

My Project

Generated by Doxygen 1.8.13

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Main	??
Mundo	??
Veiculo	??
Caminhao	??
Carro	??
Moto	??

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

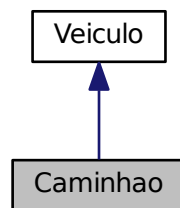
Caminhao	??
Carro	??
Main	??
Moto	??
Mundo	??
Veiculo	??

Chapter 3

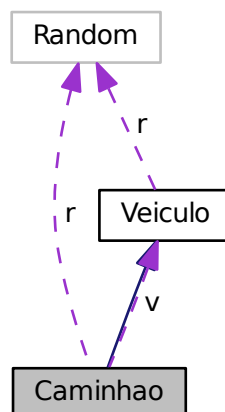
Class Documentation

3.1 Caminhao Class Reference

Inheritance diagram for Caminhao:



Collaboration diagram for Caminhao:



Public Member Functions

- [Caminhao](#) (int x, int y, int velocidade, String cor, boolean fabrica)
- void [move](#) ([Caminhao](#) d)
- int [verificaX](#) (int x)
- int [verificaY](#) (int y)

3.1.1 Detailed Description

Caminhão Classe que gera um veículo do tipo Caminhão

Author

Victor

See also

[Veiculo](#)

3.1.2 Constructor & Destructor Documentation

3.1.2.1 Caminhao()

```
Caminhao.Caminhao (  
    int x,  
    int y,  
    int velocidade,  
    String cor,  
    boolean fabrica ) [inline]
```

Construtor da classe Caminhão, que usa um super para chamar o construtor da classe Veículo

See also

[Veiculo](#)

Parameters

<i>x</i>	
<i>y</i>	
<i>velocidade</i>	
<i>cor</i>	
<i>fabrica</i>	

3.1.3 Member Function Documentation

3.1.3.1 move()

```
void Caminhao.move (
    Caminhao d ) [inline]
```

Função que movimenta o caminhão, recebendo um objeto da própria classe como parâmetro

Parameters

<i>d</i>	
----------	--

Gerando um número aleatório para movimentação do veículo em 4 direções possíveis

ifs para verificação do resultado obtido no random

3.1.3.2 verificaX()

```
int Caminhao.verificaX (
    int x ) [inline]
```

Função que verifica se o caminhão chegou ao limite do mapa em X e reseta a coordenada

Parameters

<i>x</i>	
----------	--

Returns

3.1.3.3 verificaY()

```
int Caminhao.verificaY (
    int y ) [inline]
```

Função que verifica se o caminhão chegou ao limite do mapa em Y e reseta a coordenada

Parameters

<i>y</i>	
----------	--

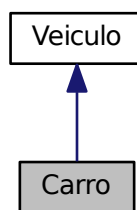
Returns

The documentation for this class was generated from the following file:

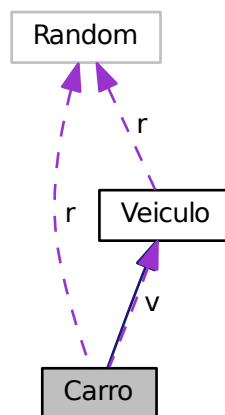
- Caminhao.java

3.2 Carro Class Reference

Inheritance diagram for Carro:



Collaboration diagram for Carro:



Public Member Functions

- **Carro** (int x, int y, int velocidade, String cor, boolean fabrica)
- void **move** (**Carro** c)
- int **verificaX** (int x)
- int **verificaY** (int y)

3.2.1 Constructor & Destructor Documentation

3.2.1.1 Carro()

```
Carro.Carro (
    int x,
    int y,
    int velocidade,
    String cor,
    boolean fabrica ) [inline]
```

Construtor da classe [Carro](#), que usa um super para chamar o construtor da classe Veículo

See also

[Veiculo](#)

