

# ElevatorControlSystem

Generated by Doxygen 1.8.14



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	Elevator Class Reference . . . . .	5
3.1.1	Constructor & Destructor Documentation . . . . .	5
3.1.1.1	Elevator() . . . . .	6
3.1.2	Member Function Documentation . . . . .	6
3.1.2.1	GetCurrentFloor() . . . . .	6
3.1.2.2	GetStatus() . . . . .	6
3.1.2.3	IsDown() . . . . .	7
3.1.2.4	IsIdle() . . . . .	7
3.1.2.5	IsUp() . . . . .	7
3.1.2.6	SetCurrentGoal() . . . . .	7
3.1.2.7	SetPickupGoal() . . . . .	8
3.1.2.8	SetState() . . . . .	8
3.2	Simulation Class Reference . . . . .	8
3.2.1	Constructor & Destructor Documentation . . . . .	9
3.2.1.1	Simulation() . . . . .	9
3.2.2	Member Function Documentation . . . . .	9
3.2.2.1	Pickup() . . . . .	9
3.2.2.2	Status() . . . . .	10
3.2.2.3	Update() . . . . .	10

<b>4 File Documentation</b>	<b>11</b>
4.1 Elevator.cc File Reference . . . . .	11
4.1.1 Detailed Description . . . . .	11
4.2 Elevator.h File Reference . . . . .	11
4.2.1 Detailed Description . . . . .	12
4.3 Simulation.cc File Reference . . . . .	12
4.3.1 Detailed Description . . . . .	12
4.4 Simulation.h File Reference . . . . .	13
4.4.1 Detailed Description . . . . .	13
<b>Index</b>	<b>15</b>

# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">Elevator</a>	.....	5
<a href="#">Simulation</a>	.....	8



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all documented files with brief descriptions:

<a href="#">Elevator.cc</a>		
Implementation of <a href="#">Elevator</a> class . . . . .		11
<a href="#">Elevator.h</a>		
<a href="#">Elevator</a> class . . . . .		11
<a href="#">Simulation.cc</a>		
Implementation of <a href="#">Simulation</a> class . . . . .		12
<a href="#">Simulation.h</a>		
<a href="#">Simulation</a> class . . . . .		13





## Chapter 3

# Class Documentation

### 3.1 Elevator Class Reference

#### Public Types

- enum **State** { **kUp** = 1, **kDown** = -1, **kIdle** = 0 }

#### Public Member Functions

- [Elevator](#) (int elevator\_id, int story\_number)  
*Constructor for [Elevator](#).*
- std::tuple< int, int, std::set< int >, std::set< int, std::greater< int > > > [GetStatus](#) ()  
*Getting function which returns the status of an [Elevator](#).*
- void [SetPickupGoal](#) (int pickup\_floor, int goal\_floor)  
*Function which inserts pickup floor and goal floor into requests sets.*
- void [SetCurrentGoal](#) (int current\_floor, std::set< int > up\_requests, std::set< int, std::greater< int > > down\_requests)  
*Setting function which sets the current floor and requests for an [Elevator](#).*
- bool [IsIdle](#) ()  
*Boolean function which check if an [Elevator](#) is idle.*
- bool [IsUp](#) ()  
*Boolean function which check if an [Elevator](#) is going up.*
- bool [IsDown](#) ()  
*Boolean function which check if an [Elevator](#) is going down.*
- int [GetCurrentFloor](#) ()  
*Getting function which returns the current floor.*
- void [SetState](#) (State state)  
*Setting function which sets the state of an [Elevator](#).*
- void [Move](#) ()  
*Function which perform one step for all Elevators in simulation.*

#### 3.1.1 Constructor & Destructor Documentation

### 3.1.1.1 Elevator()

```
Elevator::Elevator (
    int elevator_id,
    int story_number )
```

Constructor for [Elevator](#).

#### Parameters

<i>elevator_id</i>	ID of an <a href="#">Elevator</a>
<i>story_number</i>	Number of stories in the building

## 3.1.2 Member Function Documentation

### 3.1.2.1 GetCurrentFloor()

```
int Elevator::GetCurrentFloor ( )
```

Getting function which returns the current floor.

#### Returns

int Current floor

### 3.1.2.2 GetStatus()

```
std::tuple< int, int, std::set< int >, std::set< int, std::greater< int > > > Elevator::↵
GetStatus ( )
```

Getting function which returns the status of an [Elevator](#).

#### Returns

std::tuple<int, int, std::set<int>, std::set<int, std::greater<int>>> Status of an [Elevator](#) in a tuple: <ID, current floor, up requests, down requests>

### 3.1.2.3 IsDown()

```
bool Elevator::IsDown ( )
```

Boolean function which check if an [Elevator](#) is going down.

#### Returns

true [Elevator](#) goes down  
false [Elevator](#) doesn't go down

### 3.1.2.4 IsIdle()

```
bool Elevator::IsIdle ( )
```

Boolean function which check if an [Elevator](#) is idle.

#### Returns

true [Elevator](#) is idle  
false [Elevator](#) is not idle

### 3.1.2.5 IsUp()

```
bool Elevator::IsUp ( )
```

Boolean function which check if an [Elevator](#) is going up.

#### Returns

true [Elevator](#) goes up  
false [Elevator](#) doesn't go up

### 3.1.2.6 SetCurrentGoal()

```
void Elevator::SetCurrentGoal (
    int current_floor,
    std::set< int > up_requests,
    std::set< int, std::greater< int >> down_requests )
```

Setting function which sets the current floor and requests for an [Elevator](#).

