Eng1Documentation

Generated by Doxygen 1.9.4

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

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

Chapter 4

Class Documentation

4.1 Flow Class Reference

Inheritance diagram for Flow:

4.2 FlowImplementation Class Reference

`#include <FlowImplementation.h>`

Inheritance diagram for FlowImplementation:

Collaboration diagram for FlowImplementation:

Public Member Functions

- ~FlowImplementation () override
- FlowImplementation (std::string name, double value, System *systemBegin, System *systemEnd)
- FlowImplementation (const FlowImplementation &flow)
- FlowImplementation & operator= (const Flow &flow)
- void expression () override=0
- std::string getName () const override
- void setName (std::string n) override
- double getValue () const override
- void setValue (double v) override
- System * getSystemBegin () const override
- void setSystemBegin (System *system) override
- System * getSystemEnd () const override
- void setSystemEnd (System *system) override

Protected Attributes

- std::string **name**
- double **value**
- System * **systemBegin**
- System * **systemEnd**

4.2.1 Detailed Description

Flow that converges energy from a model to another.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 ~FlowImplementation()

```
FlowImplementation::~FlowImplementation ( )  [override], [default]
```

Default destructor

4.2.2.2 FlowImplementation() [1/2]

```
FlowImplementation::FlowImplementation (
            std::string name,
            double value,
            System * systemBegin,
            System * systemEnd )
```

Default constructor

Parameters

name	Inital flow name
value	Inital flow value
systemBegin	Inital system where the flow comes from
systemEnd	Inital system where the flow goes to

Returns

Flow with initial name, value, systemBegin and systemEnd

4.2.2.3 FlowImplementation() [2/2]

```
FlowImplementation::FlowImplementation (
            const FlowImplementation & flow )
```

Copy constructor

Parameters

flow	Flow to copy from

Returns

Copied flow

4.2.3 Member Function Documentation

4.2.3.1 expression()

```
void FlowImplementation::expression ( )  [override], [pure virtual]
```

Sets the expression of the flow

Implements Flow.

4.2.3.2 getName()

```
std::string FlowImplementation::getName ( ) const  [override], [virtual]
```

Get system name

Implements Flow.

4.2.3.3 getSystemBegin()

```
System * FlowImplementation::getSystemBegin ( ) const  [override], [virtual]
```

Get systemBegin

Implements Flow.

4.2.3.4 getSystemEnd()

```
System * FlowImplementation::getSystemEnd ( ) const  [override], [virtual]
```

Get systemEnd

Implements Flow.

4.2.3.5 getValue()

```
double FlowImplementation::getValue ( ) const  [override], [virtual]
```

Get system value

Implements Flow.

4.2.3.6 operator=()

```
FlowImplementation & FlowImplementation::operator= (
            const Flow & flow )
```

Copy Assignment Operator

Parameters

flow	Flow to copy from

Returns

Copied flow

4.2.3.7 setName()

```
void FlowImplementation::setName (
            std::string n )  [override], [virtual]
```

Set system name

Parameters

n	Name for the flow

Implements Flow.

4.2.3.8 setSystemBegin()

```
void FlowImplementation::setSystemBegin (
            System * system )  [override], [virtual]
```

Set systemBegin

Parameters

system	SystemBegin for the flow

Implements Flow.

4.2.3.9 setSystemEnd()

```
void FlowImplementation::setSystemEnd (
            System * system )  [override], [virtual]
```

Set systemBegin

Parameters

system	SystemEnd for the flow

Implements Flow.

4.2.3.10 setValue()

```
void FlowImplementation::setValue (
            double v )  [override], [virtual]
```

Set system value

Parameters

v	Value for the flow

Implements Flow.

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

- /home/joaozenobio/CLionProjects/Eng1/src/lib/FlowImplementation.h
- /home/joaozenobio/CLionProjects/Eng1/src/lib/FlowImplementation.cpp

4.3 Model Class Reference

Inheritance diagram for Model:

Public Member Functions

- virtual void simulate (double, double, double)=0
- virtual std::string getName () const =0
- virtual void setName (std::string)=0
- virtual double getTime () const =0
- virtual void setTime (std::string)=0
- virtual void add (System ∗)=0
- virtual void add (Flow ∗)=0

4.3.1 Member Function Documentation

4.3.1.1 add() [1/2]

```
virtual void Model::add (
            Flow *  )  [pure virtual]
```

Implemented in ModelImplementation.

