

# Project Software Engineering

2.1

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>projectSoftwareEngineering</b>	<b>1</b>
<b>2</b>	<b>Hierarchical Index</b>	<b>3</b>
2.1	Class Hierarchy . . . . .	3
<b>3</b>	<b>Class Index</b>	<b>5</b>
3.1	Class List . . . . .	5
<b>4</b>	<b>Class Documentation</b>	<b>7</b>
4.1	Albatros Class Reference . . . . .	7
4.1.1	Constructor & Destructor Documentation . . . . .	7
4.1.1.1	Albatros() [1/2] . . . . .	7
4.1.1.2	Albatros() [2/2] . . . . .	8
4.1.2	Member Function Documentation . . . . .	8
4.1.2.1	stoptInStation() . . . . .	8
4.2	ClassTestMetroNet Class Reference . . . . .	8
4.3	ClassTestPassagier Class Reference . . . . .	9
4.4	ClassTestStation Class Reference . . . . .	9
4.5	ClassTestTram Class Reference . . . . .	10
4.6	Halte Class Reference . . . . .	10
4.6.1	Constructor & Destructor Documentation . . . . .	11
4.6.1.1	Halte() [1/2] . . . . .	11
4.6.1.2	Halte() [2/2] . . . . .	11
4.6.2	Member Function Documentation . . . . .	11
4.6.2.1	albatrosCanStop() . . . . .	11

4.7	MetroNet Class Reference	12
4.7.1	Constructor & Destructor Documentation	12
4.7.1.1	MetroNet()	12
4.7.2	Member Function Documentation	12
4.7.2.1	addPassagier()	12
4.7.2.2	addStation()	13
4.7.2.3	addTram()	13
4.7.2.4	drawToOutputStream()	13
4.7.2.5	getPassagier()	13
4.7.2.6	getStation()	13
4.7.2.7	getTram()	14
4.7.2.8	isConsistent()	14
4.7.2.9	moveAllePassengers()	14
4.7.2.10	moveAlleTrams()	14
4.7.2.11	runSimulation()	14
4.7.2.12	writeToOutputStream()	14
4.8	MetroNetImporter Class Reference	15
4.8.1	Member Function Documentation	15
4.8.1.1	importMetroNet()	15
4.8.1.2	importPassengers()	15
4.9	MetroNetInputTests Class Reference	15
4.10	MetroNetOutputTest Class Reference	16
4.11	MetroStation Class Reference	17
4.11.1	Constructor & Destructor Documentation	17
4.11.1.1	MetroStation() [1/2]	17
4.11.1.2	MetroStation() [2/2]	17
4.11.2	Member Function Documentation	17
4.11.2.1	albatrosCanStop()	18
4.12	Passagier Class Reference	18
4.12.1	Constructor & Destructor Documentation	18

4.12.1.1	Passagier() [1/2]	18
4.12.1.2	Passagier() [2/2]	19
4.12.2	Member Function Documentation	19
4.12.2.1	getBeginStation()	19
4.12.2.2	getEindStation()	19
4.12.2.3	getHoeveelheid()	19
4.12.2.4	getNaam()	19
4.12.2.5	isAangekomen()	20
4.12.2.6	markAangekomen()	20
4.12.2.7	moveToBeginStation()	20
4.12.2.8	setBeginStation()	20
4.12.2.9	setEindStation()	20
4.12.2.10	setHoeveelheid()	21
4.12.2.11	setNaam()	21
4.13	PassagierInputTests Class Reference	21
4.14	PCC Class Reference	22
4.14.1	Constructor & Destructor Documentation	22
4.14.1.1	PCC() [1/2]	22
4.14.1.2	PCC() [2/2]	22
4.14.2	Member Function Documentation	23
4.14.2.1	stopInStation()	23
4.15	Station Class Reference	23
4.15.1	Constructor & Destructor Documentation	24
4.15.1.1	Station() [1/2]	24
4.15.1.2	Station() [2/2]	24
4.15.2	Member Function Documentation	24
4.15.2.1	addPassagier()	25
4.15.2.2	addVolgende()	25
4.15.2.3	addVorige()	25
4.15.2.4	albatrosCanStop()	25

4.15.2.5	getNaam()	25
4.15.2.6	getSporen()	26
4.15.2.7	getTramInStation()	26
4.15.2.8	getVolgende()	26
4.15.2.9	getVorige()	26
4.15.2.10	isInStation()	26
4.15.2.11	isTramInStation()	26
4.15.2.12	movePassagiers()	27
4.15.2.13	removePassagier()	27
4.15.2.14	setNaam()	27
4.15.2.15	setTramInStation()	27
4.16	Tram Class Reference	28
4.16.1	Constructor & Destructor Documentation	29
4.16.1.1	Tram() [1/2]	29
4.16.1.2	Tram() [2/2]	29
4.16.2	Member Function Documentation	29
4.16.2.1	addPassagier()	29
4.16.2.2	afstappenInHalte()	30
4.16.2.3	getAantalPassagiers()	30
4.16.2.4	getBeginStation()	30
4.16.2.5	getCurrentStation()	30
4.16.2.6	getLijnNr()	30
4.16.2.7	getOmzet()	30
4.16.2.8	getSnelheid()	31
4.16.2.9	getVoertuigNr()	31
4.16.2.10	getZitplaatsen()	31
4.16.2.11	isInTram()	31
4.16.2.12	moveTram()	31
4.16.2.13	removePassagier()	31
4.16.2.14	setAantalPassagiers()	32
4.16.2.15	setBeginStation()	32
4.16.2.16	setCurrentStation()	32
4.16.2.17	setLijnNr()	32
4.16.2.18	setOmzet()	32
4.16.2.19	setSnelheid()	33
4.16.2.20	setVoertuigNr()	33
4.16.2.21	setZitplaatsen()	33
4.16.2.22	stoptInStation()	33

## **Chapter 1**

# **projectSoftwareEngineering**

Project Software Engineering / Bachelor 1 Informatica / Andrei Bondarenko & Igor Schittekat





## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

MetroNet . . . . .	12
MetroNetImporter . . . . .	15
Passagier . . . . .	18
Station . . . . .	23
Halte . . . . .	10
MetroStation . . . . .	17
Test	
ClassTestMetroNet . . . . .	8
ClassTestPassagier . . . . .	9
ClassTestStation . . . . .	9
ClassTestTram . . . . .	10
MetroNetInputTests . . . . .	15
MetroNetOutputTest . . . . .	16
PassagierInputTests . . . . .	21
Tram . . . . .	28
Albatros . . . . .	7
PCC . . . . .	22



## Chapter 3

# Class Index

### 3.1 Class List

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

Albatros	7
ClassTestMetroNet	8
ClassTestPassagier	9
ClassTestStation	9
ClassTestTram	10
Halte	10
MetroNet	12
MetroNetImporter	15
MetroNetInputTests	15
MetroNetOutputTest	16
MetroStation	17
Passagier	18
PassagierInputTests	21
PCC	22
Station	23
Tram	28

