

Arrays

Prof. Fernando J. C. Branquinho
<http://br.geocities.com/branqs>

Arrays

=====

Arrays em Java são objetos que permitem armazenar coleções de valores primitivos ou objetos. Arrays também são chamados de matrizes.

Declaração

```
int[] numeros;
```

Declara um array para armazenar números inteiros

```
Contato[] contatos;
```

Declara um array para armazenar vários objetos do tipo Contato.

Instanciação

```
numeros= new int[10];
```

Cria um array para 10 valores.

```
contatos = new Contato[20];
```

Cria um array para 20 contatos

Acesso

As várias posições de um array são acessadas através de um índice. Esse índice vai de 0 até o tamanho do array menos 1.

Ex:

```
numeros[0]=30;  
numeros[1]=15;  
numeros[2]=23;
```

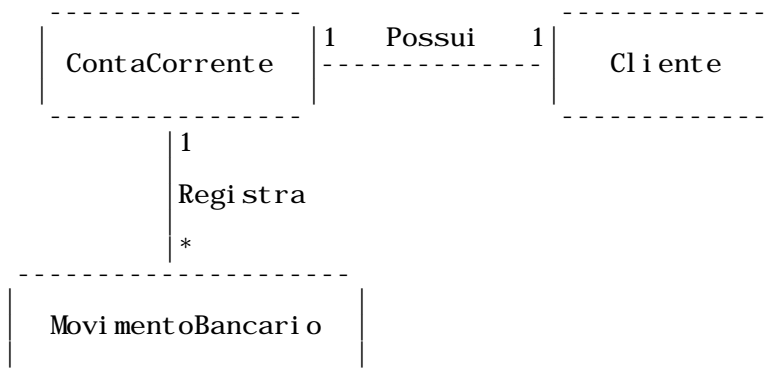
```
.  
. .  
.
```

```
numeros[9]=90;
```

Exemplo do uso de Array no projeto "Banco"

=====

Considere o seguinte diagrama simplificado de classes:



```

-----
public class MovimentoBancario
{
    private String descricao;
    private float valor;

    public MovimentoBancario(String descricao, float valor) {
        this.descricao=descricao;
        this.valor=valor;
    }

    public float obterValor() {
        return valor;
    }

    public String obterDescricao() {
        return descricao;
    }
}

```

```

-----
public class ContaCorrente {

    private float saldo;
    private Cliente correntista;
    private MovimentoBancario[] movimentos;
    private int numeroUltimoMovimento;

    public ContaCorrente(Cliente cli) {
        correntista=cli;
        saldo=0;

        numeroUltimoMovimento=0;
        movimentos = new MovimentoBancario[100];

        MovimentoBancario mov=new MovimentoBancario("Saldo inicial", saldo);
        movimentos[numeroUltimoMovimento]=mov;
    }

    public float obterSaldo() {
        return saldo;
    }

    public void realizarDeposito(float valor) {
        MovimentoBancario mov = new MovimentoBancario("Depósito", valor);
        numeroUltimoMovimento++;
        movimentos[numeroUltimoMovimento]=mov;
        saldo=saldo+valor;
    }

    public void realizarSaque(float valor, String senha) {
        if (correntista.compararSenha(senha) && saldo >= valor)
            saldo=saldo-valor;
    }

    public String obterNomeCorrentista() {
        return correntista.obterNome();
    }
}

```

```

-----
public class Cliente
{
    private String nome;
    private String senha;

    public Cliente(String nome, String senha) {
        this.nome=nome;
        this.senha=senha;
    }
}

```

```

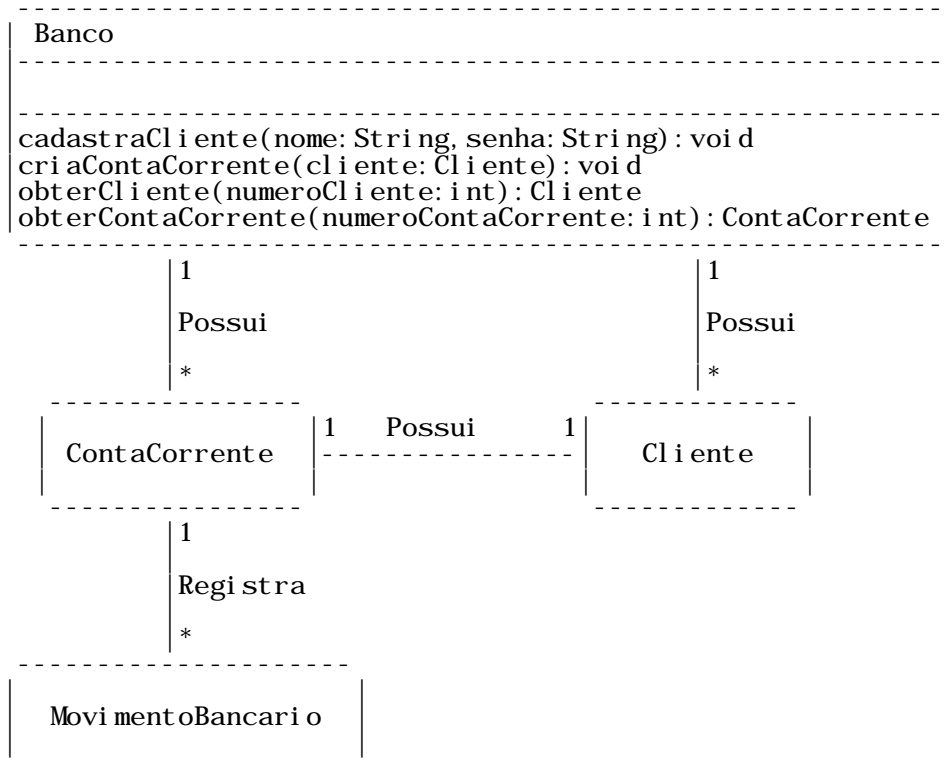
    public boolean compararSenha(String senhaInformada) {
        return this.senha.equals(senhaInformada);
    }

    public String obterNome() {
        return nome;
    }
}

```

Exercícios

- 1) Gerar um movimento bancário também na realização de saques
- 2) Inserir a classe Banco no sistema, de acordo com o diagrama abaixo:



Final do documento

Fernando J. C. Branquinho
<http://br.geocities.com/branqs>