4.3.1.2 add() [2/2]

```
virtual void Model::add (
            System *  )  [pure virtual]
```

Implemented in ModelImplementation.

4.3.1.3 getName()

```
virtual std::string Model::getName ( ) const  [pure virtual]
```

Implemented in ModelImplementation.

4.3.1.4 getTime()

```
virtual double Model::getTime ( ) const  [pure virtual]
```

Implemented in ModelImplementation.

4.3.1.5 setName()

```
virtual void Model::setName (
            std::string  )  [pure virtual]
```

Implemented in ModelImplementation.

4.3.1.6 setTime()

```
virtual void Model::setTime (
            std::string  )  [pure virtual]
```

Implemented in ModelImplementation.

4.3.1.7 simulate()

```
virtual void Model::simulate (
            double ,
            double ,
            double  )  [pure virtual]
```

Implemented in ModelImplementation.

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

- /home/joaozenobio/CLionProjects/Eng1/src/lib/Model.h

4.4 ModelImplementation Class Reference

Inheritance diagram for ModelImplementation:

Collaboration diagram for ModelImplementation:

Public Member Functions

- ~ModelImplementation () override
- ModelImplementation (std::string name, double time)
- ModelImplementation (const ModelImplementation &model)
- ModelImplementation & operator= (const ModelImplementation &model)
- void simulate (double start, double end, double timestep) override
- std::string getName () const override
- void setName (std::string n) override
- double getTime () const override
- void setTime (std::string t) override
- void add (System *system) override
- void add (Flow *flow) override

Protected Attributes

- std::string **name**
- double **time**
- std::vector< System * > **systems**
- std::vector< Flow * > **flows**

4.4.1 Constructor & Destructor Documentation

4.4.1.1 ~ModelImplementation()

```
ModelImplementation::~ModelImplementation ( )  [override]
```

Default destructor

4.4.1.2 ModelImplementation() [1/2]

```
ModelImplementation::ModelImplementation (
            std::string name,
            double time )
```

Default constructor

Parameters

name	Inital model name
time	Inital model time

Returns

Model with initial name and time

4.4.1.3 ModelImplementation() [2/2]

```
ModelImplementation::ModelImplementation (
            const ModelImplementation & model )
```

Copy constructor

Parameters

model	Model to copy from

Returns

Copied model

4.4.2 Member Function Documentation

4.4.2.1 add() [1/2]

```
void ModelImplementation::add (
            Flow * flow )  [override], [virtual]
```

Add a flow to the model

Parameters

flow	Flow to be added to the model

Implements Model.

4.4.2.2 add() [2/2]

```
void ModelImplementation::add (
```

```
            System * system )  [override], [virtual]
```

Add a system to the model

Parameters

system	System to be added to the model

Implements Model.

4.4.2.3 getName()

```
std::string ModelImplementation::getName ( ) const  [override], [virtual]
```

Get model name

Implements Model.

4.4.2.4 getTime()

```
double ModelImplementation::getTime ( ) const  [override], [virtual]
```

Get model time

Implements Model.

4.4.2.5 operator=()

```
ModelImplementation & ModelImplementation::operator= (
            const ModelImplementation & model )
```

Copy Assignment Operator

Parameters

model	Model to copy from

Returns

Copied model

4.4.2.6 setName()

```
void ModelImplementation::setName (
            std::string n )  [override], [virtual]
```

Set model name

Parameters

n	Name for the system

Implements Model.

4.4.2.7 setTime()

```
void ModelImplementation::setTime (
            std::string t )  [override], [virtual]
```

Set model time

Parameters

t	Name for the system

Implements Model.

4.4.2.8 simulate()

```
void ModelImplementation::simulate (
            double start,
            double end,
            double timestep )  [override], [virtual]
```

Simulates the model during a period of time between start and end time values with a specified timestep

Parameters

start	Time where the simulation starts
end	Time where the simulation ends
timestep	Timestep value to simulate with

Implements Model.