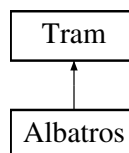


## Chapter 4

# Class Documentation

### 4.1 Albatros Class Reference

Inheritance diagram for Albatros:



#### Public Member Functions

- [Albatros](#) ()
- [Albatros](#) (const int lijnNr, const int voertuigNr, const int zitplaatsen, const std::string &beginStation, const int snelheid)
- bool [stopInStation](#) ([MetroNet](#) &metronet, std::string station) const

#### Additional Inherited Members

#### 4.1.1 Constructor & Destructor Documentation

##### 4.1.1.1 Albatros() [1/2]

```
Albatros::Albatros ( )
```

```
ENSURE(properlyInitialized(), "constructor must end in properlyInitialized state");
```

#### 4.1.1.2 Albatros() [2/2]

```
Albatros::Albatros (
    const int lijnNr,
    const int voertuigNr,
    const int zitplaatsen,
    const std::string & beginStation,
    const int snelheid )
```

```

    REQUIRE(beginStation != "", "newBeginStation must not be empty");
    REQUIRE(lijnNr >= 0 , "lijnNr must be bigger or equal to zero");
    REQUIRE(voertuigNr >= 0 , "voertuigNr must be bigger or equal to zero");
    REQUIRE(zitplaatsen >= 0 , "zitplaatsen must be bigger or equal to zero");
    REQUIRE(snelheid >= 0 , "snelheid must be bigger or equal to zero");
    ENSURE(properlyInitialized(), "constructor must end in properlyInitialized state");
```

### 4.1.2 Member Function Documentation

#### 4.1.2.1 stoptInStation()

```
bool Albatros::stoptInStation (
    MetroNet & metronet,
    std::string station ) const [virtual]
```

```

    REQUIRE(properlyInitialized(), "Albatros wasn't initialized when calling stoptInStation");
    REQUIRE(metronet.properlyInitialized(), "MetroNet wasn't initialized when calling stoptInStation");
    REQUIRE(station != "", "station must not be empty");
```

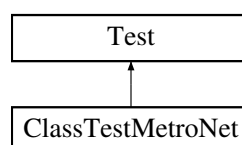
Reimplemented from [Tram](#).

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

- /Users/Andrei/Uni/SoftwareEngineering/projectSoftwareEngineering/Albatros.h
- /Users/Andrei/Uni/SoftwareEngineering/projectSoftwareEngineering/Albatros.cpp

## 4.2 ClassTestMetroNet Class Reference

Inheritance diagram for ClassTestMetroNet:



### Protected Attributes

- [MetroNet](#) **metronet**

### Friends

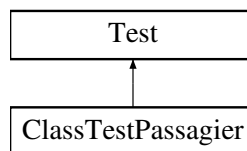
- class **MetroNet**

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

- /Users/Andrei/Uni/SoftwareEngineering/projectSoftwareEngineering/ClassTestsMetroNet.cpp

## 4.3 ClassTestPassagier Class Reference

Inheritance diagram for ClassTestPassagier:



### Protected Attributes

- [Passagier](#) **passagier**

### Friends

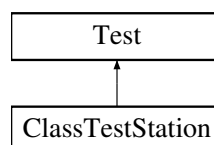
- class **MetroNet**

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

- /Users/Andrei/Uni/SoftwareEngineering/projectSoftwareEngineering/ClassTestsPassagier.cpp

## 4.4 ClassTestStation Class Reference

Inheritance diagram for ClassTestStation:



### Protected Attributes

- [Station](#) **station**
- [Halte](#) **halte**
- [MetroStation](#) **metroStation**
- [MetroNet](#) **metronet**

### Friends

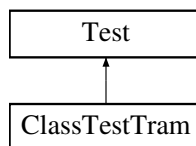
- class **MetroNet**

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

- /Users/Andrei/Uni/SoftwareEngineering/projectSoftwareEngineering/ClassTestsStation.cpp

## 4.5 ClassTestTram Class Reference

Inheritance diagram for ClassTestTram:



### Protected Attributes

- [Tram](#) **tram**
- [PCC](#) **pcc**
- [Albatros](#) **albatros**
- [MetroNet](#) **net**

### Friends

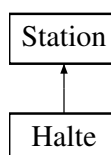
- class **MetroNet**

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

- /Users/Andrei/Uni/SoftwareEngineering/projectSoftwareEngineering/ClassTestsTram.cpp

## 4.6 Halte Class Reference

Inheritance diagram for Halte:





## Public Member Functions

- [Halte](#) ()
- [Halte](#) (const std::string &naam)
- bool [albatrosCanStop](#) () const

## Additional Inherited Members

### 4.6.1 Constructor & Destructor Documentation

#### 4.6.1.1 [Halte\(\)](#) [1/2]

```
Halte::Halte ( )
```

```
ENSURE(properlyInitialized(), "constructor must end in properlyInitialized state");
```

#### 4.6.1.2 [Halte\(\)](#) [2/2]

```
Halte::Halte (
    const std::string & naam )
```

```
REQUIRE(naam != "", "naam must not be empty");
ENSURE(properlyInitialized(), "constructor must end in properlyInitialized state");
```

### 4.6.2 Member Function Documentation

#### 4.6.2.1 [albatrosCanStop\(\)](#)

```
bool Halte::albatrosCanStop ( ) const [virtual]
```

```
REQUIRE(properlyInitialized(), "Halte wasn't initialized when calling albatrosCanStop");
```

Reimplemented from [Station](#).

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

- /Users/Andrei/Uni/SoftwareEngineering/projectSoftwareEngineering/Halte.h
- /Users/Andrei/Uni/SoftwareEngineering/projectSoftwareEngineering/Halte.cpp

## 4.7 MetroNet Class Reference

### Public Member Functions

- [MetroNet](#) ()
- bool **properlyInitialized** () const
- bool [isConsistent](#) ()
- [Station](#) \* [getStation](#) (std::string naam)
- [Tram](#) \* [getTram](#) (int voertuigNr)
- [Passagier](#) \* [getPassagier](#) (std::string naam)
- void [addStation](#) ([Station](#) \*newStation)
- void [addTram](#) ([Tram](#) \*newTram)
- void [addPassagier](#) ([Passagier](#) \*newPassagier)
- void [moveAlleTrams](#) (std::ostream &output)
- void [moveAllePassengers](#) (std::ostream &output)
- void [runSimulation](#) (std::ostream &output, const bool live=false)
- void [writeToOutputStream](#) (std::ostream &output)
- void [drawToOutputStream](#) (std::ostream &output)
- void **writeToASCII** ()

### 4.7.1 Constructor & Destructor Documentation

#### 4.7.1.1 MetroNet()

```
MetroNet::MetroNet ( )
```

```
ENSURE(properlyInitialized(), "constructor must end in properlyInitialized state");
```

