Java e Orientação a Objetos I

Sumário

Introdução

Java desde sua criação foi feita para ser uma linguagem puramente orientada a objetos por isso que dizemos que tudo em java pode ser uma classe e por sua vez um objeto. Mas o que é a orientação a objetos afinal?

Orientação a objetos é um paradigma de programação que tenta imitar o comportamento humano em relação ao mundo físico que o cerca. Essa explicação parece complexa, mas tente imaginar o seu dia a dia, estamos o tempo todo nos orientado a objetos que estão à nossa volta, por exemplo: digamos que você esteja com sede, para matar essa sede é necessário beber água (nosso primeiro objeto), porém para beber a água nós precisamos de um copo (nosso segundo objeto) e por último precisamos da torneira que fornece essa água (nosso terceiro objeto).

Olha só que incrível, apenas para beber uma água tivemos a interação com 3 objetos totalmente diferentes entre eles, mas que juntos atingem o mesmo objetivo. Essa é a ideia dentro do paradigma de orientação a objetos, é literalmente trabalhar com o código e criar “objetos” que façam as coisas acontecerem.

Classes

Vimos na introdução que a ideia da programação orientada a objetos é justamente criar objetos que interagem entre si e consigam resolver os problemas para os quais eles foram criados. Mas então onde entra a “classe” nisso tudo?

Enquanto os objetos são coisas específicas com características únicas as classes é a forma que podemos classificar esses tipos de objetos para que assim, no java, eles possam ser criados.

Aproveitando o exemplo da introdução onde falamos do copo para armazenar a água, podemos imaginar como seria a classificação dos objetos “Copo” e fazemos isso analisando suas características. Vamos pensar. Quais são as características que todo copo tem? Largura, profundidade, cor, material etc. Essas são algumas das características que podemos pensar.

Agora tente imaginar um copo usando esses atributos, nesse momento podem ter a certeza de que cada um acaba imaginando um copo diferente, um pode imaginar um copo fundo e outro razo, pode imaginar um copo feito de vidro e outro de plástico e assim por diante, e o mais importante todos são copos e sabemos disse como conseguimos classificar eles utilizando uma Classe com os atributos largura, profundidade, cor e material, mas agora os objetos de classe podem e serão diferentes entre si.

Como Criar uma Classe

Para criar essas classe no java podemos abrir nosso projeto e dentro dele localizar o pacote principal do projeto, onde a classe que carrega o método Main existe, e ali podemos criar mais uma classe.

Na maioria das IDEs você pode criar essa classe apenas clicando com o botão direito e selecionando “criar classe java” ou “new file”.

Como exemplo, vou criar uma classe chamada Contador, essa classe deverá criar objetos que são capazes de somar 2 número que são passados para ele.

Vai ficar mais ou menos assim:

```
package com.br.zup;

public class Contador {
}
```

Já vemos logo de cara que temos algumas regras para criar uma classe. A primeira é colocar a para “class” logo depois do “public” isso serve para indicar ao java que esse código se trata de uma classe.

A segunda regra é: toda classe tem a primeira letra do nome em maiúsculo, se tiver mais de uma para então é primeira letra de cada palavra em MAIÚSCULO. Isso é chamado de **PascalCase**.

Adicionando Atributos

Para essa classe “Contador” precisamos adicionar as características dele, nesse caso como nosso objetivo é apenas somar 2 números então podemos dizer que suas características são “primeira parcela” e “segunda parcela” que são os nomes dados em uma operação de soma.

```
package com.br.zup;

public class Contador {
   int primeiraParcela;
   int segundaParcela;

}
```

Aí estão nossos atributos, podemos perceber que com eles colocamos apenas em maiúsculo a segunda palavra do nome, chamamos isso de **camelCase**.

Criando Método

Agora vamos criar o método que faz a soma da primeira parcela e da segunda parcela e retorna para o usuário o valor total.

```
package com.br.zup;

public class Contador {
    int primeiraParcela;
    int segundaParcela;

    public int somarAsParcelas(){
        int total = primeiraParcela + segundaParcela;
        return total;
    }
}
```

Os métodos das classes são os comportamentos que os objetos criados tem. Ou seja, são através dos métodos que os objetos vão interagir com o código e possibilitar a execução de tarefas. No nosso caso o método somar Parcelas vai possibilitar ao objeto contator somar 2 valores.

Instanciando um Objeto

Dentro da classe principal do projeto, a que carrega o método main, podemos instanciar nosso objeto e assim utilizar sua capacidade de somar 2 valores.

```
package com.br.zup;

public class Main {

    public static void main(String[] args) {

        Contador contador = new Contador();
    }
}
```

Podemos notar que antes de instanciar o objeto contador foi necessário criar uma variável que recebe a classe Contador como tipo, e logo depois do sinal de igual instânciamos um objeto Contador usando a palavra “new” e o nome da classe Contador seguido de abre e fecha parênteses.

Preenchendo os Atributos do Objeto

Sempre que criamos um objeto a partir de uma classe os atributos são iniciados, sendo assim é necessário preencher esses atributos com um valor para inicializá-los.

```
package com.br.zup;

public class Main {

   public static void main(String[] args) {

       Contador contador = new Contador();
       contador.primeiraParcela = 13;
       contador.segundaParcela = 2;

   }
}
```

No exemplo acima inserimos dois valores, um em cada atributo da classe. Em tempo de execução do código fizemos isso de maneira separada da criação do objeto. Então em primeiro momento tínhamos um objeto sem nenhum atributo preenchido, agora temos o objeto com ambos os atributos preenchidos.

Chamando o Método do Objeto

Assim como podemos “chamar” o atributo do objeto através do ponto, podemos também “chamar” os métodos que esse objeto tem.

```
public class Main {

   public static void main(String[] args) {
       Contador contador = new Contador();
       contador.primeiraParcela = 13;
       contador.segundaParcela = 2;

       int total = contador.somarAsParcelas();
       System.out.println(total);

   }
}
```

Como podemos ver no exemplo assim, chamamos o método de **somarAsParcelas** usando o ponto logo após o nome do nosso objeto. Como esse método retorna um número inteiro, podemos armazenar esse número em uma variável, que no caso chamamos de total, e depois usar ela para imprimir na tela para o usuário.

Conclusão

Com as classes podemos criar estruturas de dados que simulam objetos em nosso código, é através das classes que podemos dar as características em forma de atributos e comportamentos em forma de métodos para nossos objetos.

É sempre bom lembrar que cada classe deve ter um propósito para existir no nosso código, não existe um limite de classes para serem feitas, mas existe a questão de coerência na existência delas.