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:

Bank . . . . . ??
Cashier . . . . . ??
Client . . . . . ??
CompareEventPriority . . . . . ??
DiscreteEventSimulation . . . . . ??
    Simulation . . . . . ??
Event . . . . . ??
    Arrival . . . . . ??
    Departure . . . . . ??
Poisson . . . . . ??
Queue . . . . . ??

Chapter 2

Class Index

2.1 Class List

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

Arrival . . . . . ??
Bank . . . . . ??
Cashier . . . . . ??
Client . . . . . ??
CompareEventPriority . . . . . ??
Departure . . . . . ??
DiscreteEventSimulation . . . . . ??
Event . . . . . ??
Poisson . . . . . ??
Queue . . . . . ??
Simulation . . . . . ??

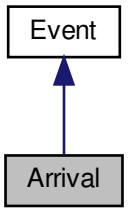
Chapter 3

Class Documentation

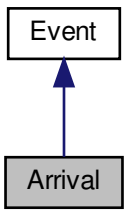
3.1 Arrival Class Reference

`#include <Arrival.h>`

Inheritance diagram for Arrival:



Collaboration diagram for Arrival:



Public Member Functions

- Arrival (double time, Simulation &simulation)
- Arrival (const Arrival &arrival)
- void process () override

3.1.1 Detailed Description

Arrival class, inheriting from Event class. Given private attributes are :

- cashier Cashier - the cashier related to the Arrival event
- client Client - the client who has just arrived in the bank
- simulation Simulation - an access to the Simulation object

3.1.2 Constructor & Destructor Documentation

3.1.2.1 Arrival() [1/2]

```
Arrival::Arrival (
            double time,
            Simulation & simulation )
```

Arrival Constructor

Parameters

time	Double - the arrival time of the client
simulation	Simulation - reference parameter to access simulation data from bank

3.1.2.2 Arrival() [2/2]

```
Arrival::Arrival (
            const Arrival & arrival )
```

Arrival Copy-Constructor

Parameters

arrival	Arrival

3.1.3 Member Function Documentation

3.1.3.1 process()

```
void Arrival::process ( )  [override], [virtual]
```

Launch Arrival's process. A new client is created. If a or several Cashier are available, the newly created client is attributed to the first available Cashier. Else, the client goes in the Queue. A new Arrival event is created during the process.

Implements Event.

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

- Arrival.h
- Arrival.cpp

3.2 Bank Class Reference

```
#include <Bank.h>
```

Public Member Functions

- Bank (double *averageServiceTimes, int cashiersCount, Simulation &simulation)
- Bank (const Bank &bank)
- ~Bank ()
- int getClientsCount ()
- void addClientToCount ()
- Cashier * getFirstAvailableCashier ()
- Queue & getQueue ()
- Simulation & getSimulation ()
- Cashier & getCashier (int index)

3.2.1 Detailed Description

Bank class. Given private attributes are :

- cashiersCount Int - the number of cashiers
- clientsCount Int - the total number of clients who came in the bank
- cashiers Cashier[] - an array of cashiers existing in the bank
- queue Queue - the clients' queue
- simulation Simulation - an access to the Simulation object

3.2.2 Constructor & Destructor Documentation

3.2.2.1 Bank() [1/2]

```
Bank::Bank (
            double * averageServiceTimes,
            int cashiersCount,
            Simulation & simulation )
```

Bank Constructor

Parameters

averageServiceTimes	Double - average Service time for each cashier during the Simulation
cashiersCount	Int - number of Cashier class objects during the simulation
simulation	Simulation - reference parameter to access simulation data from bank

3.2.2.2 Bank() [2/2]

```
Bank::Bank (
            const Bank & bank )
```

Bank Copy-Constructor

Parameters

bank	Bank

3.2.2.3 ~Bank()

```
Bank::~Bank ( )
```

Bank Destructor

3.2.3 Member Function Documentation

3.2.3.1 addClientToCount()

```
void Bank::addClientToCount ( )
```

Add a client to the clientsCount attribute by increment

3.2.3.2 getCashier()

```
Cashier & Bank::getCashier (
            int index )
```

Get the cashier at given index in cashiers attribute

Parameters

index	Int

Returns

Cashier

3.2.3.3 getClientsCount()

```
int Bank::getClientsCount ( )
```

Get clients count

Returns

clientsCount Int

3.2.3.4 getFirstAvailableCashier()

```
Cashier * Bank::getFirstAvailableCashier ( )
```