### 4.7.2 Member Function Documentation

#### 4.7.2.1 addPassagier()

```
void MetroNet::addPassagier (
    Passagier * newPassagier )
```

```

REQUIRE(properlyInitialized(), "MetroNet wasn't initialized when calling addPassagier");
REQUIRE(getPassagier(newPassagier->getNaam()) == nullptr, "This MetroNet already contains a passenger with
this name");
ENSURE(getPassagier(newPassagier->getNaam) == newPassagier, "addPassagier post condition failure");

```

#### 4.7.2.2 addStation()

```
void MetroNet::addStation (
    Station * newStation )
```

```

    REQUIRE(properlyInitialized(), "MetroNet wasn't initialized when calling addStation");
    REQUIRE(getStation(newStation->getNaam()) == nullptr, "This MetroNet already contains a station with this name");
    ENSURE(getStation(newStation->getNaam()) == newStation, "addStation post condition failure");
```

#### 4.7.2.3 addTram()

```
void MetroNet::addTram (
    Tram * newTram )
```

```

    REQUIRE(properlyInitialized(), "MetroNet wasn't initialized when calling addTram");
    REQUIRE(getTram(newTram->voertuigNr()) == nullptr, "This MetroNet already contains a Tram with this voertuigNr");
    REQUIRE(getStation(newTram->getBeginStation())->isTramInStation(newTram->getLijnNr()) == false, "BeginStation of newTram isn't empty");
    ENSURE(getTram(newTram->voertuigNr()) == newTram, "addTram post condition failure");
    ENSURE(getStation(newTram->getBeginStation())->isTramInStation(), "addTram post condition failure");
```

#### 4.7.2.4 drawToOutputStream()

```
void MetroNet::drawToOutputStream (
    std::ostream & output )
```

```

    REQUIRE(isConsistent(), "MetroNet is not consistent");
```

#### 4.7.2.5 getPassagier()

```
Passagier * MetroNet::getPassagier (
    std::string naam )
```

```

    REQUIRE(properlyInitialized(), "MetroNet wasn't initialized when calling getPassagier");
    REQUIRE(naam != "", "naam must not be empty");
```

#### 4.7.2.6 getStation()

```
Station * MetroNet::getStation (
    std::string naam )
```

```

    REQUIRE(properlyInitialized(), "MetroNet wasn't initialized when calling getAlleStations");
    REQUIRE(naam != "", "naam must not be empty");
```

#### 4.7.2.7 getTram()

```
Tram * MetroNet::getTram (
    int voertuigNr )
```

REQUIRE(`properlyInitialized()`, "MetroNet wasn't initialized when calling `getAlleTrams`");  
REQUIRE(`voertuigNr >= 0`, "voertuigNr must be greater or equal to zero");

#### 4.7.2.8 isConsistent()

```
bool MetroNet::isConsistent ( )
```

REQUIRE(`properlyInitialized()`, "MetroNet wasn't initialized when calling `isConsistent`");

#### 4.7.2.9 moveAllePassengers()

```
void MetroNet::moveAllePassengers (
    std::ostream & output )
```

REQUIRE(`properlyInitialized()`, "MetroNet wasn't initialized when calling `moveAllePassengers`");

#### 4.7.2.10 moveAlleTrams()

```
void MetroNet::moveAlleTrams (
    std::ostream & output )
```

REQUIRE(`properlyInitialized()`, "MetroNet wasn't initialized when calling `moveAlleTrams`");  
ENSURE(`isConsistent()`, "moveAlleTrams made MetroNet inconsistent");

#### 4.7.2.11 runSimulation()

```
void MetroNet::runSimulation (
    std::ostream & output,
    const bool live = false )
```

REQUIRE(`properlyInitialized()`, "MetroNet wasn't initialized when calling `runSimulation`");  
ENSURE(`isConsistent()`, "runSimulation made MetroNet inconsistent");

#### 4.7.2.12 writeToOutputStream()

```
void MetroNet::writeToOutputStream (
    std::ostream & output )
```

REQUIRE(`isConsistent()`, "MetroNet is not consistent");

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

- /Users/Andrei/Uni/SoftwareEngineering/projectSoftwareEngineering/MetroNet.h
- /Users/Andrei/Uni/SoftwareEngineering/projectSoftwareEngineering/MetroNet.cpp

## 4.8 MetroNetImporter Class Reference

### Static Public Member Functions

- static SuccessEnum [importMetroNet](#) (const char \*inputfilename, std::ostream &errStream, [MetroNet](#) &metronet)
- static SuccessEnum [importPassengers](#) (const char \*inputfilename, std::ostream &errStream, [MetroNet](#) &metronet)

### 4.8.1 Member Function Documentation

#### 4.8.1.1 importMetroNet()

```
SuccessEnum MetroNetImporter::importMetroNet (
    const char * inputfilename,
    std::ostream & errStream,
    MetroNet & metronet ) [static]
```

REQUIRE(metronet.properlyInitialized(), "metronet wasn't initialized when passed to MetroNetImporter::importMetroNet");  
 ENSURE(metronet.isConsistent(), "MetroNet is not consistent");

#### 4.8.1.2 importPassengers()

```
SuccessEnum MetroNetImporter::importPassengers (
    const char * inputfilename,
    std::ostream & errStream,
    MetroNet & metronet ) [static]
```

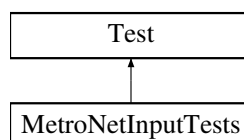
REQUIRE(metronet.properlyInitialized(), "metronet wasn't initialized when passed to MetroNetImporter::importPassengers");  
 ENSURE(metronet.isConsistent(), "MetroNet is not consistent");

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

- /Users/Andrei/Uni/SoftwareEngineering/projectSoftwareEngineering/MetroNetImporter.h
- /Users/Andrei/Uni/SoftwareEngineering/projectSoftwareEngineering/MetroNetImporter.cpp

## 4.9 MetroNetInputTests Class Reference

Inheritance diagram for MetroNetInputTests:



### Protected Member Functions

- virtual void **SetUp** ()

### Protected Attributes

- [MetroNet](#) \* **metronet** = new [MetroNet](#)()

### Friends

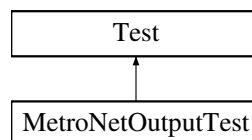
- class **MetroNet**

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

- /Users/Andrei/Uni/SoftwareEngineering/projectSoftwareEngineering/MetroNetInputTests.cpp

## 4.10 MetroNetOutputTest Class Reference

Inheritance diagram for MetroNetOutputTest:



### Protected Member Functions

- virtual void **SetUp** ()

### Protected Attributes

- [MetroNet](#) \* **metronet** = new [MetroNet](#)()

### Friends

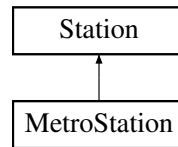
- class **MetroNet**

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

- /Users/Andrei/Uni/SoftwareEngineering/projectSoftwareEngineering/MetroNetOutputTests.cpp

## 4.11 MetroStation Class Reference

Inheritance diagram for MetroStation:



### Public Member Functions

- [MetroStation\(\)](#)
- [MetroStation\(const std::string &naam\)](#)
- [bool albatrosCanStop\(\) const](#)

### Additional Inherited Members

#### 4.11.1 Constructor & Destructor Documentation

##### 4.11.1.1 MetroStation() [1/2]

```
MetroStation::MetroStation ( )
```

```
ENSURE(properlyInitialized(), "constructor must end in properlyInitialized state");
```

##### 4.11.1.2 MetroStation() [2/2]

```
MetroStation::MetroStation (
    const std::string & naam )
```

```
REQUIRE(naam != "", "naam must not be empty");
ENSURE(properlyInitialized(), "constructor must end in properlyInitialized state");
```

#### 4.11.2 Member Function Documentation

#### 4.11.2.1 albatrosCanStop()

```
bool MetroStation::albatrosCanStop ( ) const [virtual]
```

REQUIRE(properlyInitialized(), "MetroStation wasn't initialized when calling albatrosCanStop");

Reimplemented from [Station](#).

