor(25));
    }
}

```

```
float[] cópia = af.paraArray();
System.out.println("CÓPIA");
System.out.println("Na posição 0 = "+cópia[0]);
System.out.println("Na posição 0 = "+af.paraArray()[0]);
}
}
```

Observe a classe abaixo:

```
public class Aluno {
    private String nome;
    private float[] notas;
    private int numeroAvaliacoes;

    public Aluno(String nome) {
        this.nome = nome;
        notas = new float[4];
        numeroAvaliacoes = 0;
    }

    public boolean setNota(float n) {
        if (numeroAvaliacoes >= 4) return false;
        else {
            notas[numeroAvaliacoes] = n;
            numeroAvaliacoes++;
            return true;
        }
    }

    public String toString() {
        String retorno = "nome: "+nome+"\n";
        for (int i = 0; i <= 3; i++)
            retorno += "nota: "+(i+1)+" : "+notas[i)+"\n";
        return retorno;
    }
}
```

Altere este código para:

- Calcular e retornar a média aritmética das provas;
- O resultado final ("Aprovado", "Prova Final" ou "Reprovado"), considerando os critérios adotados na faculdade;
- Altere o método toString para retornar também estas informações.