UNIVERSIDADE PAULISTA

BACHAREL EM CIÊNCIA DA COMPUTAÇÃO

TRABALHO DE LINGUAGEM ORIENTADA A OBJETOS – 3º SEMESTRE

Autores:	**RA:**
Diego Freire de Almeida	N8059D2
Kaiky de Lara Sales	N9218H8
Vitor dos Santos Rosa	G521CE3
Nicolas Pimenta	N863579
Leonardo de Souza Rodrigues	F344HB2

Jundiaí/SP

2023

Sumário

1. Descrição do Projeto

Com o intuito de colocar em prática o que aprendemos nas aulas de Linguagem de Programação Orientada a Objetos utilizando a Linguagem Java e as técnicas apresentadas iremos construir um programa que terá as principais operações de um programa prático, sendo elas: Gravação, Consulta, Atualização e Exclusão de dados sem a utilização de um banco de dados, através da técnica de serialização.

O sistema tem como contexto uma concessionária que necessita de um programa simples para a organização dos automóveis que possui para venda, sendo eles de três tipos: Carros, Caminhões e Motos. Todos eles possuem características comuns e únicas para cada tipo, sendo todas elas cadastradas pelo usuário que estará responsável por esta atividade. Caso o utilitário deseje, a alteração de características não fundamentais para a origem do automóvel poderão ser realizadas, como cor e preço do veículo e consequentemente a sua consulta e exclusão será possível.

2. Objetivo

O sistema tem objetivo de propor uma forma de facilitar e agilizar o cadastro, consulta e armazenamento dos dados, de forma eficaz, rápida, simples e intuitiva, prevendo e tratando potenciais erros que o usuário possa realizar na utilização de um sistema similar ao que estamos desenvolvendo, por meio de técnicas relativamente simples, aplicando todos os conceitos apresentados nas aulas de Linguagem de Programação Orientada a Objetos (Herança, Sobrecarga, Sobreposição, entre outros).

3. Aplicação do Sistema

Para a aplicação do sistema, usamos o exemplo de uma concessionária com o intuito de visualizar melhor o contexto em que o programa seria utilizado. Com uma interface simplificada, o objetivo é que o usuário tenha um controle rápido e organizado dos seus veículos, por meio das diversas funções dentro do programa, sendo elas:

- **Cadastro**: Através da operação de cadastro, o sistema irá coletar todas as características do veículo e então irá guardar todos estes atributos em um arquivo serializado na própria raiz do programa.
- **Consulta**: Com a opção de consulta, o programa localiza o arquivo serializado através da placa do veículo (no qual será o nome do arquivo serializado) efetuando a leitura do arquivo, e mostrando as características do veículo armazenadas dentro dele.
- **Atualização**: Caso o usuário necessite, ele poderá realizar a atualização de alguns dados dentro do arquivo já serializado. Alguns atributos não serão possíveis mudar por conta de conflitos com a realidade (ex: não podemos alterar o modelo de um carro depois que ele foi cadastrado), gerando a necessidade de realizar um novo cadastro caso um dos atributos cruciais esteja errado.
- **Exclusão**: E por último, teremos a opção de exclusão. Esta opção serve para quando o usuário não quer mais armazenar o veículo, e funciona de forma similar com a opção de consulta, onde o programa irá solicitar a placa do veículo, localizar o arquivo correspondente e então excluí-lo.

Todas as funções possuem tratamentos de erros específicos para cada caso levando em conta o escopo em que se encontram, tentando evitar ao máximo erros humanos ou falhas que possam ocorrer na utilização do programa.

Abaixo, consta o funcionamento do programa em si com o intuito de demonstrar a aplicação prática do sistema.