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

- /Users/Andrei/Uni/SoftwareEngineering/projectSoftwareEngineering/MetroStation.h
- /Users/Andrei/Uni/SoftwareEngineering/projectSoftwareEngineering/MetroStation.cpp

## 4.12 Passagier Class Reference

### Public Member Functions

- [Passagier](#) ()
- [Passagier](#) (std::string naam, std::string beginStation, std::string eindStation, int hoeveelheid)
- bool **properlyInitialized** () const
- std::string [getNaam](#) () const
- std::string [getBeginStation](#) () const
- std::string [getEindStation](#) () const
- int [getHoeveelheid](#) () const
- bool [isAangekomen](#) () const
- void [setNaam](#) (const std::string &newNaam)
- void [setBeginStation](#) (const std::string &newBeginStation)
- void [setEindStation](#) (const std::string &newEindStation)
- void [setHoeveelheid](#) (const int newHoeveelheid)
- void [moveToBeginStation](#) ([MetroNet](#) &metronet) const
- void [markAangekomen](#) ()

### 4.12.1 Constructor & Destructor Documentation

#### 4.12.1.1 Passagier() [1/2]

```
Passagier::Passagier ( )
```

ENSURE(properlyInitialized(), "constructor must end in properlyInitialized state");



#### 4.12.1.2 Passagier() [2/2]

```
Passagier::Passagier (
    std::string naam,
    std::string beginStation,
    std::string eindStation,
    int hoeveelheid )
```

```
 REQUIRE(naam != "", "naam must not be empty");
 REQUIRE(beginStation != "", "beginStation must not be empty");
 REQUIRE(eindStation != "", "eindStation must not be empty");
 REQUIRE(hoeveelheid >= 0 , "hoeveelheid must be bigger or equal to zero");
 ENSURE(properlyInitialized(), "constructor must end in properlyInitialized state");
```

### 4.12.2 Member Function Documentation

#### 4.12.2.1 getBeginStation()

```
std::string Passagier::getBeginStation ( ) const
```

```
 REQUIRE(properlyInitialized(), "Passagier wasn't initialized when calling getBeginStation");
```

#### 4.12.2.2 getEindStation()

```
std::string Passagier::getEindStation ( ) const
```

```
 REQUIRE(properlyInitialized(), "Passagier wasn't initialized when calling getEindStation");
```

#### 4.12.2.3 getHoeveelheid()

```
int Passagier::getHoeveelheid ( ) const
```

```
 REQUIRE(properlyInitialized(), "Passagier wasn't initialized when calling getHoeveelheid");
```

#### 4.12.2.4 getNaam()

```
std::string Passagier::getNaam ( ) const
```

```
 REQUIRE(properlyInitialized(), "Passagier wasn't initialized when calling getNaam");
```

#### 4.12.2.5 isAangekomen()

```
bool Passagier::isAangekomen ( ) const
```

```
    REQUIRE(properlyInitialized(), "Passagier wasn't initialized when calling isAangekomen");
```

#### 4.12.2.6 markAangekomen()

```
void Passagier::markAangekomen ( )
```

```
    REQUIRE(properlyInitialized(), "Passagier wasn't initialized when calling markAangekomen");
    ENSURE(isAangekomen() == true, "markAangekomen post condition failure");
```

#### 4.12.2.7 moveToBeginStation()

```
void Passagier::moveToBeginStation (
    MetroNet & metronet ) const
```

```
    REQUIRE(properlyInitialized(), "Passagier wasn't initialized when calling moveToBeginStation");
    REQUIRE(metronet->properlyInitialized(), "MetroNet wasn't initialized when calling Passagier::moveToBeginStation");
    ENSURE(station->isInStation(naam), "moveToBeginStation post condition failure");
    ENSURE(metronet->isConsistent(), "Passagier::moveToBeginStation made MetroNet inconsistent");
```

#### 4.12.2.8 setBeginStation()

```
void Passagier::setBeginStation (
    const std::string & newBeginStation )
```

```
    REQUIRE(properlyInitialized(), "Passagier wasn't initialized when calling setBeginStation");
    REQUIRE(newBeginStation != "", "newBeginStation must not be empty");
    ENSURE(getBeginStation() == newBeginStation, "setBeginStation post condition failure");
```

#### 4.12.2.9 setEindStation()

```
void Passagier::setEindStation (
    const std::string & newEindStation )
```

```
    REQUIRE(properlyInitialized(), "Passagier wasn't initialized when calling setEindStation");
    REQUIRE(newEindStation != "", "newEindStation must not be empty");
    ENSURE(getEindStation() == newEindStation, "setEindStation post condition failure");
```

## 4.12.2.10 setHoeveelheid()

```
void Passagier::setHoeveelheid (
    const int newHoeveelheid )
```

```

    REQUIRE(properlyInitialized(), "Passagier wasn't initialized when calling setHoeveelheid");
    REQUIRE(newHoeveelheid >= 0 , "newHoeveelheid must be bigger or equal to zero");
    ENSURE(getHoeveelheid() == newHoeveelheid, "setHoeveelheid post condition failure");
```

## 4.12.2.11 setNaam()

```
void Passagier::setNaam (
    const std::string & newNaam )
```

```

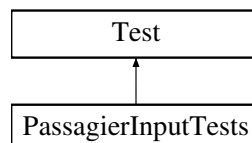
    REQUIRE(properlyInitialized(), "Passagier wasn't initialized when calling setNaam");
    REQUIRE(newNaam != "" , "newNaam must not be empty");
    ENSURE(getNaam() == newNaam, "setNaam post condition failure");
```

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

- /Users/Andrei/Uni/SoftwareEngineering/projectSoftwareEngineering/Passagier.h
- /Users/Andrei/Uni/SoftwareEngineering/projectSoftwareEngineering/Passagier.cpp

## 4.13 PassagierInputTests Class Reference

Inheritance diagram for PassagierInputTests:



## Protected Member Functions

- virtual void **SetUp** ()