## Parameters

<i>current_floor</i>	Current floor
<i>up_requests</i>	Requests for going up
<i>down_requests</i>	Requests for going down

## 3.1.2.7 SetPickupGoal()

```
void Elevator::SetPickupGoal (
    int pickup_floor,
    int goal_floor )
```

Function which inserts pickup floor and goal floor into requests sets.

## Parameters

<i>pickup_floor</i>	pickup floor to be inserted
<i>goal_floor</i>	goal floor to be inserted

## 3.1.2.8 SetState()

```
void Elevator::SetState (
    State state )
```

Setting function which sets the state of an [Elevator](#).

## Parameters

<i>state</i>	State of an <a href="#">Elevator</a>
--------------	--------------------------------------

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

- [Elevator.h](#)
- [Elevator.cc](#)

## 3.2 Simulation Class Reference

## Public Member Functions

- [Simulation](#) (int elevator\_number, int story\_number)  
*Constructor for [Simulation](#).*
- `std::vector< std::tuple< int, int, std::set< int >, std::set< int, std::greater< int > > > > Status ()`

*Function which returns the status of Elevators in simulation.*

- void [Update](#) (int elevator\_id, int current\_floor, std::set< int > up\_requests, std::set< int, std::greater< int >> down\_requests)

*Function which updates information about an [Elevator](#).*

- void [Pickup](#) (int pickup\_floor, int goal\_floor)

*Function which picks up passengers in simulation.*

- void [Step](#) ()

*Function which performs one moving step for simulation system.*

### 3.2.1 Constructor & Destructor Documentation

#### 3.2.1.1 Simulation()

```
Simulation::Simulation (
    int elevator_number,
    int story_number )
```

Constructor for [Simulation](#).

##### Parameters

<i>elevator_number</i>	Number of Elevators in simulation
<i>story_number</i>	Number of stories in the building

### 3.2.2 Member Function Documentation

#### 3.2.2.1 Pickup()

```
void Simulation::Pickup (
    int pickup_floor,
    int goal_floor )
```

Function which picks up passengers in simulation.

##### Parameters

<i>pickup_floor</i>	Pickup floor
<i>goal_floor</i>	Goal floor

### 3.2.2.2 Status()

```
std::vector< std::tuple< int, int, std::set< int >, std::set< int, std::greater< int > > > >
> Simulation::Status ( )
```

Function which returns the status of Elevators in simulation.

#### Returns

`std::vector<std::tuple<int, int, std::set<int>, std::set<int, std::greater<int>>>>` Vector of status of Elevators

### 3.2.2.3 Update()

```
void Simulation::Update (
    int elevator_id,
    int current_floor,
    std::set< int > up_requests,
    std::set< int, std::greater< int >> down_requests )
```

Function which updates information about an [Elevator](#).

#### Parameters

<i>elevator_id</i>	ID
<i>current_floor</i>	Current floor
<i>up_requests</i>	Requests for going up
<i>down_requests</i>	Requests for going down

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

- [Simulation.h](#)
- [Simulation.cc](#)

## Chapter 4

# File Documentation

### 4.1 Elevator.cc File Reference

Implementation of [Elevator](#) class.

```
#include "Elevator.h"  
#include <tuple>  
#include <vector>  
#include <set>
```

#### 4.1.1 Detailed Description

Implementation of [Elevator](#) class.

**Author**

Zihan Qi

**Date**

2018-05-15

### 4.2 Elevator.h File Reference

[Elevator](#) class.

```
#include <iostream>  
#include <tuple>  
#include <vector>  
#include <set>
```

## Classes

- class [Elevator](#)

### 4.2.1 Detailed Description

[Elevator](#) class.

#### Author

Zihan Qi

#### Date

2018-05-15

## 4.3 Simulation.cc File Reference

Implementation of [Simulation](#) class.

```
#include "Simulation.h"
#include <iostream>
#include <set>
#include <vector>
#include <algorithm>
#include "Elevator.h"
```

## Variables

- const int **kMaxElevatorNumber** = 16

### 4.3.1 Detailed Description

Implementation of [Simulation](#) class.

#### Author

Zihan Qi

#### Date

2018-05-15



## 4.4 Simulation.h File Reference

[Simulation](#) class.

```
#include <vector>
#include "Elevator.h"
```

### Classes

- class [Simulation](#)

### 4.4.1 Detailed Description

[Simulation](#) class.

#### Author

Zihan Qi

#### Date

2018-05-15



# Index

- Elevator, [5](#)
  - Elevator, [5](#)
  - GetCurrentFloor, [6](#)
  - GetStatus, [6](#)
  - IsDown, [6](#)
  - IsIdle, [7](#)
  - IsUp, [7](#)
  - SetCurrentGoal, [7](#)
  - SetPickupGoal, [8](#)
  - SetState, [8](#)
- Elevator.cc, [11](#)
- Elevator.h, [11](#)
- GetCurrentFloor
  - Elevator, [6](#)
- GetStatus
  - Elevator, [6](#)
- IsDown
  - Elevator, [6](#)
- IsIdle
  - Elevator, [7](#)
- IsUp
  - Elevator, [7](#)
- Pickup
  - Simulation, [9](#)
- SetCurrentGoal
  - Elevator, [7](#)
- SetPickupGoal
  - Elevator, [8](#)
- SetState
  - Elevator, [8](#)
- Simulation, [8](#)
  - Pickup, [9](#)
  - Simulation, [9](#)
  - Status, [9](#)
  - Update, [10](#)
- Simulation.cc, [12](#)
- Simulation.h, [13](#)
- Status
  - Simulation, [9](#)
- Update
  - Simulation, [10](#)