Get the first (from 0 to cashiersCount) available cashier

Returns

Cashier|nullptr

3.2.3.5 getQueue()

```
Queue & Bank::getQueue ( )
```

Get Queue attribute

Returns

Queue

3.2.3.6 getSimulation()

```
Simulation & Bank::getSimulation ( )
```

Get Simulation attribute

Returns

Simulation

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

- Bank.h
- Bank.cpp

3.3 Cashier Class Reference

```
#include <Cashier.h>
```

Public Member Functions

- Cashier (double averageServiceTime, Bank &bank)
- Cashier (const Cashier &cashier)
- int getClientsCount ()
- double getOccupationRate ()
- bool isAvailable ()
- void serveClient (Client &client)
- void free ()
- Bank & getBank ()
- double getOccupiedTime ()

3.3.1 Detailed Description

Cashier class. Given private attributes are :

- averageServiceTime Double - the average service time of the cashier
- clientsCount Int - the number of clients served by the cashier
- bank Bank - an access to the Bank object
- available Boolean - the cashier's availability
- occupiedTime Double - the total amount of occupied time of the cashier during the simulation

3.3.2 Constructor & Destructor Documentation

3.3.2.1 Cashier() [1/2]

```
Cashier::Cashier (
            double averageServiceTime,
            Bank & bank )
```

Cashier Constructor

Parameters

averageServiceTime	Double - the average service time of the cashier
bank	Bank - the bank in which the cashier is part of

3.3.2.2 Cashier() [2/2]

```
Cashier::Cashier (
            const Cashier & cashier )
```

Cashier Copy-Constructor

Parameters

cashier	Cashier

3.3.3 Member Function Documentation

3.3.3.1 free()

```
void Cashier::free ( )
```

Set available attribute to true

3.3.3.2 getBank()

```
Bank & Cashier::getBank ( )
```

Get bank attribute

Returns

Bank

3.3.3.3 getClientsCount()

```
int Cashier::getClientsCount ( )
```

Get clients count

Returns

clientsCount Int

3.3.3.4 getOccupationRate()

```
double Cashier::getOccupationRate ( )
```

Get occupation rate. The rate is calculated by using the formula : (occupiedTime∗100.00)/realDuration. Result is given in percentages

Returns

occupationRate Double

3.3.3.5 getOccupiedTime()

```
double Cashier::getOccupiedTime ( )
```

Get occupied time attribute

Returns

occupiedTime Double

3.3.3.6 isAvailable()

```
bool Cashier::isAvailable ( )
```

Get available attribute

Returns

Boolean - according to cashier's availability

3.3.3.7 serveClient()

```
void Cashier::serveClient (
            Client & client )
```

Serve the given client

Parameters

client	Client

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

- Cashier.h
- Cashier.cpp

3.4 Client Class Reference

```
#include <Client.h>
```

Public Member Functions

- Client (double arrivalTime)
- Client (const Client &client)
- double getArrivalTime ()

3.4.1 Detailed Description

Client class. Given private attribute is :

- arrivalTime Double - the client's time of arrival

3.4.2 Constructor & Destructor Documentation

3.4.2.1 Client() [1/2]

```
Client::Client (
            double arrivalTime )  [explicit]
```

Client Constructor

Parameters

arrivalTime	Double - the client's time of arrival in the bank

3.4.2.2 Client() [2/2]

```
Client::Client (
            const Client & client )
```

Client Copy-Constructor

Parameters

client	Client

3.4.3 Member Function Documentation

3.4.3.1 getArrivalTime()

```
double Client::getArrivalTime ( )
```

Get arrival time

Returns

arrivalTime Double

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

- Client.h
- Client.cpp

3.5 CompareEventPriority Class Reference

Public Member Functions

- int operator() (Event *&e1, Event *&e2)

3.5.1 Member Function Documentation

3.5.1.1 operator()()

```
int CompareEventPriority::operator() (
            Event *& e1,
            Event *& e2 )
```

Compare events priority

Parameters

e1	Event
e2	Event

Returns

Int - 0 or 1

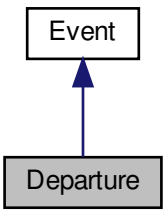
The documentation for this class was generated from the following files:

- CompareEventPriority.h
- CompareEventPriority.cpp

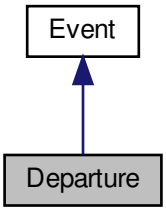
3.6 Departure Class Reference

```
#include <Departure.h>
```

Inheritance diagram for Departure:



Collaboration diagram for Departure:



Public Member Functions

- Departure (double time, Client &client, Cashier &cashier)
- Departure (const Departure &departure)
- void process () override

3.6.1 Detailed Description

Departure class, inheriting from Event class. Given private attributes are :

- cashier Cashier - the cashier related to the Departure event
- client Client - the client who has just arrived in the bank

3.6.2 Constructor & Destructor Documentation

3.6.2.1 Departure() [1/2]

```
Departure::Departure (
            double time,
            Client & client,
            Cashier & cashier )
```

Departure Constructor

Parameters

time	Double - the arrival time of the client
client	Client - the client who is leaving the bank
cashier	Cashier - the cashier who served the client

3.6.2.2 Departure() [2/2]

```
Departure::Departure (
            const Departure & departure )
```

Departure Copy-Constructor

Parameters

departure	Departure

3.6.3 Member Function Documentation

3.6.3.1 process()

```
void Departure::process ( )  [override], [virtual]
```

Launch Departure's process. The cashier associated with the departure is going to be available, while the client will be destroyed.