## Protected Attributes

- **MetroNet** \* **metronet** = new **MetroNet**()

## Friends

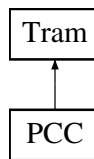
- class **MetroNet**

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

- /Users/Andrei/Uni/SoftwareEngineering/projectSoftwareEngineering/PassagierInputTests.cpp

## 4.14 PCC Class Reference

Inheritance diagram for PCC:



### Public Member Functions

- [PCC](#) ()
- [PCC](#) (const int lijnNr, const int voertuigNr, const int zitplaatsen, const std::string &beginStation, const int snelheid)
- bool [stoptInStation](#) ([MetroNet](#) &metronet, std::string station) const

### Additional Inherited Members

#### 4.14.1 Constructor & Destructor Documentation

##### 4.14.1.1 PCC() [1/2]

```
PCC::PCC ( )
```

```
ENSURE(properlyInitialized(), "constructor must end in properlyInitialized state");
```

##### 4.14.1.2 PCC() [2/2]

```
PCC::PCC (
    const int lijnNr,
    const int voertuigNr,
    const int zitplaatsen,
    const std::string & beginStation,
    const int snelheid )
```

```

REQUIRE(beginStation != "", "newBeginStation must not be empty");
REQUIRE(lijnNr >= 0 , "lijnNr must be bigger or equal to zero");
REQUIRE(voertuigNr >= 0 , "voertuigNr must be bigger or equal to zero");
REQUIRE(zitplaatsen >= 0 , "zitplaatsen must be bigger or equal to zero");
REQUIRE(snelheid >= 0 , "snelheid must be bigger or equal to zero");
ENSURE(properlyInitialized(), "constructor must end in properlyInitialized state");

```

## 4.14.2 Member Function Documentation

### 4.14.2.1 stopInStation()

```
bool PCC::stopInStation (
    MetroNet & metronet,
    std::string station ) const [virtual]
```

REQUIRE(properlyInitialized(), "PCC wasn't initialized when calling afstappenInHalte");  
 REQUIRE(metronet.properlyInitialized(), "MetroNet wasn't initialized when calling afstappenInHalte");  
 REQUIRE(station != "", "station must not be empty");

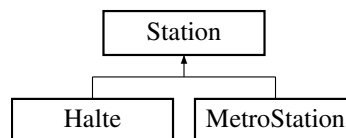
Reimplemented from [Tram](#).

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

- /Users/Andrei/Uni/SoftwareEngineering/projectSoftwareEngineering/PCC.h
- /Users/Andrei/Uni/SoftwareEngineering/projectSoftwareEngineering/PCC.cpp

## 4.15 Station Class Reference

Inheritance diagram for Station:



### Public Member Functions

- bool **properlyInitialized** () const
- std::string **getNaam** () const
- std::string **getVorige** (const int &spoor) const
- std::string **getVolgende** (const int &spoor) const
- std::set< int > **getSporen** () const
- bool **isTramInStation** (const int &spoor) const
- int **getTramInStation** (const int spoor) const
- void **setNaam** (const std::string &newNaam)
- void **addVorige** (const int &spoor, const std::string &newVorige)
- void **addVolgende** (const int &spoor, const std::string &newVolgende)
- bool **setTramInStation** (const int &spoor, const int voertuigNr, const bool &newTramInStation)
- bool **isInStation** (std::string passagier)
- void **addPassagier** (std::string passagier)
- void **removePassagier** (std::string passagier)
- void **movePassagiers** ([MetroNet](#) &metronet, std::ostream &output)
- virtual bool **albatrosCanStop** () const

## Protected Member Functions

- [Station](#) ()
- [Station](#) (const std::string &naam)

## Protected Attributes

- [Station](#) \* **initCheck**
- std::string **naam**
- std::map< int, std::string > **vorige**
- std::map< int, std::string > **volgende**
- std::map< std::pair< int, int >, bool > **tramInStation**
- std::set< std::string > **passagiers**

## Friends

- class **ClassTestStation**

## 4.15.1 Constructor & Destructor Documentation

### 4.15.1.1 [Station\(\)](#) [1/2]

```
Station::Station ( ) [protected]
```

```
ENSURE(properlyInitialized(), "constructor must end in properlyInitialized state");
```

### 4.15.1.2 [Station\(\)](#) [2/2]

```
Station::Station (
    const std::string & naam ) [protected]
```

```
REQUIRE(naam != "", "naam must not be empty");
ENSURE(properlyInitialized(), "constructor must end in properlyInitialized state");
```

## 4.15.2 Member Function Documentation

#### 4.15.2.1 addPassagier()

```
void Station::addPassagier (
    std::string passagier )
```

```

    REQUIRE(properlyInitialized(), "Station wasn't initialized when calling addPassagier");
    REQUIRE(passagier != "", "passagier must not be empty");
    REQUIRE(isInStation(passagier) == false, "passenger already in isInStation");
    ENSURE(isInStation(passagier) == true, "addPassagier post condition failure");
```

#### 4.15.2.2 addVolgende()

```
void Station::addVolgende (
    const int & spoor,
    const std::string & newVolgende )
```

```

    REQUIRE(properlyInitialized(), "Station wasn't initialized when calling setVolgende");
    REQUIRE(newVolgende != "", "newVolgende must not be empty");
    REQUIRE(spoor >= 0, "spoor must be bigger or equal to zero");
    ENSURE(getVolgende() == newVolgende, "setVolgende post condition failure");
```

#### 4.15.2.3 addVorige()

```
void Station::addVorige (
    const int & spoor,
    const std::string & newVorige )
```

```

    REQUIRE(properlyInitialized(), "Station wasn't initialized when calling setVorige");
    REQUIRE(newVorige != "", "newVorige must not be empty");
    REQUIRE(spoor >= 0, "spoor must be bigger or equal to zero");
    ENSURE(getVorige() == newVorige, "setVorige post condition failure");
```

#### 4.15.2.4 albatrosCanStop()

```
bool Station::albatrosCanStop ( ) const [virtual]
```

```

    REQUIRE(properlyInitialized(), "Station wasn't initialized when calling albatrosCanStop");
```

Reimplemented in [Halte](#), and [MetroStation](#).

