

```

package Locadora;

import java.time.LocalDate;

public class Moto extends Veiculo {
    private int      cilindradas;
    private String    tipo_carenagem;
    private String    tipo_uso;
    private String    tipo_freio;

    public int getCilindradas() {
        return cilindradas;
    }

    public void setCilindradas(int cilindradas) throws Exception{
        if(cilindradas < 50)
            throw new Exception("Cilindrada deve ser maior que 50cc!");
        this.cilindradas = cilindradas;
    }

    public String getTipo_carenagem() {
        return tipo_carenagem;
    }

    public void setTipo_carenagem(String tipo_carenagem) throws Exception {
        tipo_carenagem = tipo_carenagem.toLowerCase();

        if(tipo_carenagem.equals("") || tipo_carenagem.isEmpty())
        {
            throw new Exception("O tipo de carenagem não pode ser vazio");
        }
        else if(!tipo_carenagem.equals("carenada") && !
tipo_carenagem.equals("naked") && !tipo_carenagem.equals("trial"))
        {
            throw new Exception("O tipo de carenagem deve ser carenada, naked
ou trial.");
        }
        this.tipo_carenagem = tipo_carenagem;
    }

    public String getTipo_uso() {
        return tipo_uso;
    }

    public void setTipo_uso(String tipo_uso) throws Exception {
        tipo_uso = tipo_uso.toLowerCase();

        if(tipo_uso.equals("") || tipo_uso.isEmpty())
        {
            throw new Exception("O tipo de uso não pode ser vazio");
        }
        else if(!tipo_uso.equals("trilha") && !tipo_uso.equals("urbano") && !
tipo_uso.equals("pista"))
        {
            throw new Exception("O tipo de uso deve ser trilha, urbano ou
pista.");
        }
    }
}

```

```

        this.tipo_uso = tipo_uso;
    }

    public String getTipo_freio() {
        return tipo_freio;
    }

    public void setTipo_freio(String tipo_freio) throws Exception {

        tipo_freio = tipo_freio.toLowerCase();

        if(tipo_freio.equals("") || tipo_freio.isEmpty())
        {
            throw new Exception("O tipo de freio não pode ser vazio");
        }
        else if(!tipo_freio.equals("freio a disco") && !
tipo_freio.equals("freio a tambor"))
        {
            throw new Exception("O tipo de freio deve ser freio a disco ou
freio a tambor.");
        }
        this.tipo_freio = tipo_freio;
    }

    public Moto(String modelo, String placa, String montadora, String cor,
LocalDate data_fabricacao,
                String tipo_combustivel, int cilindradas, String tipo_carenagem,
String tipo_uso, String tipo_freio) {
        super(modelo, placa, montadora, cor, data_fabricacao,
tipo_combustivel);
        this.cilindradas = cilindradas;
        this.tipo_carenagem = tipo_carenagem;
        this.tipo_uso = tipo_uso;
        this.tipo_freio = tipo_freio;
    }

    public String descricao(Moto moto)
    {
        return new String("Modelo: "+moto.getModelo()+"", Placa:
"+moto.getPlaca()+"", Montadora: "+moto.getMontadora()+"", Cor: "+moto.getCor()+"",
Data de fabricacao: "+moto.getData_fabricacao()+"", Tipo de combustivel:
"+moto.getTipo_combustivel()+"", "
                + " Cilindradas: "+moto.getCilindradas()+"", Tipo de
Carenagem:"+moto.getTipo_carenagem()+"", Tipo do uso: "+moto.getTipo_uso()+"", Tipo
de freio: "+moto.getTipo_freio() +", Placa valida:"+moto.getPlaca_valida());
    }

    public String retornaInsert(Moto moto)
    {
        return new String("INSERT INTO carro (modelo, placa, montadora, cor,
data_fabricacao, tipo_combustivel, cilindradas, tipo_carenagem, tipo_uso,
tipo_freio)"
                + " VALUES (" +moto.getModelo()+", "+moto.getPlaca()+",
"+moto.getMontadora()+", "+moto.getCor()+", "+moto.getData_fabricacao()+",
"+moto.getTipo_combustivel()+", "
                + moto.getCilindradas()+", "+moto.getTipo_carenagem()+",
"+moto.getTipo_uso()+", "+moto.getTipo_freio()+");");
    }

```

}