Implements Event.

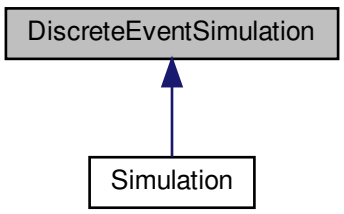
The documentation for this class was generated from the following files:

- Departure.h
- Departure.cpp

3.7 DiscreteEventSimulation Class Reference

```
#include <DiscreteEventSimulation.h>
```

Inheritance diagram for DiscreteEventSimulation:



Public Member Functions

- DiscreteEventSimulation (double startTime)
- DiscreteEventSimulation (const DiscreteEventSimulation &discreteEventSimulation)
- void addEvent (Event *event)
- double getCurrentTime ()
- void launch ()

3.7.1 Detailed Description

DiscreteEventSimulation class. Given private attributes are :

- currentTime Double - the current time of the simulation
- startTime Double - the start time of the simulation
- eventQueue priority_queue<Event*, vector<Event*>, CompareEventPriority> - Queue of all simulation's events to be processed

3.7.2 Constructor & Destructor Documentation

3.7.2.1 DiscreteEventSimulation() [1/2]

```
DiscreteEventSimulation::DiscreteEventSimulation (
            double startTime )  [explicit]
```

DiscreteEventSimulation Constructor

Parameters

startTime	Double - start time of the simulation

3.7.2.2 DiscreteEventSimulation() [2/2]

```
DiscreteEventSimulation::DiscreteEventSimulation (
            const DiscreteEventSimulation & discreteEventSimulation )
```

DiscreteEventSimulation Copy-Constructor

Parameters

discreteEventSimulation	DiscreteEventSimulation

3.7.3 Member Function Documentation

3.7.3.1 addEvent()

```
void DiscreteEventSimulation::addEvent (
            Event * event )
```

Add the given event to eventQueue

Parameters

event	Event

3.7.3.2 getCurrentTime()

```
double DiscreteEventSimulation::getCurrentTime ( )
```

Get current time

Returns

currentTime Double

3.7.3.3 launch()

```
void DiscreteEventSimulation::launch ( )
```

Launch simulation

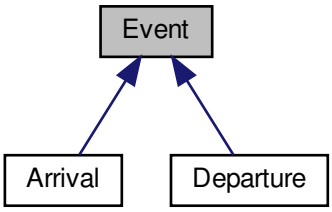
The documentation for this class was generated from the following files:

- DiscreteEventSimulation.h
- DiscreteEventSimulation.cpp

3.8 Event Class Reference

`#include <Event.h>`

Inheritance diagram for Event:



Public Member Functions

- Event (double time)
- Event (const Event &event)
- virtual void process ()=0
- double getTime ()

3.8.1 Detailed Description

Event class, working as a virtual class. Given attribute is :

- time : double attribute indicating the time when the event took place

3.8.2 Constructor & Destructor Documentation

3.8.2.1 Event() [1/2]

```
Event::Event (
            double time )  [explicit]
```

Event Constructor

Parameters

time	Double - the arrival or departure time of the client

3.8.2.2 Event() [2/2]

```
Event::Event (
            const Event & event )
```

Event Copy-Constructor

Parameters

event	Event

3.8.3 Member Function Documentation

3.8.3.1 getTime()

```
double Event::getTime ( )
```

Get time

Returns

time Double

3.8.3.2 process()

```
virtual void Event::process ( )  [pure virtual]
```

Launch Event's process

Implemented in Arrival, and Departure.

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

- Event.h
- Event.cpp

3.9 Poisson Class Reference

Static Public Member Functions

- static void init (int seed=0)
- static double next (double moy=1.0)

3.9.1 Member Function Documentation

3.9.1.1 init()

```
void Poisson::init (
            int seed = 0 )  [static]
```

Initialise Poisson's Law

Parameters

seed	Int

3.9.1.2 next()

```
double Poisson::next (
            double moy = 1.0 )  [static]
```

Get a random following Poisson's Law

Parameters

moy	Double

Returns

random Double

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

- Poisson.h
- Poisson.cpp

3.10 Queue Class Reference

```
#include <Queue.h>
```

Public Member Functions

- Queue (Bank &bank)
- Queue (const Queue &queue)
- int getMaxLength ()
- double getAverageLength ()
- double getAverageWaitingTime ()
- void addClient (Client &client)
- bool isEmpty ()
- Client * remove ()