#### 4.15.2.5 getNaam()

```
std::string Station::getNaam ( ) const
```

```

    REQUIRE(properlyInitialized(), "Station wasn't initialized when calling getNaam");
```

#### 4.15.2.6 getSporen()

```
std::set< int > Station::getSporen ( ) const
```

REQUIRE(properlyInitialized(), "Station wasn't initialized when calling getSporen");

#### 4.15.2.7 getTramInStation()

```
int Station::getTramInStation (
    const int spoor ) const
```

REQUIRE(properlyInitialized(), "Station wasn't initialized when calling getTramInStation");

REQUIRE(spoor >= 0, "spoor must be bigger or equal to zero");

REQUIRE(isTramInStation(spoor), "no tram in station when calling getTramInStation");

#### 4.15.2.8 getVolgende()

```
std::string Station::getVolgende (
    const int & spoor ) const
```

REQUIRE(properlyInitialized(), "Station wasn't initialized when calling getVolgende");

REQUIRE(spoor >= 0, "parameter spoor must be >= 0, when passed to getVolgende");

#### 4.15.2.9 getVorige()

```
std::string Station::getVorige (
    const int & spoor ) const
```

REQUIRE(properlyInitialized(), "Station wasn't initialized when calling getVorige");

REQUIRE(spoor >= 0, "parameter spoor must be >= 0, when passed to getVorige");

#### 4.15.2.10 isInStation()

```
bool Station::isInStation (
    std::string passagier )
```

REQUIRE(properlyInitialized(), "Station wasn't initialized when calling isInStation");

REQUIRE(passagier != "", "passagier must not be empty");

#### 4.15.2.11 isTramInStation()

```
bool Station::isTramInStation (
    const int & spoor ) const
```

REQUIRE(properlyInitialized(), "Station wasn't initialized when calling isTramInStation");

REQUIRE(spoor >= 0, "parameter spoor must be >= 0, when passed to isTramInStation");



## 4.15.2.12 movePassagiers()

```
void Station::movePassagiers (
    MetroNet & metronet,
    std::ostream & output )
```

REQUIRE(properlyInitialized(), "Station wasn't initialized when calling movePassagiers");  
 REQUIRE(metronet.properlyInitialized(), "MetroNet wasn't initialized when calling movePassagiers");  
 ENSURE(metronet.isConsistent(), "movePassagiers made MetroNet inconsistent");

## 4.15.2.13 removePassagier()

```
void Station::removePassagier (
    std::string passagier )
```

REQUIRE(properlyInitialized(), "Station wasn't initialized when calling removePassagier");  
 REQUIRE(passagier != "", "passagier must not be empty");  
 REQUIRE(isInStation(passagier) == true, "passenger not in isInStation");  
 ENSURE(isInStation(passagier) == false, "removePassagier post condition failure");

## 4.15.2.14 setNaam()

```
void Station::setNaam (
    const std::string & newNaam )
```

REQUIRE(properlyInitialized(), "Station wasn't initialized when calling setNaam");  
 REQUIRE(newNaam != "", "newNaam must not be empty");  
 ENSURE(getNaam() == newNaam, "setNaam post condition failure");

## 4.15.2.15 setTramInStation()

```
bool Station::setTramInStation (
    const int & spoor,
    const int voertuigNr,
    const bool & newTramInStation )
```

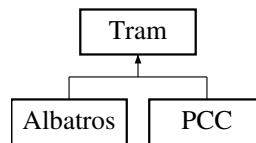
REQUIRE(properlyInitialized(), "Station wasn't initialized when calling setTramInStation");  
 REQUIRE(spoor >= 0, "spoor must be bigger or equal to zero");  
 REQUIRE(voertuigNr >= 0, "voertuigNr must be bigger or equal to zero");  
 ENSURE(isTramInStation(spoor) != newTramInStation, "setTramInStation post condition failure"); // false  
 ENSURE(isTramInStation(spoor) == newTramInStation, "setTramInStation post condition failure"); // true

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

- /Users/Andrei/Uni/SoftwareEngineering/projectSoftwareEngineering/Station.h
- /Users/Andrei/Uni/SoftwareEngineering/projectSoftwareEngineering/Station.cpp

## 4.16 Tram Class Reference

Inheritance diagram for Tram:



### Public Member Functions

- bool **properlyInitialized** () const
- int [getLijnNr](#) () const
- int [getZitplaatsen](#) () const
- std::string [getBeginStation](#) () const
- std::string [getCurrentStation](#) () const
- int [getSnelheid](#) () const
- int [getAantalPassagiers](#) () const
- int [getVoertuigNr](#) () const
- int [getOmzet](#) () const
- bool [isInTram](#) (std::string passagier) const
- void [setLijnNr](#) (const int newLijnNr)
- void [setZitplaatsen](#) (const int newZitplaatsen)
- void [setBeginStation](#) (const std::string &newBeginStation)
- void [setCurrentStation](#) (const std::string &newCurrentStation)
- void [setSnelheid](#) (const int newSnelheid)
- void [setAantalPassagiers](#) (const int newAantalPassagiers)
- void [setVoertuigNr](#) (const int newVoertuigNr)
- void [setOmzet](#) (const int newOmzet)
- void [addPassagier](#) (std::string passagier, int aantal)
- void [removePassagier](#) (std::string passagier)
- std::set< std::string > [afstappenInHalte](#) ([MetroNet](#) &metronet, std::string station)
- void [moveTram](#) ([MetroNet](#) &metronet, std::ostream &output)
- virtual bool [stopInStation](#) ([MetroNet](#) &metronet, std::string station) const

### Protected Member Functions

- [Tram](#) ()
- [Tram](#) (const int lijnNr, const int voertuigNr, const int zitplaatsen, const std::string &beginStation, const int snelheid)

### Protected Attributes

- [Tram](#) \* **initCheck**
- int **lijnNr**
- int **voertuigNr**
- int **zitplaatsen**
- int **snelheid**
- std::string **beginStation**
- std::string **currentStation**
- std::set< std::string > **passagiers**
- int **aantalPassagiers** = 0
- int **omzet** = 0
- const int **ticketPrijs** = 2

## Friends

- class **ClassTestTram**

### 4.16.1 Constructor & Destructor Documentation

#### 4.16.1.1 Tram() [1/2]

```
Tram::Tram ( ) [protected]
```

ENSURE(properlyInitialized(), "constructor must end in properlyInitialized state");

#### 4.16.1.2 Tram() [2/2]

```
Tram::Tram (
    const int lijnNr,
    const int voertuigNr,
    const int zitplaatsen,
    const std::string & beginStation,
    const int snelheid ) [protected]
```

```

    REQUIRE(beginStation != "", "newBeginStation must not be empty");
    REQUIRE(lijnNr >= 0 , "lijnNr must be bigger or equal to zero");
    REQUIRE(newVoertuigNr >= 0 , "newVoertuigNr must be bigger or equal to zero");
    REQUIRE(zitplaatsen >= 0 , "zitplaatsen must be bigger or equal to zero");
    REQUIRE(snelheid >= 0 , "snelheid must be bigger or equal to zero");
    ENSURE(properlyInitialized(), "constructor must end in properlyInitialized state");
```

### 4.16.2 Member Function Documentation

#### 4.16.2.1 addPassagier()

```
void Tram::addPassagier (
    std::string passagier,
    int aantal )
```

```

    REQUIRE(properlyInitialized(), "Tram wasn't initialized when calling addPassagier");
    REQUIRE(passagier != "", "passagier must not be empty");
    REQUIRE(aantal >= 0, "aantal must be bigger or equal to zero");
    REQUIRE(isInTram(passagier) == false, "passenger already in Tram");
    ENSURE(isInTram(passagier) == true, "addPassagier post condition failure");
```

