

09

Mão na massa: Arrays

Chegou a hora de praticarmos o que vimos neste capítulo.

1 - Temos a seguinte classe:

```
package br.com.bytebank.banco.test;

import br.com.bytebank.banco.modelo.ContaCorrente;

public class TestArrayReferencias {

    public static void main(String[] args) {

        ContaCorrente[] contas = new ContaCorrente[5];
        ContaCorrente cc1 = new ContaCorrente(22, 11);
        contas[0] = cc1;

        ContaCorrente cc2 = new ContaCorrente(22, 22);
        contas[1] = cc2;

        System.out.println( contas[1].getNumero() );

        ContaCorrente ref = contas[0];
        System.out.println(cc2.getNumero());
        System.out.println(ref.getNumero());

    }
}
```

2 - Queremos guardar no array `conta` instâncias de `ContaCorrente` ou `ContaPoupanca`. Para isso, precisamos que `conta` seja de um tipo mais genérico, no caso `Conta`.

```
package br.com.bytebank.banco.test;

import br.com.bytebank.banco.modelo.ContaCorrente;
// importou
import br.com.bytebank.banco.modelo.ContaPoupanca;

public class TestArrayReferencias {

    public static void main(String[] args) {

        // alterando o tipo
        Conta[] contas = new Conta[5];
        ContaCorrente cc1 = new ContaCorrente(22, 11);
        contas[0] = cc1;

        // cria instância de ContaPoupanca
        ContaPoupanca cc2 = new ContaPoupanca(22, 22);
        contas[1] = cc2;

    }
}
```

```

System.out.println(contas[1].getNumero()  );

// não compila
ContaCorrente ref = contas[0];
System.out.println(cc2.getNumero());
System.out.println(ref.getNumero());

}

}

```

Ao mudarmos o tipo do array `conta` de `ContaCorrente`, a seguinte instrução deixa de compilar:

```
ContaCorrente ref = contas[0];
```

Por que, em tempo de execução, tenhamos um objeto do tipo `ContaCorrente` como primeiro elemento do array `contas`, sua referência é do tipo `Conta` quando acessarmos `contas[0]`. Sendo assim, uma solução é alterarmos o tipo da variável `ref` para `Conta`:

```

package br.com.bytebank.banco.test;

import br.com.bytebank.banco.modelo.ContaCorrente;
// importou
import br.com.bytebank.banco.modelo.ContaPoupanca;

public class TestArrayReferencias {

    public static void main(String[] args) {

        // alterando o tipo
        Conta[] contas = new Conta[5];
        ContaCorrente cc1 = new ContaCorrente(22, 11);
        contas[0] = cc1;

        // cria instância de ContaPoupanca
        ContaPoupanca cc2 = new ContaPoupanca(22, 22);
        contas[1] = cc2;

        System.out.println(contas[1].getNumero()  );

        // alterou o tipo, agora compila
        Conta ref = contas[0];
        System.out.println(cc2.getNumero());
        System.out.println(ref.getNumero());
    }
}

```

3 - Modifique o código para que meso utilizando o tipo `ContaCorrente` seja possível atribuir à variável `ref` o valor de `contas[1]`. Como temos certeza que o elemento na posição `contas[0]` é uma instância de `ContaCorrente`, podemos assumir a responsabilidade de conversão através de um `type cast`:

```
package br.com.bytebank.banco.test;
```

```
import br.com.bytebank.banco.modelo.ContaCorrente;
// importou
import br.com.bytebank.banco.modelo.ContaPoupanca;

public class TestArrayReferencias {

    public static void main(String[] args) {

        // alterando o tipo
        Conta[] contas = new Conta[5];
        ContaCorrente cc1 = new ContaCorrente(22, 11);
        contas[0] = cc1;

        // cria instância de ContaPoupanca
        ContaPoupanca cc2 = new ContaPoupanca(22, 22);
        contas[1] = cc2;

        System.out.println(contas[1].getNumero() );

        // alterou o tipo, realizando o cast
        ContaCorrente ref = (ContaCorrente) contas[0];
        System.out.println(cc2.getNumero());
        System.out.println(ref.getNumero());
    }
}
```

4 - Experimente agora acessar o elemento na posição `contas[1]`. Como em tempo de execução ele é do tipo `ContaPoupanca`, o cast não funcionará e um exceção será lançada no console.