

Prática

Refatoração LojaMotos

A aplicação tem como objetivo fornecer os dados de certa moto na loja a partir do nome e marca.

- (1) Crie um projeto JAVA: LojaMotos
- (2) Crie a classe Honda no pacote entidades:

```
1 package entidades;
2
3 public class Honda {
4     public String nome;
5     public String cilindrada;
6     public String cor;
7     public Honda() {
8         nome = "CBX (HONDA)";
9         cilindrada = "500";
10        cor = "Azul";
11    }
12 }
```

- (3) Crie a classe Yamaha no pacote entidades:

(3) Crie a classe **Yamaha** no pacote entidades:

```
1 package entidades;
2
3 public class Yamaha {
4     public String nome;
5     public String cilindrada;
6     public String cor;
7     public Yamaha() {
8         nome = "RD (YAMAHA)";
9         cilindrada = "350";
10        cor = "Vermelho";
11    }
12 }
```

- (4) Crie a classe Yamaha no pacote entidades:

(4) Crie a classe **Suzuki** no pacote entidades:

```
1 package entidades;
2
3 public class Suzuki {
4     public String nome;
5     public String cilindrada;
6     public String cor;
7     public Suzuki() {
8         nome = "DRZ (SUZUKI)";
9         cilindrada = "400";
10        cor = "Amarelo";
11    }
12 }
```

5) Crie a classe principal (Programa) no pacote “aplicação”

(6) Codifique/copie o programa:

```
package Aplicacao;
import javax.swing.JOptionPane;
import Entidades.Honda;
import Entidades.Suzuki;
import Entidades.Yamaha;
public class Motos {

    public static void main(String[] args) {

        while (true)
        {
            try
            {
                String nome = JOptionPane.showInputDialog("Informe a marca
[Cancelar] para sair: ");
                if (nome.equalsIgnoreCase("Honda")) {
                    Honda moto = new Honda();
                    JOptionPane.showMessageDialog(null, "\nNome: " +
moto.nome
                    + "\nCilindrada: " + moto.cilindradas + "\nCor: "
                    + moto.cor, "Dados da moto",
                    JOptionPane.CLOSED_OPTION);
                } else if (nome.equalsIgnoreCase("Yamaha")) {
                    Yamaha moto = new Yamaha();
                    JOptionPane.showMessageDialog(null, "\nNome: " +
moto.nome
                    + "\nCilindrada: " + moto.cilindradas + "\nCor: "
                    + moto.cor, "Dados da moto",
                    JOptionPane.CLOSED_OPTION);
                } else if (nome.equalsIgnoreCase("Suzuki"))
                {
                    Suzuki moto = new Suzuki();
                    JOptionPane.showMessageDialog(null, "\nNome: " +
moto.nome
                    + "\nCilindrada: " + moto.cilindradas + "\nCor: "
                    + moto.cor, "Dados da moto",
                    JOptionPane.CLOSED_OPTION);
                }
            } catch (Exception e)
            {
                break;
            }
        }

        System.out.println("\nPROGRAMA FINALIZADO!");
    }
}
```

(8) Fazer a refatoração de extração de superclasse criando a classe Moto. Utilize a ferramenta refatorar do Eclipse. Crie a super classe e em seguida use Pull UP.

(9) Se necessário, altere a criação dos objetos Honda, Yamaha e Suzuki para que usem as abstração, “ao invés” de fazer a criação das classes concretas, por exemplo:

Moto **moto** = **new** Honda(); => “ao invés” de => Honda **moto** = **new** Honda();

Moto **moto** = **new** Yamaha(); => “ao invés” de => Yamaha **moto** = **new** Honda();

Moto **moto** = **new** Suzuki(); => “ao invés” de => Suzuki **moto** = **new** Honda();

Dessa forma aplicamos o princípio da inversão de dependência:

“Depende de abstrações. Não depende de classes concretas.”

Freeman, Eric e Beth, Head First Design Patterns, 2007, pág. 124

(10) Faça a refatoração para evitar código redundante da impressão dos dados:

```
JOptionPane.showMessageDialog(null, "\nNome: " + moto.nome  
+ "\nCilindrada: " + moto.cilindrada + "\nCor: " + moto.cor, "Dados da moto", JOptionPane.CLOSED_OPTION);
```

Crie o método: “imprimirDadosMoto”

A seguir o diagrama de classe construído de forma correta.