#### 4.16.2.2 afstappenInHalte()

```
std::set< std::string > Tram::afstappenInHalte (
    MetroNet & metronet,
    std::string station )
```

```
    REQUIRE(properlyInitialized(), "Tram wasn't initialized when calling afstappenInHalte");
    REQUIRE(metronet.properlyInitialized(), "MetroNet wasn't initialized when calling afstappenInHalte");
    REQUIRE(station != "", "station must not be empty");
```

#### 4.16.2.3 getAantalPassagiers()

```
int Tram::getAantalPassagiers ( ) const
```

```
    REQUIRE(properlyInitialized(), "Tram wasn't initialized when calling getAantalPassagiers");
```

#### 4.16.2.4 getBeginStation()

```
std::string Tram::getBeginStation ( ) const
```

```
    REQUIRE(properlyInitialized(), "Tram wasn't initialized when calling getBeginStation");
```

#### 4.16.2.5 getCurrentStation()

```
std::string Tram::getCurrentStation ( ) const
```

```
    REQUIRE(properlyInitialized(), "Tram wasn't initialized when calling getCurrentStation");
```

#### 4.16.2.6 getLijnNr()

```
int Tram::getLijnNr ( ) const
```

```
    REQUIRE(properlyInitialized(), "Tram wasn't initialized when calling getLijnNr");
```

#### 4.16.2.7 getOmzet()

```
int Tram::getOmzet ( ) const
```

```
    REQUIRE(properlyInitialized(), "Tram wasn't initialized when calling getOmzet");
```

#### 4.16.2.8 getSnelheid()

```
int Tram::getSnelheid ( ) const
```

REQUIRE(properlyInitialized(), "Tram wasn't initialized when calling getSnelheid");

#### 4.16.2.9 getVoertuigNr()

```
int Tram::getVoertuigNr ( ) const
```

REQUIRE(properlyInitialized(), "Tram wasn't initialized when calling getVoertuigNr");

#### 4.16.2.10 getZitplaatsen()

```
int Tram::getZitplaatsen ( ) const
```

REQUIRE(properlyInitialized(), "Tram wasn't initialized when calling getZitplaatsen");

#### 4.16.2.11 isInTram()

```
bool Tram::isInTram (
    std::string passagier ) const
```

REQUIRE(properlyInitialized(), "Tram wasn't initialized when calling isInTram");  
 REQUIRE(passagier != "", "passagier must not be empty");

#### 4.16.2.12 moveTram()

```
void Tram::moveTram (
    MetroNet & metronet,
    std::ostream & output )
```

REQUIRE(properlyInitialized(), "Tram wasn't initialized when calling moveTram");  
 REQUIRE(metronet.properlyInitialized(), "MetroNet wasn't initialized when calling moveTram");  
 ENSURE(moved || attemptedToMove, "moveTram post condition failure");  
 ENSURE(metronet.isConsistent(), "moveTram made MetroNet inconsistent");

#### 4.16.2.13 removePassagier()

```
void Tram::removePassagier (
    std::string passagier )
```

REQUIRE(properlyInitialized(), "Tram wasn't initialized when calling removePassagier");  
 REQUIRE(passagier != "", "passagier must not be empty");  
 REQUIRE(isInTram(passagier) == true, "passenger not in Tram");  
 ENSURE(isInTram(passagier) == false, "removePassagier post condition failure");

#### 4.16.2.14 setAantalPassagiers()

```
void Tram::setAantalPassagiers (
    const int newAantalPassagiers )
```

REQUIRE(properlyInitialized(), "Tram wasn't initialized when calling setAantalPassagiers");  
REQUIRE(newAantalPassagiers >= 0 , "newAantalPassagiers must be bigger or equal to zero");  
ENSURE([getAantalPassagiers\(\)](#) == newAantalPassagiers, "setAantalPassagiers post condition failure");

#### 4.16.2.15 setBeginStation()

```
void Tram::setBeginStation (
    const std::string & newBeginStation )
```

REQUIRE(properlyInitialized(), "Tram wasn't initialized when calling setBeginStation");  
REQUIRE(newBeginStation != "", "newBeginStation must not be empty");  
ENSURE([getBeginStation\(\)](#) == newBeginStation, "setBeginStation post condition failure");

#### 4.16.2.16 setCurrentStation()

```
void Tram::setCurrentStation (
    const std::string & newCurrentStation )
```

REQUIRE(properlyInitialized(), "Tram wasn't initialized when calling setCurrentStation");  
REQUIRE(newCurrentStation != "", "newCurrentStation must not be empty");  
ENSURE([getCurrentStation\(\)](#) == newCurrentStation, "setCurrentStation post condition failure");

#### 4.16.2.17 setLijnNr()

```
void Tram::setLijnNr (
    const int newLijnNr )
```

REQUIRE(properlyInitialized(), "Tram wasn't initialized when calling setLijnNr");  
REQUIRE(newLijnNr >= 0 , "newLijnNr must be bigger or equal to zero");  
ENSURE([getLijnNr\(\)](#) == newLijnNr, "setLijnNr post condition failure");

#### 4.16.2.18 setOmzet()

```
void Tram::setOmzet (
    const int newOmzet )
```

REQUIRE(properlyInitialized(), "Tram wasn't initialized when calling setOmzet");  
REQUIRE(newOmzet >= 0 , "newOmzet must be bigger or equal to zero");  
ENSURE([getOmzet\(\)](#) == newOmzet, "setOmzet post condition failure");

## 4.16.2.19 setSnelheid()

```
void Tram::setSnelheid (
    const int newSnelheid )
```

```
REQUIRE(properlyInitialized(), "Tram wasn't initialized when calling setSnelheid");
REQUIRE(newSnelheid >= 0 , "newSnelheid must be bigger or equal to zero");
ENSURE(getSnelheid() == newSnelheid, "setSnelheid post condition failure");
```

## 4.16.2.20 setVoertuigNr()

```
void Tram::setVoertuigNr (
    const int newVoertuigNr )
```

```
REQUIRE(properlyInitialized(), "Tram wasn't initialized when calling setVoertuigNr");
REQUIRE(newVoertuigNr >= 0 , "newVoertuigNr must be bigger or equal to zero");
ENSURE(getVoertuigNr() == newVoertuigNr, "setVoertuigNr post condition failure");
```

## 4.16.2.21 setZitplaatsen()

```
void Tram::setZitplaatsen (
    const int newZitplaatsen )
```

```
REQUIRE(properlyInitialized(), "Tram wasn't initialized when calling setZitplaatsen");
REQUIRE(newZitplaatsen >= 0 , "newZitplaatsen must be bigger or equal to zero");
ENSURE(getZitplaatsen() == newZitplaatsen, "setZitplaatsen post condition failure");
```

## 4.16.2.22 stoptInStation()

```
bool Tram::stoptInStation (
    MetroNet & metronet,
    std::string station ) const [virtual]
```

```
REQUIRE(properlyInitialized(), "Tram wasn't initialized when calling afstappenInHalte");
REQUIRE(metronet.properlyInitialized(), "MetroNet wasn't initialized when calling afstappenInHalte");
REQUIRE(station != "", "station must not be empty");
```

Reimplemented in [Albatros](#), and [PCC](#).