Menu Principal:

```
### Menu Principal - Concessionaria ###
1. Cadastrar um Veículo
2. Consultar um Veículo
3. Atualizar dados de um Veiculo
4. Excluir um Veiculo
0. Sair da aplicacao
Digite uma opcao:
```

Opção 1: Cadastrar Veículo

```
### 1. Cadastrar um Veiculo ###
1. Carro
2. Caminhao
3. Moto
0. Cancelar
Tipo: 1
Digite o chassi do Veiculo: MPE-123
Digite a cor do Veiculo: Preto
Digite o modelo do Veiculo: Mustang
Digite a placa do Veiculo: QWE123
Digite o ano do Veiculo: 2010
Digite o valor do Veiculo: 58000
Digite a quantidade de portas: 4
Carro Criado!
Objeto Carro, Salvo!
```

Opção 2: Consultar Veículo

```
### 2. Localizando um Automóvel ###
Qual o tipo do veículo?
1. Carro
2. Caminhao
3. Moto
0. Voltar
Tipo: 1
Digite a placa do veiculo: QWE123
### Dados do Veículo ###
Carro:
Chassi= MPE-123
Cor= Preto
Modelo= Mustang
Placa= QWE123
Ano= 2010
Quantidade de Portas= 4
Valor= R$58000.0
```

Opção 3: Atualizando um Veículo

```
### 3. Atualizando um Automóvel ###
Qual o tipo do veículo?
1. Carro
2. Caminhao
3. Moto
0. Voltar
Tipo: 1
Digite a placa do veiculo: QWE123
Dados do veiculo QWE123
### Dados do Veículo ###
Carro:
Chassi= MPE-123
Cor= Preto
Modelo= Mustang
Placa= QWE123
Ano= 2010
Quantidade de Portas= 4
Valor= R$58000.0
### Digite os novos dados:
Digite a cor do Veiculo: Vermelho
Digite o valor do Veiculo: 60000
Objeto Carro, Salvo!
Veiculo QWE123 alterado!
### Novos dados:
### Dados do Veículo ###
Carro:
Chassi= MPE-123
Cor= Vermelho
Modelo= Mustang
Placa= QWE123
Ano= 2010
Quantidade de Portas= 4
Valor= R$60000.0
```

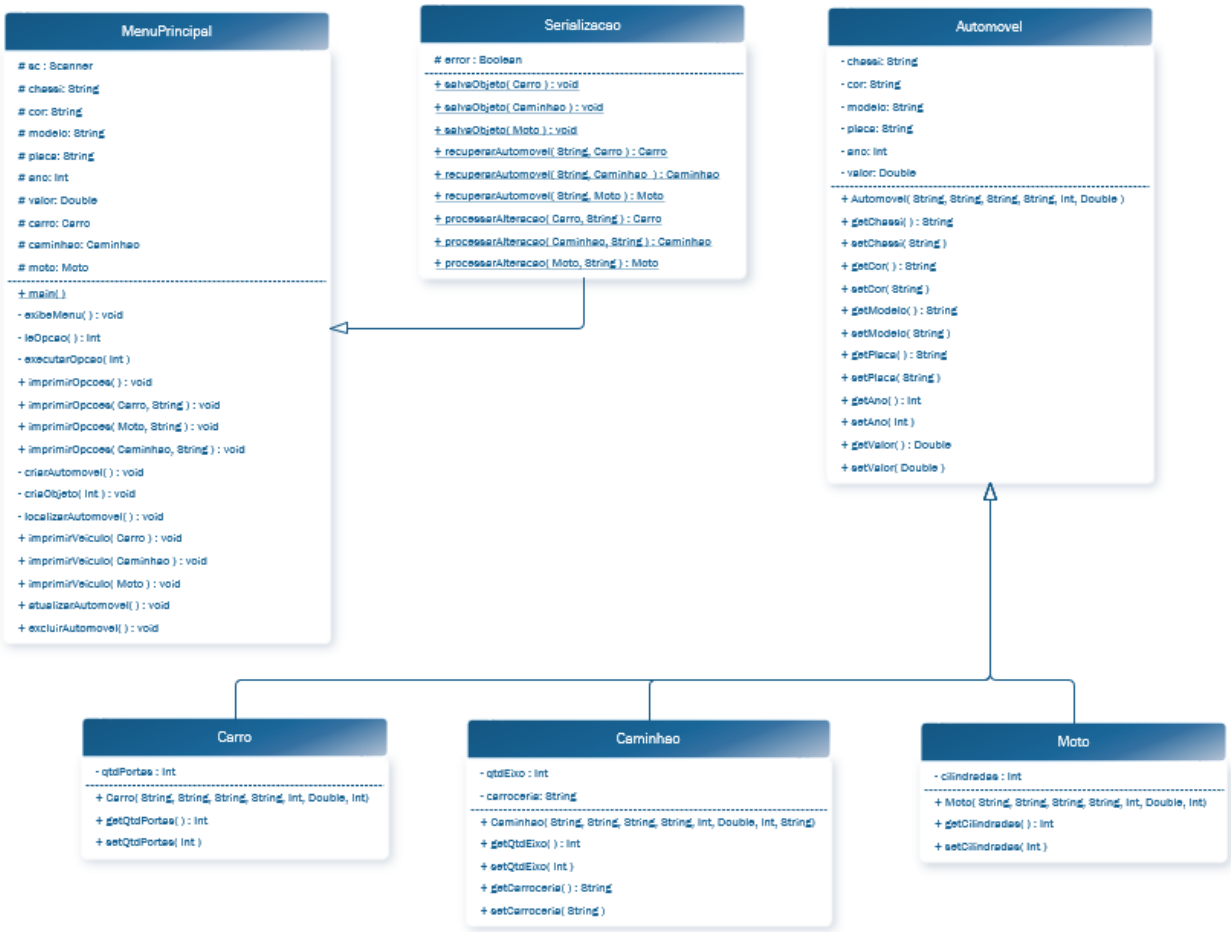
Opção 4: Exclusão de um Veículo

```
### 4. Excluir veículo ###
Digite a placa do veículo: QWE123
Veiculo QWE123 excluído com sucesso!
```

