

Passo a Passo para criar classes e objetos

Exercício sobre gestão de um cartão de pontos do continente

1. Criar a Classe

Vamos começar apenas com o que uma Conta tem, e não com o que ela faz. Um tipo desses, como o especificado de Conta acima, pode ser facilmente traduzido para Java:

```
class Conta {  
    int nif;  
    String nome;  
    double saldo;  
    double totalcompras; }
```

2. Criar um programa principal

Já temos uma classe em Java que especifica tudo o que o objeto dessa classe deve ter.

Mas como usá-la?

Além dessa classe, vamos ter o Programa.java e a partir dele é que iremos utilizar a classe Conta. Para criar (construir, fornecer, etc) uma Conta, basta usar a palavra chave **new**, utilizamos também os parênteses:

```
class Programa {  
    public static void main(String[] args) {  
        new Conta();  
    }  
}
```

Bem, o código acima cria um objeto do tipo Conta, mas como aceder a esse objeto que foi criado? Precisamos ter alguma forma de referenciar esse objeto. Precisamos de uma variável:

```
Conta minhaConta;  
minhaConta = new Conta();
```

Pode parecer estranho escrevermos duas vezes Conta: uma vez na declaração da variável e outra vez no uso do **new**.

Através da variável minhaConta, podemos aceder ao objeto recém-criado para alterar a informação.

```
minhaConta = new conta();  
  
minhaConta.nome = "António";  
minhaConta.saldo = 1000.0;
```

É importante fixar que o ponto foi utilizado para aceder a algo na minhaConta. A minhaConta pertence ao António, e tem saldo de mil euros.

Método

Dentro da classe, também declaramos o que cada conta faz e como isto é feito - os comportamentos que cada classe tem, isto é, o que ela faz. Por exemplo, de que maneira que uma Conta movimenta o dinheiro? Especificaremos isso dentro da própria classe Conta, e não num local desligado das informações da própria Conta. É por isso que a essas “funções” chamamos métodos. Pois é a maneira de fazer uma operação com um objeto. Queremos criar um método que movimenta uma determinada quantidade e que não devolve nenhuma informação para quem ativar esse método:

```
public class conta {  
  
    int nif;  
    String nome;  
    double saldo;  
    double totalcompras;  
  
    void movimento (double quantidade) {  
        double novoSaldo = this.saldo - quantidade;  
        this.saldo = novoSaldo;  
        this.totalcompras += this.totalcompras;  
    }  
  
}
```

A palavra chave **void** diz quando pedimos para a conta movimentar uma quantia, nenhuma informação será enviada de regresso a quem pediu. Quando alguém pedir para movimentar, ele também vai dizer quanto quer movimentar.

Por isso precisamos declarar o método com algo dentro dos parênteses - o que vai aí dentro é chamado de argumento do método (ou parâmetro). Essa variável é uma variável comum, chamada também de temporária ou local, pois, ao final da execução desse método, ela deixa de existir.

Dentro do método, vamos declarar uma nova variável. Essa variável, assim como o argumento, vai morrer no fim do método, pois este é seu objetivo. No momento que vamos aceder ao nosso atributo, usamos a palavra chave **this** para mostrar que esse é um atributo, e não uma simples variável. Repare que, nesse caso, a conta vai atualizar o total de compras.

Outro método:

```
boolean retirar(double valor) {  
    if (this.saldo < valor) {  
        return false;  
    }  
    else {  
        this.saldo = this.saldo - valor;  
        this.totalcompras += this.totalcompras;  
        return true;  
    }  
}
```

Duas contas na mesma classe:

```
public class programa {  
  
    public static void main(String[] args) {  
        conta minhaConta, outraconta;  
        minhaConta = new conta();  
        outraconta= new conta();  
  
        minhaConta.nome = "Antônio";  
        minhaConta.saldo = 1000.0;  
        outraconta.nome = "Ronaldo";  
        outraconta.saldo = 10000000.0;  
  
        System.out.println("Saldo atual: " + minhaConta.saldo);  
    }  
}
```

Quando declaramos uma variável para associar a um objeto, na verdade, essa variável não guarda o objeto, é sim uma maneira de aceder ou seja a chamada de referência. É por esse motivo que é, diferente dos tipos primitivos como int e long, precisamos dar new depois de declarada a variável.

O correto aqui, é dizer que minhaconta se refere a um objeto. Não é correto dizer que minhaconta é um objeto, pois minhaconta é uma variável referência, apesar dos programadores Java dizerem “Tenho um objeto minhaconta do tipo Conta”, mas apenas para encurtar a frase “Tenho uma referência minhaconta a um objeto do tipo Conta”. Basta lembrar que, em Java, uma variável nunca é um objeto. Não há, no Java, uma maneira de criarmos o que é conhecido como “objeto pilha” ou “objeto local”, pois todo objeto em Java, sem exceção, é acedido por uma variável referência.