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

- /home/joaozenobio/CLionProjects/Eng1/src/lib/ModelImplementation.h
- /home/joaozenobio/CLionProjects/Eng1/src/lib/ModelImplementation.cpp

4.5 System Class Reference

Inheritance diagram for System:

Public Member Functions

- virtual std::string getName () const =0
- virtual void setName (std::string)=0
- virtual double getValue () const =0
- virtual void setValue (double)=0

4.5.1 Member Function Documentation

4.5.1.1 getName()

```
virtual std::string System::getName ( ) const  [pure virtual]
```

Implemented in SystemImplementation.

4.5.1.2 getValue()

```
virtual double System::getValue ( ) const  [pure virtual]
```

Implemented in SystemImplementation.

4.5.1.3 setName()

```
virtual void System::setName (
            std::string  )  [pure virtual]
```

Implemented in SystemImplementation.

4.5.1.4 setValue()

```
virtual void System::setValue (
            double  )  [pure virtual]
```

Implemented in SystemImplementation.

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

- /home/joaozenobio/CLionProjects/Eng1/src/lib/System.h

4.6 SystemImplementation Class Reference

Inheritance diagram for SystemImplementation:

Collaboration diagram for SystemImplementation:

Public Member Functions

- ~SystemImplementation () override
- SystemImplementation (std::string name, double value)
- SystemImplementation (const SystemImplementation &system)
- SystemImplementation & operator= (const SystemImplementation &system)
- std::string getName () const override
- void setName (std::string n) override
- double getValue () const override
- void setValue (double v) override

Protected Attributes

- std::string **name**
- double **value**

4.6.1 Constructor & Destructor Documentation

4.6.1.1 ~SystemImplementation()

```
SystemImplementation::~SystemImplementation ( )  [override], [default]
```

Default destructor

4.6.1.2 SystemImplementation() [1/2]

```
SystemImplementation::SystemImplementation (
            std::string name,
            double value )
```

Default constructor

Parameters

name	Inital system name
value	Inital system value

Returns

System with initial name and value

4.6.1.3 SystemImplementation() [2/2]

```
SystemImplementation::SystemImplementation (
            const SystemImplementation & system )
```

Copy constructor

Parameters

system	System to copy from

Returns

Copied system

4.6.2 Member Function Documentation

4.6.2.1 getName()

```
std::string SystemImplementation::getName ( ) const  [override], [virtual]
```

Get system name

Implements System.

4.6.2.2 getValue()

```
double SystemImplementation::getValue ( ) const  [override], [virtual]
```

Get system value

Implements System.

4.6.2.3 operator=()

```
SystemImplementation & SystemImplementation::operator= (
            const SystemImplementation & system )
```

Copy Assignment Operator

Parameters

system	System to copy from

Returns

Copied system

4.6.2.4 setName()

```
void SystemImplementation::setName (
            std::string n )  [override], [virtual]
```

Set system name

Parameters

n	Name for the system

Implements System.

4.6.2.5 setValue()

```
void SystemImplementation::setValue (
            double v )  [override], [virtual]
```

Set system value

Parameters

v	Value for the system

Implements System.

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

- /home/joaozenobio/CLionProjects/Eng1/src/lib/SystemImplementation.h
- /home/joaozenobio/CLionProjects/Eng1/src/lib/SystemImplementation.cpp

Chapter 5

File Documentation

5.1 Flow.h

```
1 //
2 // Created by joaozenobio on 27/04/2022.
3 //
4
5 #ifndef ENG1_FLOW_H
6 #define ENG1_FLOW_H
7
8 #include <string>
9
10 #include "System.h"
11
12 class Flow {
13 public:
14     virtual ~Flow() = default;
15     virtual std::string getName() const = 0;
16     virtual void setName(std::string) = 0;
17     virtual double getValue() const = 0;
18     virtual void setValue(double) = 0;
19     virtual void expression() = 0;
20     virtual System* getSystemBegin() const = 0;
21     virtual void setSystemBegin(System*) = 0;
22     virtual System* getSystemEnd() const = 0;
23     virtual void setSystemEnd(System*) = 0;
24 };
25
26
27 #endif //ENG1_FLOW_H
```