3.10.1 Detailed Description

Queue class. Given private attributes are :

- clientQueue deque<Client *> - basically a FIFO (First In First out) queue
- maxLength Int - the maximum length of the queue reached during the simulation
- integral Double - the result of the calculated integral for the average length
- bank Bank - an access to the Bank object
- waitingTime Double - the average waiting time in the queue

3.10.2 Constructor & Destructor Documentation

3.10.2.1 Queue() [1/2]

```
Queue::Queue (
            Bank & bank )  [explicit]
```

Queue Constructor

Parameters

bank	Bank - reference parameter to access simulation data from bank

3.10.2.2 Queue() [2/2]

```
Queue::Queue (
            const Queue & queue )
```

Queue Copy-Constructor

Parameters

queue	Queue

3.10.3 Member Function Documentation

3.10.3.1 addClient()

```
void Queue::addClient (
            Client & client )
```

Add client to clientQueue

Parameters

client	Client

3.10.3.2 getAverageLength()

```
double Queue::getAverageLength ( )
```

Get average length of the queue

Returns

averageLength Double

3.10.3.3 getAverageWaitingTime()

```
double Queue::getAverageWaitingTime ( )
```

Get average waiting time in the queue

Returns

averageWaitingTime Double

3.10.3.4 getMaxLength()

```
int Queue::getMaxLength ( )
```

Get max length of the queue reached during the simulation

Returns

maxLength Int

3.10.3.5 isEmpty()

```
bool Queue::isEmpty ( )
```

Get clientQueue emptiness

Returns

Boolean - according to queue emptiness

3.10.3.6 remove()

```
Client * Queue::remove ( )
```

Remove first client from clientQueue and return it

Returns

Client

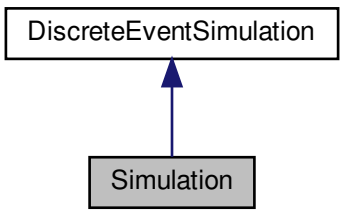
The documentation for this class was generated from the following files:

- Queue.h
- Queue.cpp

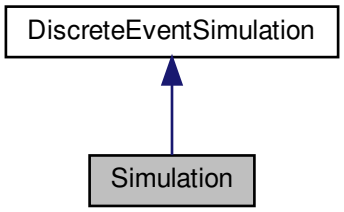
3.11 Simulation Class Reference

```
#include <Simulation.h>
```

Inheritance diagram for Simulation:



Collaboration diagram for Simulation:



Public Member Functions

- Simulation (double plannedDuration, double averageArrivalTime, double ∗averageServiceTimes, int cashiersCount)
- Simulation (const Simulation &simulation)
- double getPlannedDuration ()
- double getAverageArrivalTime ()
- Bank & getBank ()
- void setRealDuration (double realDuration)
- double getRealDuration ()

Public Attributes

- bool **DEBUG** = false

3.11.1 Detailed Description

Simulation class. Given private attributes are :

- plannedDuration Double - planned duration for the simulation
- realDuration Double - real duration at the end of the simulation
- averageArrivalTime Double - average arrival time wanted for the simulation
- bank bank - an access to the Bank object

3.11.2 Constructor & Destructor Documentation

3.11.2.1 Simulation() [1/2]

```
Simulation::Simulation (
            double plannedDuration,
            double averageArrivalTime,
            double * averageServiceTimes,
            int cashiersCount )
```

Simulation Constructor

Parameters

plannedDuration	Double - planned duration for the simulation
averageArrivalTime	Double - average arrival time wanted for the simulation
averageServiceTimes	Double[] - average service times wanted for the cashiers
cashiersCount	Int - wanted number of cashiers for the bank

3.11.2.2 Simulation() [2/2]

```
Simulation::Simulation (
            const Simulation & simulation )
```

Simulation Copy-Constructor

Parameters

simulation	Simulation

3.11.3 Member Function Documentation

3.11.3.1 getAverageArrivalTime()

```
double Simulation::getAverageArrivalTime ( )
```

Get average arrival time

Returns

averageArrivalTime Double

3.11.3.2 getBank()

```
Bank & Simulation::getBank ( )
```

Get bank

Returns

Bank

3.11.3.3 getPlannedDuration()

```
double Simulation::getPlannedDuration ( )
```

Get simulation's planned duration

Returns

plannedDuration Double

3.11.3.4 getRealDuration()

```
double Simulation::getRealDuration ( )
```

Get simulation's real duration

Returns

realDuration Double

3.11.3.5 setRealDuration()

```
void Simulation::setRealDuration (
            double realDuration )
```

Set simulation's real duration

Parameters

realDuration	Double

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

- Simulation.h
- Simulation.cpp