Parameters

<i>x</i>	
<i>y</i>	
<i>velocidade</i>	
<i>cor</i>	
<i>fabrica</i>	

3.2.2 Member Function Documentation

3.2.2.1 move()

```
void Carro.move (
    Carro c ) [inline]
```

Função que movimenta o carro, recebendo um objeto da própria classe como parâmetro

Parameters

<i>c</i>	
----------	--

< Gerando um número aleatório para movimentação do veículo em 4 direções possíveis

ifs para verificação do resultado obtido no random

< Obtendo o valor atual de X

< Somando 2 (por ser a velocidade do carro) no valor atual de X

< Chamando a função andaX da classe Veículo para realizar a movimentação

Tudo se repete, da mesma forma que o if acima

3.2.2.2 verificaX()

```
int Carro.verificaX (
    int x ) [inline]
```

Função que verifica se o carro chegou ao limite do mapa em X e reseta a coordenada

Parameters

x	
---	--

Returns

3.2.2.3 verificaY()

```
int Carro.verificaY (
    int y ) [inline]
```

Função que verifica se o carro chegou ao limite do mapa em Y e reseta a coordenada

Parameters

y	
---	--

Returns

The documentation for this class was generated from the following file:

- Carro.java

3.3 Main Class Reference

Static Public Member Functions

- static void **main** (String[] args) throws InterruptedException

3.3.1 Detailed Description

Author

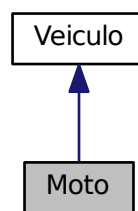
Victor

The documentation for this class was generated from the following file:

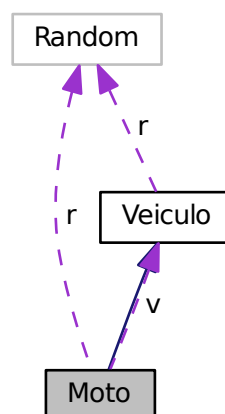
- Main.java

3.4 Moto Class Reference

Inheritance diagram for Moto:



Collaboration diagram for Moto:



Public Member Functions

- [Moto](#) (int x, int y, int velocidade, String cor, boolean fabrica)
- void [move](#) ([Moto](#) m)
- int [verificaX](#) (int x)
- int [verificaY](#) (int y)

3.4.1 Constructor & Destructor Documentation

3.4.1.1 Moto()

```
Moto.Moto (
    int x,
    int y,
    int velocidade,
    String cor,
    boolean fabrica ) [inline]
```

Construtor da classe [Moto](#), que usa um super para chamar o construtor da classe Veículo

See also

[Veiculo](#)

Parameters

<i>x</i>	
<i>y</i>	
<i>velocidade</i>	
<i>cor</i>	
<i>fabrica</i>	

3.4.2 Member Function Documentation

3.4.2.1 move()

```
void Moto.move (
    Moto m ) [inline]
```

Função que movimenta a moto, recebendo um objeto da própria classe como parâmetro

Parameters

<i>m</i>	
----------	--

Gerando um número aleatório para movimentação do veículo em 4 direções possíveis

ifs para verificação do resultado obtido no random

3.4.2.2 verificaX()

```
int Moto.verificaX (
    int x ) [inline]
```

Função que verifica se a moto chegou ao limite do mapa em X e reseta a coordenada

Parameters

x	
---	--

Returns

3.4.2.3 verificaY()

```
int Moto.verificaY (
    int y ) [inline]
```

Função que verifica se a moto chegou ao limite do mapa em Y e reseta a coordenada

Parameters

y	
---	--

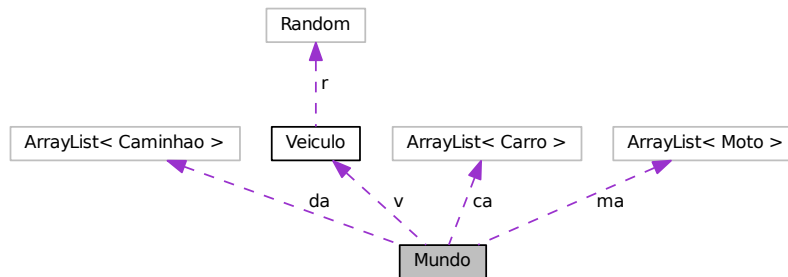
Returns

The documentation for this class was generated from the following file:

- Moto.java

3.5 Mundo Class Reference

Collaboration diagram for Mundo:



Public Member Functions

- void [geraVeiculos](#) ()
- void [refazMapa](#) ()
- void [desenhaMundo](#) ()
- void [atualizaMundo](#) ()
- void [detectaColisao](#) ()
- void [geraVeiculo](#) ()

Public Attributes

- int **mapa** [][]

3.5.1 Member Function Documentation

3.5.1.1 atualizaMundo()

```
void Mundo.atualizaMundo ( ) [inline]
```

Função que atualiza o mundo fazendo os veículos se moverem

3.5.1.2 desenhaMundo()

```
void Mundo.desenhaMundo ( ) [inline]
```

Função que desenha o mundo Contadores para exibir quantidade de veículos

Desenhando o mapa com base no que foi feito

Imprimindo a legenda

Contadores de veículos

3.5.1.3 detectaColisao()

```
void Mundo.detectaColisao ( ) [inline]
```

Função que detecta colisão Colisão entre carros apenas

Essa parte verifica se o veículo é ele mesmo, e não o remove

Colisão entre caminhões apenas

Colisão entre motos apenas

Colisão entre caminhão e carro

Colisão entre caminhão e moto

Colisão entre carro e moto

3.5.1.4 geraVeiculo()

```
void Mundo.geraVeiculo ( ) [inline]
```

Função que gera um veículo se passarem em fábricas Já que o veículo passou numa fábrica, ele fica impossibilitado de gerar novamente caso não saia da fabrica

3.5.1.5 geraVeiculos()

```
void Mundo.geraVeiculos ( ) [inline]
```

Função que gera veículos aleatoriamente < Verificando se o carro foi gerado onde há uma fábrica, se sim, trocando-o

3.5.1.6 refazMapa()

```
void Mundo.refazMapa ( ) [inline]
```

Função que zera o mapa, acabando com o rastro que ficaria caso o mapa não fosse zerado Esse é um comentário a parte, mas que achei legal estar na documentação: Parte do código que alteraria a situação de um veículo que não está mais em uma fábrica para false Essa parte foi comentada porém não removida pois faz a quantidade de veículos aumentar ao invés de zerar, como no vídeo de exemplo Apesar de ser o mais lógico colocar um veículo fora de uma fábrica como false, decidi remover do código final Removi pois o funcionamento do vídeo de exemplo se assemelha mais com o código funcionando sem essa parte

```
for(int a = 0; a < ca.size(); a++) { int xa = ca.get(a).getX(); int ya = ca.get(a).getY(); for(int i = 0; i<30; i++) { for(int j = 0; j < 60; j++) { if(i == xa && j == ya) { if(mapa[i][j] != 2 && ca.get(a).getFabrica() == true) { ca.get(a).setFabrica(false); } } } }
```

```
for(int a = 0; a < da.size(); a++) { int xa = da.get(a).getX(); int ya = da.get(a).getY(); for(int i = 0; i<30; i++) { for(int j = 0; j < 60; j++) { if(i == xa && j == ya) { if(mapa[i][j] != 2 && da.get(a).getFabrica() == true) { da.get(a).setFabrica(false); } } } }
```

```
for(int a = 0; a < da.size(); a++) { int xa = da.get(a).getX(); int ya = da.get(a).getY(); for(int i = 0; i<30; i++) { for(int j = 0; j < 60; j++) { if(i == xa && j == ya) { if(mapa[i][j] != 2 && da.get(a).getFabrica() == true) { da.get(a).setFabrica(false); } } } }
```