4. Diagrama de Classes

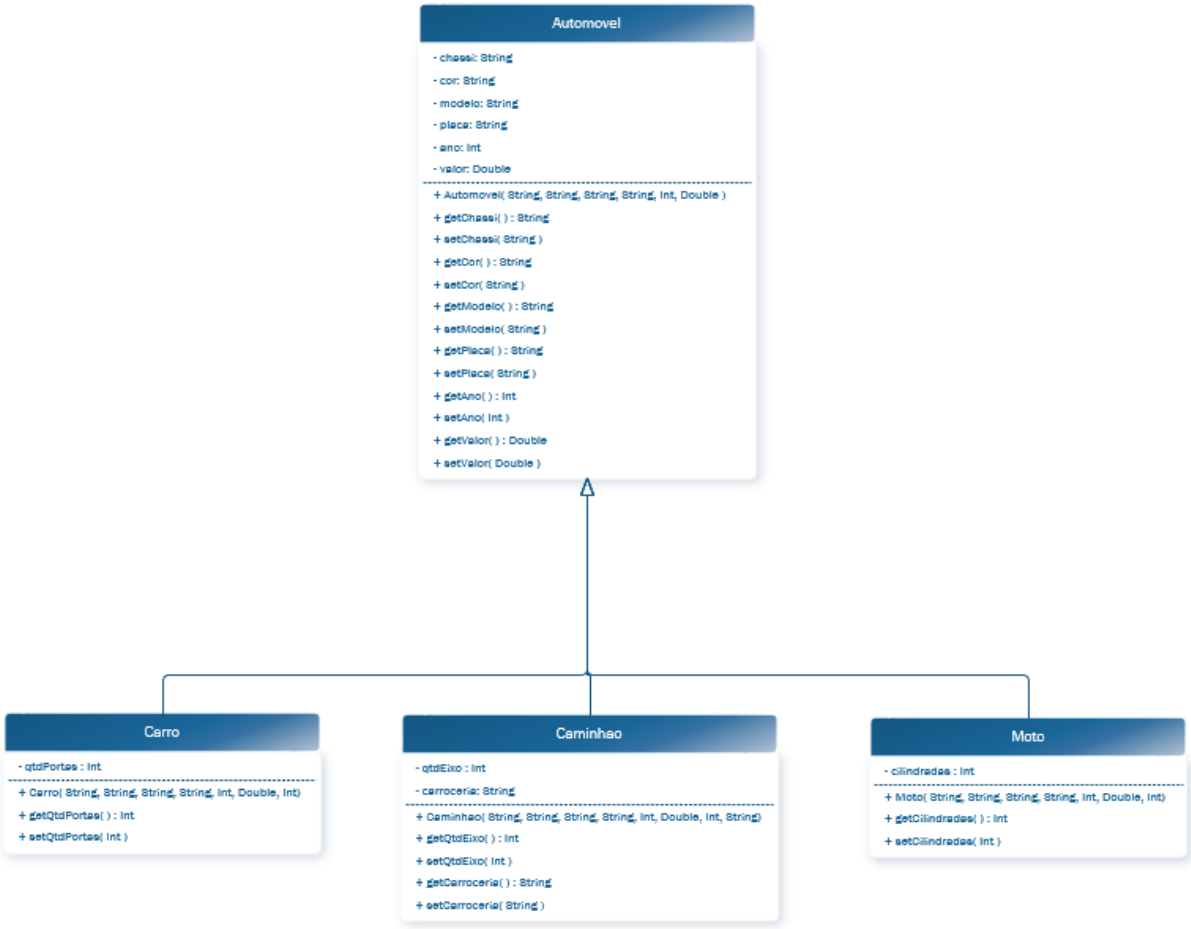
A seguir encontra-se o diagrama UML do programa, no qual pode se observar todas as heranças, métodos, acessos e encapsulamentos dos diversos métodos e atributos que se encontram no programa. Importante notar que as variáveis da classe “Menu Principal”, existem apenas de passagem, onde serão utilizadas para a criação dos objetos nas classes do domínio (Carro, Caminhao ou Moto).