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

- /Users/Andrei/Uni/SoftwareEngineering/projectSoftwareEngineering/Tram.h
- /Users/Andrei/Uni/SoftwareEngineering/projectSoftwareEngineering/Tram.cpp





# Index

- addPassagier
  - MetroNet, [12](#)
  - Station, [24](#)
  - Tram, [29](#)
- addStation
  - MetroNet, [12](#)
- addTram
  - MetroNet, [13](#)
- addVolgende
  - Station, [25](#)
- addVorige
  - Station, [25](#)
- afstappenInHalte
  - Tram, [29](#)
- Albatros, [7](#)
  - Albatros, [7](#)
  - stoptInStation, [8](#)
- albatrosCanStop
  - Halte, [11](#)
  - MetroStation, [17](#)
  - Station, [25](#)
- ClassTestMetroNet, [8](#)
- ClassTestPassagier, [9](#)
- ClassTestStation, [9](#)
- ClassTestTram, [10](#)
- drawToOutputStream
  - MetroNet, [13](#)
- getAantalPassagiers
  - Tram, [30](#)
- getBeginStation
  - Passagier, [19](#)
  - Tram, [30](#)
- getCurrentStation
  - Tram, [30](#)
- getEindStation
  - Passagier, [19](#)
- getHoeveelheid
  - Passagier, [19](#)
- getLijnNr
  - Tram, [30](#)
- getNaam
  - Passagier, [19](#)
  - Station, [25](#)
- getOmzet
  - Tram, [30](#)
- getPassagier
  - MetroNet, [13](#)
- getSnelheid
  - Tram, [30](#)
- getSporen
  - Station, [25](#)
- getStation
  - MetroNet, [13](#)
- getTram
  - MetroNet, [13](#)
- getTramInStation
  - Station, [26](#)
- getVoertuigNr
  - Tram, [31](#)
- getVolgende
  - Station, [26](#)
- getVorige
  - Station, [26](#)
- getZitplaatsen
  - Tram, [31](#)
- Halte, [10](#)
  - albatrosCanStop, [11](#)
  - Halte, [11](#)
- importMetroNet
  - MetroNetImporter, [15](#)
- importPassengers
  - MetroNetImporter, [15](#)
- isAangekomen
  - Passagier, [19](#)
- isConsistent
  - MetroNet, [14](#)
- isInStation
  - Station, [26](#)
- isInTram
  - Tram, [31](#)
- isTramInStation
  - Station, [26](#)
- markAangekomen
  - Passagier, [20](#)
- MetroNet, [12](#)
  - addPassagier, [12](#)
  - addStation, [12](#)
  - addTram, [13](#)
  - drawToOutputStream, [13](#)
  - getPassagier, [13](#)
  - getStation, [13](#)
  - getTram, [13](#)
  - isConsistent, [14](#)
  - MetroNet, [12](#)

- moveAllePassengers, [14](#)
  - moveAlleTrams, [14](#)
  - runSimulation, [14](#)
  - writeToOutputStream, [14](#)
- MetroNetImporter, [15](#)
  - importMetroNet, [15](#)
  - importPassengers, [15](#)
- MetroNetInputTests, [15](#)
- MetroNetOutputTest, [16](#)
- MetroStation, [17](#)
  - albatrosCanStop, [17](#)
  - MetroStation, [17](#)
- moveAllePassengers
  - MetroNet, [14](#)
- moveAlleTrams
  - MetroNet, [14](#)
- movePassagiers
  - Station, [26](#)
- moveToBeginStation
  - Passagier, [20](#)
- moveTram
  - Tram, [31](#)
- PCC, [22](#)
  - PCC, [22](#)
  - stoptInStation, [23](#)
- Passagier, [18](#)
  - getBeginStation, [19](#)
  - getEindStation, [19](#)
  - getHoeveelheid, [19](#)
  - getNaam, [19](#)
  - isAangekomen, [19](#)
  - markAangekomen, [20](#)
  - moveToBeginStation, [20](#)
  - Passagier, [18](#)
  - setBeginStation, [20](#)
  - setEindStation, [20](#)
  - setHoeveelheid, [20](#)
  - setNaam, [21](#)
- PassagierInputTests, [21](#)
- removePassagier
  - Station, [27](#)
  - Tram, [31](#)
- runSimulation
  - MetroNet, [14](#)
- setAantalPassagiers
  - Tram, [31](#)
- setBeginStation
  - Passagier, [20](#)
  - Tram, [32](#)
- setCurrentStation
  - Tram, [32](#)
- setEindStation
  - Passagier, [20](#)
- setHoeveelheid
  - Passagier, [20](#)
- setLijnNr
  - Tram, [32](#)
- setNaam
  - Passagier, [21](#)
  - Station, [27](#)
- setOmzet
  - Tram, [32](#)
- setSnelheid
  - Tram, [32](#)
- setTramInStation
  - Station, [27](#)
- setVoertuigNr
  - Tram, [33](#)
- setZitplaatsen
  - Tram, [33](#)
- Station, [23](#)
  - addPassagier, [24](#)
  - addVolgende, [25](#)
  - addVorige, [25](#)
  - albatrosCanStop, [25](#)
  - getNaam, [25](#)
  - getSporen, [25](#)
  - getTramInStation, [26](#)
  - getVolgende, [26](#)
  - getVorige, [26](#)
  - isInStation, [26](#)
  - isTramInStation, [26](#)
  - movePassagiers, [26](#)
  - removePassagier, [27](#)
  - setNaam, [27](#)
  - setTramInStation, [27](#)
  - Station, [24](#)
- stoptInStation
  - Albatros, [8](#)
  - PCC, [23](#)
  - Tram, [33](#)
- Tram, [28](#)
  - addPassagier, [29](#)
  - afstappenInHalte, [29](#)
  - getAantalPassagiers, [30](#)
  - getBeginStation, [30](#)
  - getCurrentStation, [30](#)
  - getLijnNr, [30](#)
  - getOmzet, [30](#)
  - getSnelheid, [30](#)
  - getVoertuigNr, [31](#)
  - getZitplaatsen, [31](#)
  - isInTram, [31](#)
  - moveTram, [31](#)
  - removePassagier, [31](#)
  - setAantalPassagiers, [31](#)
  - setBeginStation, [32](#)
  - setCurrentStation, [32](#)
  - setLijnNr, [32](#)
  - setOmzet, [32](#)
  - setSnelheid, [32](#)
  - setVoertuigNr, [33](#)
  - setZitplaatsen, [33](#)
  - stoptInStation, [33](#)

Tram, [29](#)

writeOutputStream  
MetroNet, [14](#)