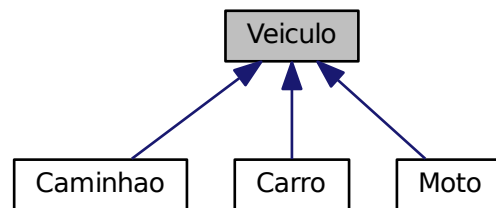
Adicionando os novos veículos no mapa

Obtendo as coordenadas do veículo

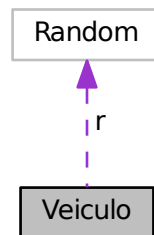
SObscrevendo 2 caso seja uma fábrica

3.6 Veiculo Class Reference

Inheritance diagram for Veiculo:



Collaboration diagram for Veiculo:



Public Member Functions

- [Veiculo](#) ()
Função random, utilizada para gerar posições aleatórias.
- [Veiculo](#) (int x, int y, int velocidade, String cor, boolean fabrica)
- int [setX](#) ()
- void [andaX](#) (int x)
- void [andaY](#) (int y)
- int [setY](#) ()
- int [getX](#) ()
- int [getY](#) ()
- int [getVelocidade](#) ()
- String [getCor](#) ()
- void [setFabrica](#) (boolean status)
- boolean [getFabrica](#) ()

3.6.1 Constructor & Destructor Documentation

3.6.1.1 Veiculo() [1/2]

```
Veiculo.Veiculo ( ) [inline]
```

Função random, utilizada para gerar posições aleatórias.

Construtor da classe Veículo Inicializa as variáveis de veículo

3.6.1.2 Veiculo() [2/2]

```
Veiculo.Veiculo (
    int x,
    int y,
    int velocidade,
    String cor,
    boolean fabrica ) [inline]
```

Construtor da classe Veículo Cria veículos com variáveis que são recebidas na chamada das funções de cada veículo específico

Parameters

<i>x</i>	
<i>y</i>	
<i>velocidade</i>	
<i>cor</i>	
<i>fabrica</i>	

3.6.2 Member Function Documentation

3.6.2.1 andaX()

```
void Veiculo.andax (
    int x ) [inline]
```

Altera o valor de X de um veículo com base no que será passado dentro de cada veículo específico

Parameters

<i>x</i>	
----------	--

3.6.2.2 andaY()

```
void Veiculo.anday (
    int y ) [inline]
```

Altera o valor de Y de um veículo com base no que será passado dentro de cada veículo específico

Parameters

y	
---	--

3.6.2.3 getCor()

```
String Veiculo.getCor ( ) [inline]
```

Returns

A cor do veículo

3.6.2.4 getFabrica()

```
boolean Veiculo.getFabrica ( ) [inline]
```

Returns

Se um veículo está ou não em uma fábrica

3.6.2.5 getVelocidade()

```
int Veiculo.getVelocidade ( ) [inline]
```

Returns

A velocidade do veículo

3.6.2.6 getX()

```
int Veiculo.getX ( ) [inline]
```

Returns

O valor de X do veículo

3.6.2.7 getY()

```
int Veiculo.getY ( ) [inline]
```

Returns

O valor de Y do veículo

3.6.2.8 setFabrica()

```
void Veiculo.setFabrica (
    boolean status ) [inline]
```

Define se um veículo está ou não dentro de uma fábrica, com base no que é recebido em status

Parameters

<i>status</i>	
---------------	--

3.6.2.9 setX()

```
int Veiculo.setX ( ) [inline]
```

Cria um valor aleatório para o X do veículo

Returns

O valor aleatório gerado para X

3.6.2.10 setY()

```
int Veiculo.setY ( ) [inline]
```

Cria um valor aleatório para o Y do veículo

Returns

O valor aleatório gerado para Y

The documentation for this class was generated from the following file:

- Veiculo.java