4.1. Diagrama Completo



4.2. Classes Principais

4.3. Classes de Domínio



5. Código Fonte do Sistema

O código fonte do programa encontra-se documentado e organizado de acordo com os diagramas, seguindo a sua hierarquia e com comentários para melhor compreensão do código.

5.1. MenuPrincipal

```
package J12B.LPOO_Trabalho;

import java.io.File;
import java.util.Scanner;

public class MenuPrincipal {
    private static Scanner sc = new Scanner(System.in);
    private static Carro carro;
    private static Caminhao caminhao;
    private static Moto moto;

    public static void main(String[] args){
```

```
        while (true){
            // 1. Exibe o menu
            exibeMenu();
            // 2. Lê opção do usuário
            int opcao = leOpcao();
            // 3. Executa a opção do usuário
            executarOpcao(opcao);
        }
    }

    private static void exibeMenu() {
        System.out.println("### Menu Principal - Concessionaria ###");
        System.out.println("1. Cadastrar um Veículo");
        System.out.println("2. Consultar um Veículo");
        System.out.println("3. Atualizar dados de um Veiculo");
        System.out.println("4. Excluir um Veiculo");
        System.out.println("0. Sair da aplicacao");
    }

    private static int leOpcao() {
        int op = -1;
        while (op < 0 || op > 99){
            System.out.print("Digite uma opcao: ");
            op = sc.nextInt();
        }
        sc.nextLine();
        return op;
    }

    private static void executarOpcao(int opcao) {
        switch(opcao){
            case 1:
                criarAutomovel();
                break;
            case 2:
                localizarAutomovel();
                break;
            case 3:
                atualizarAutomovel();
                break;
            case 4:
                excluirAutomovel();
                break;
            case 0:
                System.out.println("Aplicacao Encerrada.");
                System.exit(0);
                break;
```

```
            default:
                System.out.println("Opcao invalida!");
        }
    }

    public static void imprimirOpcoes(){
        System.out.print("Digite o chassi do Veiculo: ");
        Serializacao.setChassi(sc.next());
        System.out.print("Digite a cor do Veiculo: ");
        Serializacao.setCor(sc.next());
        System.out.print("Digite o modelo do Veiculo: ");
        Serializacao.setModelo(sc.next());
        System.out.print("Digite a placa do Veiculo: ");
        Serializacao.setPlaca(sc.next());
        System.out.print("Digite o ano do Veiculo: ");
        Serializacao.setAno(sc.nextInt());
        System.out.print("Digite o valor do Veiculo: ");
        Serializacao.setValor(sc.nextDouble());
    }

    public static void imprimirOpcoes(Carro veiculo, String pl){
        System.out.print("Digite a cor do Veiculo: ");
        Serializacao.setCor(sc.next());
        Serializacao.setPlaca(pl);
        System.out.print("Digite o valor do Veiculo: ");
        Serializacao.setValor(sc.nextDouble());
    }

    public static void imprimirOpcoes(Moto veiculo, String pl){
        System.out.print("Digite a cor do Veiculo: ");
        Serializacao.setCor(sc.next());
        Serializacao.setPlaca(pl);
        System.out.print("Digite o valor do Veiculo: ");
        Serializacao.setValor(sc.nextDouble());
    }

    public static void imprimirOpcoes(Caminhao veiculo, String pl){
        System.out.print("Digite a cor do Veiculo: ");
        Serializacao.setCor(sc.next());
        Serializacao.setPlaca(pl);
        System.out.print("Digite o valor do Veiculo: ");
        Serializacao.setValor(sc.nextDouble());
    }


    // Opção 1
    private static void criarAutomovel() {
```

```
        System.out.println("### 1. Cadastrar um Veiculo ###");
        System.out.println("1. Carro");
        System.out.println("2. Caminhao");
        System.out.println("3. Moto");
        System.out.println("0. Cancelar");

        int op = -1;
        while(op < 0){
            System.out.print("Tipo: ");
            op = sc.nextInt();
            if (op<=0 || op >3){
                System.out.println("Processo Cancelado!");
                break;
            }
            criaObjeto(op);
        }
    }

    private static void criaObjeto(int op){
        imprimirOpcoes();

        switch(op){
            case 1:
                System.out.print("Digite a quantidade de portas: ");
                int qtdPortas = sc.nextInt();
                Carro carro = new
Carro(Serializacao.getChassi(),Serializacao.getCor(),Serializacao.getModelo(),Serializacao.
getPlaca(),Serializacao.getAno(),Serializacao.getValor(),qtdPortas);
                System.out.println("Carro Criado!");
                Serializacao.salvaObjeto(carro);
                break;
            case 2:
                System.out.print("Digite a quantidade de eixos: ");
                int qtdEixo = sc.nextInt();
                System.out.print("Digite o tipo de carroceria: ");
                String carroceria = sc.next();
                Caminhao caminhao = new
Caminhao(Serializacao.getChassi(),Serializacao.getCor(),Serializacao.getModelo(),Serializac
ao.getPlaca(),Serializacao.getAno(),Serializacao.getValor(),qtdEixo,carroceria);
                System.out.println("Caminhao Criado!");
                Serializacao.salvaObjeto(caminhao);
                break;
            case 3:
                System.out.print("Digite a quantidade de cilindradas: ");
                int cilindradas = sc.nextInt();
```

```
                Moto moto = new
Moto(Serializacao.getChassi(),Serializacao.getCor(),Serializacao.getModelo(),Serializacao.g
etPlaca(),Serializacao.getAno(),Serializacao.getValor(),cilindradas);
                System.out.println("Moto Criada!");
                Serializacao.salvaObjeto(moto);
                break;
            default:
                System.out.println("Opcao Invalida!");
        }
    }

    // Opção 2

    private static void localizarAutomovel() {
        System.out.println("### 2. Localizando um Automóvel ###");
        System.out.println("Qual o tipo do veículo? ");
        System.out.println("1. Carro");
        System.out.println("2. Caminhao");
        System.out.println("3. Moto");
        System.out.println("0. Voltar");
        int op = -1;
        while(op < 0){
            System.out.print("Tipo: ");
            op = sc.nextInt();
            if (op<=0 || op >3){
                System.out.println("Processo Cancelado!");
                break;
            }
        }
        System.out.print("Digite a placa do veiculo: ");
        String placa = sc.next();

        switch(op){
            case 1:
                carro = Serializacao.recuperarAutomovel(placa,carro);
                imprimirVeiculo(carro);
                break;
            case 2:
                caminhao = Serializacao.recuperarAutomovel(placa,caminhao);
                imprimirVeiculo(caminhao);
                break;
            case 3:
                moto = Serializacao.recuperarAutomovel(placa,moto);
                imprimirVeiculo(moto);
                break;
            case 0:
                break;
```

```
    }
}


// Metodos para a impressao do objeto | sobrecarregado
public static void imprimirVeiculo(Carro veiculo){
    try{
        System.out.println("### Dados do Veículo ###");
        System.out.println(veiculo.toString());
        } catch(Throwable e) {
            System.out.print("");
        }
}
public static void imprimirVeiculo(Caminhao veiculo){
    try{
        System.out.println("### Dados do Veículo ###");
        System.out.println(veiculo.toString());
        } catch(Throwable e) {
            System.out.print("");
        }
}
public static void imprimirVeiculo(Moto veiculo){
    try{
        System.out.println("### Dados do Veículo ###");
        System.out.println(veiculo.toString());
        } catch(Throwable e) {
            System.out.print("");
        }
}

// Opção 3

public static void atualizarAutomovel(){
    System.out.println("### 3. Atualizando um Automóvel ###");
    System.out.println("Qual o tipo do veículo? ");
    System.out.println("1. Carro");
    System.out.println("2. Caminhao");
    System.out.println("3. Moto");
    System.out.println("0. Voltar");
    int op = -1;
    while(op < 0){
        System.out.print("Tipo: ");
        op = sc.nextInt();
        if (op<=0 || op >3){
            System.out.println("Processo Cancelado!");
            break;
        }
```

```
        }
        if(op !=0){
            System.out.print("Digite a placa do veiculo: ");
            String placa = sc.next();

            switch(op){
                case 1:
                    carro = Serializacao.recuperarAutomovel(placa,carro);
                    if (Serializacao.isError()) break;
                    System.out.println("Dados do veiculo " + placa);
                    imprimirVeiculo(carro);
                    System.out.println("### Digite os novos dados: ");
                    Serializacao.processarAlteracao(carro, placa);
                    System.out.println("### Novos dados: ");
                    imprimirVeiculo(carro);
                    break;
                case 2:
                    caminhao = Serializacao.recuperarAutomovel(placa,caminhao);
                    if (Serializacao.isError()) break;
                    System.out.println("Dados do veiculo " + placa);
                    imprimirVeiculo(caminhao);
                    System.out.println("### Digite os novos dados: ");
                    Serializacao.processarAlteracao(caminhao, placa);
                    System.out.println("### Novos dados: ");
                    imprimirVeiculo(caminhao);
                    break;
                case 3:
                    moto = Serializacao.recuperarAutomovel(placa,moto);
                    if (Serializacao.isError()) break;
                    System.out.println("Dados do veiculo " + placa);
                    imprimirVeiculo(moto);
                    System.out.println("### Digite os novos dados: ");
                    Serializacao.processarAlteracao(moto, placa);
                    System.out.println("### Novos dados: ");
                    imprimirVeiculo(moto);
                    break;
                case 0:
                    break;
            }
        }
    }

    // Opção 4

    public static void excluirAutomovel(){
        System.out.println("### 4. Excluir veículo ###");
        System.out.print("Digite a placa do veículo: ");
```

```
        String nome = sc.next();
        File arquivo = new File(nome + ".ser");
        if (arquivo.exists()){
            arquivo.delete();
            System.out.println("Veiculo "+ nome + " excluído com sucesso!");
        } else {
            System.out.println("Veículo Inexistente!");
        }
    }
}
```

