03

Mão na massa: GuardadorDeContas

Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, execute os passos listados abaixo.

1) Crie a classe `GuardadorDeContas` :

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

public class GuardadorDeContas {

    private Conta[] referencias;

    public GuardadorDeContas() {
        this.referencias = new Conta[10];
    }

}
```

2) Crie uma classe para testar o guardador de contas.

```
public class Teste {
    public static void main(String[] args) {
        GuardadorDeContas guardador = new GuardadorDeContas();

        Conta cc = new ContaCorrente(22, 11);
        guardador.adiciona(cc);
    }
}
```

3) Crie o método `adiciona()` no guardador:

```
public class GuardadorDeContas {

    private Conta[] referencias;
    private int posicaoLivre;

    public GuardadorDeContas() {
        this.referencias = new Conta[10];
        this.posicaoLivre = 0;
    }

    public void adiciona(Conta ref) {
        referencias[this.posicaoLivre] = ref;
        this.posicaoLivre++;
    }

}
```

4) Modifique sua classe de teste para incluir mais uma conta:

```
public class Teste {
    public static void main(String[] args) {
        GuardadorDeContas guardador = new GuardadorDeContas();

        Conta cc = new ContaCorrente(22, 11);
        guardador.adiciona(cc);

        Conta cc2 = new ContaCorrente(22, 22);
        guardador.adiciona(cc2);
    }
}
```

5) Agora queremos verificar se a quantidade de elementos dentro do guardador é 2. Crie o código na classe de teste e aproveite a sugestão do Eclipse para criar o método para você...

```
public class Teste {
    public static void main(String[] args) {
        GuardadorDeContas guardador = new GuardadorDeContas();

        Conta cc = new ContaCorrente(22, 11);
        guardador.adiciona(cc);

        Conta cc2 = new ContaCorrente(22, 22);
        guardador.adiciona(cc2);

        int tamanho = guardador.getQuantidadeDeElementos();
        System.out.println(tamanho);
    }
}
```

...e em seguida preencha a implementação do método na classe `GuardadorDeContas` :

```
public class GuardadorDeContas {

    private Conta[] referencias;
    private int posicaoLivre;

    public GuardadorDeContas() {
        this.referencias = new Conta[10];
        this.posicaoLivre = 0;
    }

    public void adiciona(Conta ref) {
        referencias[this.posicaoLivre] = ref;
        this.posicaoLivre++;
    }

    public int getQuantidadeDeElementos() {
        return this.posicaoLivre;
    }
}
```

6) Adicione mais uma funcionalidade na classe de teste para recuperar determinado elemento do guardador a partir de uma posição. Novamente use o Eclipse para gerar o método para você:

```
public class Teste {
    public static void main(String[] args) {
        GuardadorDeContas guardador = new GuardadorDeContas();

        Conta cc = new ContaCorrente(22, 11);
        guardador.adiciona(cc);

        Conta cc2 = new ContaCorrente(22, 22);
        guardador.adiciona(cc2);

        int tamanho = guardador.getQuantidadeDeElementos();
        System.out.println(tamanho);

        Conta ref = guardador.getReferencia(0);
        System.out.println(ref.getNumero());
    }
}
```

7) Agora implemente o método em `GuardadorDeContas`:

```
public class GuardadorDeContas {

    private Conta[] referencias;
    private int posicaoLivre;

    public GuardadorDeContas() {
        this.referencias = new Conta[10];
        this.posicaoLivre = 0;
    }

    public void adiciona(Conta ref) {
        referencias[this.posicaoLivre] = ref;
        this.posicaoLivre++;
    }

    public int getQuantidadeDeElementos() {
        return this.posicaoLivre;
    }

    public Conta getReferencia(int pos) {
        return this.referencias[pos];
    }
}
```

8) Execute a classe `Teste` para verificar que o guardador está funcionando.

9) (Desafio) Agora experimente criar um guardador que saiba guardar qualquer tipo de referências, usando a classe `Object`.