**Introdução**

Este documento fornece uma visão detalhada do código-fonte para o sistema de gerenciamento de ônibus. O sistema permite adicionar, excluir e exibir informações sobre funcionários, ônibus e linhas de ônibus. O código é escrito em Java e utiliza a biblioteca Swing para a interface gráfica do usuário.

**Estrutura do Sistema de Ônibus**

O sistema é composto por várias classes, cada uma responsável por um aspecto diferente do gerenciamento de ônibus. As principais classes são **SistemaOnibus**, **Funcionario**, **Onibus**, e **Linha**.

**Classe SistemaOnibus**

A classe **SistemaOnibus** é a classe principal que gerencia a interface do usuário e as operações principais. Ela contém métodos para adicionar, excluir e exibir funcionários, ônibus e linhas de ônibus.

**Atributos:**

* **List<Funcionario> funcionarios**: Lista estática para armazenar os funcionários.
* **List<Onibus> onibusList**: Lista estática para armazenar os ônibus.
* **List<Linha> linhas**: Lista estática para armazenar as linhas de ônibus.
* **JTextArea textarea**: Área de texto para exibir informações.
* **JFrame frame**: Frame principal da interface gráfica.

**Métodos Principais:**

* **public static void main(String[] args)**: Método principal que configura a interface gráfica e apresenta o menu de opções ao usuário.
* **private static void adicionarFuncionario()**: Adiciona um novo funcionário ao sistema.
* **private static void adicionarOnibus()**: Adiciona um novo ônibus ao sistema.
* **private static void adicionarLinha()**: Adiciona uma nova linha de ônibus ao sistema.
* **private static void excluirFuncionario()**: Exclui um funcionário com base no CPF.
* **private static void excluirOnibus()**: Exclui um ônibus com base na placa.
* **private static void excluirLinha()**: Exclui uma linha de ônibus com base no ID.
* **private static void mostrarTodosFuncionarios()**: Exibe todos os funcionários na área de texto.
* **private static void mostrarTodosOnibus()**: Exibe todos os ônibus na área de texto.
* **private static void mostrarTodasLinhas()**: Exibe todas as linhas de ônibus na área de texto.

**Classe Funcionario**

A classe **Funcionario** representa um funcionário do sistema.

**Atributos:**

* **private String nome**: Nome do funcionário.
* **private String cpf**: CPF do funcionário.
* **private String cargo**: Cargo do funcionário.
* **private String id**: ID do funcionário.
* **private String idade**: Idade do funcionário.
* **private String turno**: Turno do funcionário.

**Métodos:**

* **public Funcionario(String nome, String cpf, String cargo, String id, String idade, String turno)**: Construtor da classe.
* **public String getCpf()**: Retorna o CPF do funcionário.
* **public String toString()**: Retorna uma representação textual do funcionário.

**Classe Onibus**

A classe **Onibus** representa um ônibus do sistema.

**Atributos:**

* **private String nome**: Nome do ônibus.
* **private String placa**: Placa do ônibus.
* **private String status**: Status do ônibus.
* **private int numeroDeSerie**: Número de série do ônibus.
* **private String possuiAr**: Indica se o ônibus possui ar condicionado.
* **private String idVeiculo**: ID do veículo.

**Métodos:**

* **public Onibus(String nome, String placa, String status, int numeroDeSerie, String possuiAr, String idVeiculo)**: Construtor da classe.
* **public String getPlaca()**: Retorna a placa do ônibus.
* **public String toString()**: Retorna uma representação textual do ônibus.

**Classe Linha**

A classe **Linha** representa uma linha de ônibus do sistema.

**Atributos:**

* **private String nome**: Nome da linha.
* **private String id**: ID da linha.

**Métodos:**

* **public Linha(String nome, String id)**: Construtor da classe.
* **public String getId()**: Retorna o ID da linha.
* **public String toString()**: Retorna uma representação textual da linha.

**Descrição das Funções**

**Adicionar Funcionário**

O método **adicionarFuncionario** solicita ao usuário as informações do novo funcionário (nome, CPF, cargo, ID, idade, turno) e adiciona um novo objeto **Funcionario** à lista de funcionários.

**Adicionar Ônibus**

O método **adicionarOnibus** solicita ao usuário as informações do novo ônibus (nome, placa, status, número de série, se possui ar condicionado, ID do veículo) e adiciona um novo objeto **Onibus** à lista de ônibus.

**Adicionar Linha de Ônibus**

O método **adicionarLinha** solicita ao usuário as informações da nova linha de ônibus (nome, ID) e adiciona um novo objeto **Linha** à lista de linhas.

**Excluir Funcionário**

O método **excluirFuncionario** solicita ao usuário o CPF do funcionário a ser excluído, procura o funcionário na lista e, se encontrado, o remove.

**Excluir Ônibus**

O método **excluirOnibus** solicita ao usuário a placa do ônibus a ser excluído, procura o ônibus na lista e, se encontrado, o remove.

**Excluir Linha de Ônibus**

O método **excluirLinha** solicita ao usuário o ID da linha a ser excluída, procura a linha na lista e, se encontrada, a remove.

**Mostrar Todos os Funcionários**

O método **mostrarTodosFuncionarios** exibe na área de texto todos os funcionários cadastrados no sistema.

**Mostrar Todos os Ônibus**

O método **mostrarTodosOnibus** exibe na área de texto todos os ônibus cadastrados no sistema.

**Mostrar Todas as Linhas**

O método **mostrarTodasLinhas** exibe na área de texto todas as linhas de ônibus cadastradas no sistema.

**Exemplo de Uso**

Ao iniciar o programa, o usuário é apresentado a um menu com as opções de adicionar, excluir e exibir funcionários, ônibus e linhas de ônibus. As escolhas são feitas através de uma interface gráfica baseada em diálogos do Swing. Abaixo está um exemplo de como a interface do usuário se comporta:

1. O usuário escolhe "Adicionar Funcionário".
2. O sistema solicita o nome, CPF, cargo, ID, idade e turno do novo funcionário.
3. O novo funcionário é adicionado à lista e a lista atualizada de funcionários é exibida na área de texto.
4. O usuário pode então optar por adicionar mais itens, excluir itens ou visualizar listas de itens existentes.