5.2. Serializacao

```
package J12B.LPOO_Trabalho;

import java.io.FileInputStream;
import java.lang.Throwable ;
import java.lang.Exception;
import java.util.Scanner;
import java.io.ObjectOutputStream;
import java.io.FileOutputStream;
import java.io.ObjectInputStream;

public class Serializacao {

    private static Scanner sc = new Scanner(System.in);
    private static boolean error;

    private static String chassi, cor, modelo, placa;
    private static int ano;
    private static double valor;


    public static boolean isError() {
        return error;
    }

    public static void setError(boolean error) {
        Serializacao.error = error;
    }

    public static String getChassi() {
        return chassi;
    }

    public static void setChassi(String chassi) {
        Serializacao.chassi = chassi;
    }
```

```
    public static String getCor() {
        return cor;
    }

    public static void setCor(String cor) {
        Serializacao.cor = cor;
    }

    public static String getModelo() {
        return modelo;
    }

    public static void setModelo(String modelo) {
        Serializacao.modelo = modelo;
    }

    public static String getPlaca() {
        return placa;
    }

    public static void setPlaca(String placa) {
        Serializacao.placa = placa;
    }

    public static int getAno() {
        return ano;
    }

    public static void setAno(int ano) {
        Serializacao.ano = ano;
    }

    public static double getValor() {
        return valor;
    }

    public static void setValor(double valor) {
        Serializacao.valor = valor;
    }

    // Salvar objeto em .ser
    public static void salvaObjeto(Carro veiculo) {
        try {
            FileOutputStream fos = new FileOutputStream(veiculo.getPlaca()+".ser");
            ObjectOutputStream oos = new ObjectOutputStream(fos);
            oos.writeObject(veiculo);
            oos.close();
```

```
            System.out.println("Objeto Carro, Salvo!");
        } catch (Exception e){
            e.printStackTrace();
        }
    }

    public static void salvaObjeto(Caminhao veiculo) {
        try {
            FileOutputStream fos = new FileOutputStream(veiculo.getPlaca() + ".ser");
            ObjectOutputStream oos = new ObjectOutputStream(fos);
            oos.writeObject(veiculo);
            oos.close();
            System.out.println("Objeto Caminhao, Salvo!");
        } catch (Exception e){
            e.printStackTrace();
        }
    }

    public static void salvaObjeto(Moto veiculo) {
        try {
            FileOutputStream fos = new FileOutputStream(veiculo.getPlaca() + ".ser");
            ObjectOutputStream oos = new ObjectOutputStream(fos);
            oos.writeObject(veiculo);
            oos.close();
            System.out.println("Objeto Moto, Salvo!");
        } catch (Exception e){
            e.printStackTrace();
        }
    }

    // Recuperar objeto através do .ser

    public static Carro recuperarAutomovel(String placa, Carro veiculo) {
        try {
                FileInputStream fis = new FileInputStream(placa+".ser");
                ObjectInputStream ois = new ObjectInputStream(fis);
                veiculo = (Carro) ois.readObject();
                ois.close();
                error = false;
            } catch (Throwable e) {
                error = true;
                System.out.println("Nenhum carro com esta placa!");
            }
            return veiculo;
    }

    public static Caminhao recuperarAutomovel(String placa, Caminhao veiculo) {
```

```
        try {
            FileInputStream fis = new FileInputStream(placa+".ser");
            ObjectInputStream ois = new ObjectInputStream(fis);
            veiculo = (Caminhao) ois.readObject();
            ois.close();
            error = false;
        } catch (Throwable e) {
            error = true;
            System.out.println("Nenhum caminhao com esta placa!");
        }
        return veiculo;
    }

    public static Moto recuperarAutomovel(String placa, Moto veiculo) {
        try {
            FileInputStream fis = new FileInputStream(placa+".ser");
            ObjectInputStream ois = new ObjectInputStream(fis);
            veiculo = (Moto) ois.readObject();
            ois.close();
            error = false;
        } catch (Throwable e) {
            error = true;
            System.out.println("Nenhuma moto com esta placa!");
        }
        return veiculo;
    }

    // Processar a atualização

    public static Carro processarAlteracao(Carro veiculo, String placa){
        MenuPrincipal.imprimirOpcoes(veiculo,placa);
        veiculo.setCor(cor);
        veiculo.setPlaca(placa);
        veiculo.setValor(valor);
        salvaObjeto(veiculo);
        System.out.println("Veiculo "+veiculo.getPlaca() + " alterado!");
        return veiculo;
    }
    public static Caminhao processarAlteracao(Caminhao veiculo, String placa){
        MenuPrincipal.imprimirOpcoes(veiculo,placa);
        System.out.print("Digite o tipo de carroceria: ");
        String carroceria = sc.next();
        veiculo.setCarroceria(carroceria);
        veiculo.setCor(cor);
        veiculo.setPlaca(placa);
        veiculo.setValor(valor);
        salvaObjeto(veiculo);
```

```
        System.out.println("Veiculo "+veiculo.getPlaca() + " alterado!");
        return veiculo;
    }
    public static Moto processarAlteracao(Moto veiculo, String placa){
        MenuPrincipal.imprimirOpcoes(veiculo,placa);
        veiculo.setCor(cor);
        veiculo.setPlaca(placa);
        veiculo.setValor(valor);
        salvaObjeto(veiculo);
        System.out.println("Veiculo "+veiculo.getPlaca() + " alterado!");
        return veiculo;
    }
}
```