5.2 FlowImplementation.h

```
1 //
2 // Created by joaozenobio on 28/04/2022.
3 //
4
5 #ifndef ENG1_FLOWIMPLEMENTATION_H
6 #define ENG1_FLOWIMPLEMENTATION_H
7
8
9 #include "Flow.h"
10
14 class FlowImplementation : public Flow {
15 protected:
16     std::string name;
17     double value;
18     System* systemBegin;
19     System* systemEnd;
20 public:
24     ~FlowImplementation() override;
25
34     FlowImplementation(std::string name, double value, System* systemBegin, System* systemEnd);
35
41     FlowImplementation(const FlowImplementation& flow);
42
48     FlowImplementation& operator=(const Flow& flow);
```

```
49
53      void expression() override = 0;
54
58      std::string getName() const override;
59
64      void setName(std::string n) override;
65
69      double getValue() const override;
70
75      void setValue(double v) override;
76
80      System* getSystemBegin() const override;
81
86      void setSystemBegin(System* system) override;
87
91      System* getSystemEnd() const override;
92
97      void setSystemEnd(System* system) override;
98 };
99
100
101 #endif //ENG1_FLOWIMPLEMENTATION_H
```

5.3 Model.h

```
1 //
2 // Created by joaozenobio on 27/04/2022.
3 //
4
5 #ifndef ENG1_MODEL_H
6 #define ENG1_MODEL_H
7
8
9 #include <vector>
10 #include <string>
11
12 #include "System.h"
13 #include "Flow.h"
14
15 class Model {
16 public:
17      virtual ~Model() = default;
18      virtual void simulate(double, double, double) = 0;
19      virtual std::string getName() const = 0;
20      virtual void setName(std::string) = 0;
21      virtual double getTime() const = 0;
22      virtual void setTime(std::string) = 0;
23      virtual void add(System*) = 0;
24      virtual void add(Flow*) = 0;
25 };
26
27
28 #endif //ENG1_MODEL_H
```

5.4 ModelImplementation.h

```
1 //
2 // Created by joaozenobio on 28/04/2022.
3 //
4
5 #ifndef ENG1_MODELIMPLEMENTATION_H
6 #define ENG1_MODELIMPLEMENTATION_H
7
8
9 #include "Model.h"
10
11 class ModelImplementation: public Model {
12 protected:
13      std::string name;
14      double time;
15      std::vector<System*> systems;
16      std::vector<Flow*> flows;
17 public:
21      ~ModelImplementation() override;
22
29      ModelImplementation(std::string name, double time);
30
36      ModelImplementation(const ModelImplementation& model);
37
```

```
43      ModelImplementation& operator=(const ModelImplementation& model);
44
51      void simulate(double start, double end, double timestep) override;
52
56      std::string getName() const override;
57
62      void setName(std::string n) override;
63
67      double getTime() const override;
68
73      void setTime(std::string t) override;
74
79      void add(System* system) override;
80
85      void add(Flow* flow) override;
86  };
87
88
89  #endif //ENG1_MODELIMPLEMENTATION_H
```

5.5 System.h

```
1  //
2  // Created by joaozenobio on 27/04/2022.
3  //
4
5  #ifndef ENG1_SYSTEM_H
6  #define ENG1_SYSTEM_H
7
8  #include <string>
9
10 class System {
11 public:
12      virtual ~System() = default;
13      virtual std::string getName() const = 0;
14      virtual void setName(std::string) = 0;
15      virtual double getValue() const = 0;
16      virtual void setValue(double) = 0;
17 };
18
19
20 #endif //ENG1_SYSTEM_H
```

5.6 SystemImplementation.h

```
1  //
2  // Created by joaozenobio on 28/04/2022.
3  //
4
5  #ifndef ENG1_SYSTEMIMPLEMENTATION_H
6  #define ENG1_SYSTEMIMPLEMENTATION_H
7
8  #include "System.h"
9
10 class SystemImplementation : public System {
11 protected:
12      std::string name;
13      double value;
14 public:
18      ~SystemImplementation() override;
19
26      SystemImplementation(std::string name, double value);
27
33      SystemImplementation(const SystemImplementation& system);
34
40      SystemImplementation& operator=(const SystemImplementation& system);
41
45      std::string getName() const override;
46
51      void setName(std::string n) override;
52
56      double getValue() const override;
57
62      void setValue(double v) override;
63 };
64
65
66 #endif //ENG1_SYSTEMIMPLEMENTATION_H
```

Index