5.3. Automovel

```
package J12B.LPOO_Trabalho;

import java.io.Serializable;

public class Automovel implements Serializable{
    private String chassi, cor, modelo, placa;
    private int ano;
    private double valor;

    public Automovel(String chassi, String cor, String modelo, String placa, int ano,
double valor) {
        this.chassi = chassi;
        this.cor = cor;
        this.modelo = modelo;
        this.placa = placa;
        this.ano = ano;
        this.valor = valor;
    }

    public String getChassi() {return chassi;}
    public void setChassi(String chassi) {
        this.chassi = chassi;
    }

    public String getCor() {return cor;}
    public void setCor(String cor) {
        this.cor = cor;
    }

    public String getModelo() {return modelo;}
    public void setModelo(String modelo) {
        this.modelo = modelo;
    }
```

```
    public String getPlaca() {return placa;}
    public void setPlaca(String placa) {
        this.placa = placa;
    }

    public int getAno() {return ano;}
    public void setAno(int ano) {
        this.ano = ano;
    }

    public double getValor() {return valor;}

    public void setValor(double valor) {
        this.valor = valor;
    }

    @Override
    public String toString() {
        return "Automovel{" +
                "chassi=" + chassi +
                ", cor=" + cor +
                ", modelo=" + modelo +
                ", placa=" + placa +
                ", ano=" + ano +
                ", valor=" + valor +
                '}';
    }
}
```

5.4. Carro

```
package J12B.LPOO_Trabalho;
import java.io.Serializable;

public class Carro extends Automovel implements Serializable{

    private int qtdPortas;

    public Carro(
        String chassi,
        String cor,
        String modelo,
        String placa,
        int ano,
        double valor,
        int qtdPortas) {
        super(chassi, cor, modelo, placa, ano, valor);
        setQtdPortas(qtdPortas);
```

```
    }

    @Override
    public String toString() {
        return "Carro:" +
                "\nChassi= " + super.getChassi() +
                "\nCor= " + super.getCor() +
                "\nModelo= " + super.getModelo() +
                "\nPlaca= " + super.getPlaca() +
                "\nAno= " + super.getAno() +
                "\nQuantidade de Portas= " + getQtdPortas() +
                "\nValor= R$" + super.getValor();
    }

    public int getQtdPortas() {
        return qtdPortas;
    }

    public void setQtdPortas(int qtdPortas) {
        this.qtdPortas = qtdPortas;
    }
}
```

5.5. Caminhao

```
package J12B.LPOO_Trabalho;

import java.io.Serializable;

public class Caminhao extends Automovel implements Serializable{

    private int qtdEixo;
    private String carroceria;

    public Caminhao(
        String chassi,
        String cor,
        String modelo,
        String placa,
        int ano,
        double valor,
        int qtdEixo,
        String carro)
        {
        super(chassi, cor, modelo, placa, ano, valor);
        setQtdEixo(qtdEixo);
        setCarroceria(carro);
    }
```

```
    public int getQtdEixo() {
        return qtdEixo;
    }

    public void setQtdEixo(int qtdEixo) {
        this.qtdEixo = qtdEixo;
    }

    public String getCarroceria() {
        return carroceria;
    }

    public void setCarroceria(String carroceria) {
        this.carroceria = carroceria;
    }

    @Override
    public String toString() {
        return "Caminhao:" +
                "\nChassi= " + super.getChassi() +
                "\nCor= " + super.getCor() +
                "\nModelo= " + super.getModelo() +
                "\nPlaca= " + super.getPlaca() +
                "\nAno= " + super.getAno() +
                "\nQuantidade de Eixos= " + getQtdEixo() +
                "\nValor= R$" + super.getValor();
    }
}
```

5.6. Moto

```
package J12B.LPOO_Trabalho;

import java.io.Serializable;

public class Moto extends Automovel implements Serializable{

    private int cilindradas;

    public Moto(
        String chassi,
        String cor,
        String modelo,
        String placa,
        int ano,
        double valor,
        int cilindradas) {
        super(chassi, cor, modelo, placa, ano, valor);
        setCilindradas(cilindradas);
```

```
    }

    @Override
    public String toString() {
        return "Moto:" +
                "\nChassi= " + super.getChassi() +
                "\nCor= " + super.getCor() +
                "\nModelo= " + super.getModelo() +
                "\nPlaca= " + super.getPlaca() +
                "\nAno= " + super.getAno() +
                "\nCilindradas= " + getCilindradas() +
                "\nValor= R$" + super.getValor();
    }

    public int getCilindradas() {
        return cilindradas;
    }

    public void setCilindradas(int cilindradas) {
        this.cilindradas = cilindradas;
    }
}
```