

LogikGatter

Erzeugt von Doxygen 1.9.4



<b>1 Hierarchie-Verzeichnis</b>	<b>1</b>
1.1 Klassenhierarchie	1
<b>2 Klassen-Verzeichnis</b>	<b>3</b>
2.1 Auflistung der Klassen	3
<b>3 Datei-Verzeichnis</b>	<b>5</b>
3.1 Auflistung der Dateien	5
<b>4 Klassen-Dokumentation</b>	<b>7</b>
4.1 binaerGatter Klassenreferenz	7
4.1.1 Beschreibung der Konstruktoren und Destruktoren	7
4.1.1.1 binaerGatter()	8
4.1.1.2 ~binaerGatter()	8
4.1.2 Dokumentation der Elementfunktionen	8
4.1.2.1 getEingang()	8
4.1.2.2 getPin()	8
4.1.2.3 getVerbunden()	9
4.1.2.4 setEingang()	9
4.1.3 Dokumentation der Datenelemente	9
4.1.3.1 eingang	9
4.1.3.2 pin	9
4.1.3.3 verbunden	10
4.2 gatter Klassenreferenz	10
4.2.1 Beschreibung der Konstruktoren und Destruktoren	10
4.2.1.1 gatter()	10
4.2.1.2 ~gatter()	11
4.2.2 Dokumentation der Elementfunktionen	11
4.2.2.1 gatterLogik()	11
4.2.2.2 getAusgang()	11
4.2.2.3 getName()	11
4.2.2.4 getNameChar()	11
4.2.2.5 getVerbunden()	12
4.2.2.6 setEingang()	12
4.2.3 Dokumentation der Datenelemente	12
4.2.3.1 ausgang	12
4.2.3.2 name	12
4.3 nichtGatter Klassenreferenz	13
4.3.1 Beschreibung der Konstruktoren und Destruktoren	13
4.3.1.1 nichtGatter()	13
4.3.1.2 ~nichtGatter()	13
4.3.2 Dokumentation der Elementfunktionen	13
4.3.2.1 gatterLogik()	14

4.4 oderGatter Klassenreferenz	14
4.4.1 Beschreibung der Konstruktoren und Destrukturen	14
4.4.1.1 oderGatter() [1/2]	15
4.4.1.2 oderGatter() [2/2]	15
4.4.1.3 ~oderGatter()	15
4.4.2 Dokumentation der Elementfunktionen	15
4.4.2.1 gatterLogik()	15
4.5 quelle Klassenreferenz	15
4.5.1 Beschreibung der Konstruktoren und Destrukturen	16
4.5.1.1 quelle()	16
4.5.1.2 ~quelle()	16
4.5.2 Dokumentation der Elementfunktionen	16
4.5.2.1 getBis()	16
4.5.2.2 operator"!"()	17
4.6 unaerGatter Klassenreferenz	17
4.6.1 Beschreibung der Konstruktoren und Destrukturen	17
4.6.1.1 unaerGatter()	18
4.6.1.2 ~unaerGatter()	18
4.6.2 Dokumentation der Elementfunktionen	18
4.6.2.1 getEingang()	18
4.6.2.2 getVerbunden()	18
4.6.2.3 setEingang()	19
4.6.3 Dokumentation der Datenelemente	19
4.6.3.1 eingang	19
4.6.3.2 verbunden	19
4.7 undGatter Klassenreferenz	19
4.7.1 Beschreibung der Konstruktoren und Destrukturen	20
4.7.1.1 undGatter() [1/2]	20
4.7.1.2 undGatter() [2/2]	20
4.7.1.3 ~undGatter()	20
4.7.2 Dokumentation der Elementfunktionen	20
4.7.2.1 gatterLogik()	20
4.8 verbindung Klassenreferenz	21
4.8.1 Beschreibung der Konstruktoren und Destrukturen	21
4.8.1.1 verbindung() [1/2]	21
4.8.1.2 verbindung() [2/2]	21
4.8.1.3 ~verbindung()	22
4.8.2 Dokumentation der Elementfunktionen	22
4.8.2.1 aktual()	22
4.8.2.2 getBis()	22
4.8.2.3 getPin()	22
4.8.2.4 getVon()	22

<b>5 Datei-Dokumentation</b>	<b>23</b>
5.1 gatter_logik.cpp-Dateireferenz	23
5.1.1 Ausführliche Beschreibung	23
5.1.2 Makro-Dokumentation	23
5.1.2.1 _CRT_SECURE_NO_WARNINGS	23
5.2 gatter_logik.hpp-Dateireferenz	24
5.3 gatter_logik.hpp	24
5.4 gatter_print.cpp-Dateireferenz	25
5.4.1 Ausführliche Beschreibung	25
5.4.2 Dokumentation der Funktionen	25
5.4.2.1 operator<<() [1/2]	25
5.4.2.2 operator<<() [2/2]	26
5.5 gatter_print.hpp-Dateireferenz	26
5.5.1 Dokumentation der Funktionen	26
5.5.1.1 operator<<() [1/2]	26
5.5.1.2 operator<<() [2/2]	27
5.6 gatter_print.hpp	27
5.7 gatter_sammlung.cpp-Dateireferenz	27
5.7.1 Ausführliche Beschreibung	28
5.7.2 Dokumentation der Funktionen	28
5.7.2.1 add()	28
5.7.2.2 deleteSammlung()	28
5.7.2.3 find()	29
5.7.3 Variablen-Dokumentation	29
5.7.3.1 element	29
5.7.3.2 gesamt	29
5.8 gatter_sammlung.hpp-Dateireferenz	29
5.8.1 Dokumentation der Funktionen	29
5.8.1.1 add()	29
5.8.1.2 deleteSammlung()	30
5.8.1.3 find()	30
5.9 gatter_sammlung.hpp	30
5.10 gatter_steuer.cpp-Dateireferenz	30
5.10.1 Ausführliche Beschreibung	31
5.11 gatter_steuer.hpp-Dateireferenz	31
5.12 gatter_steuer.hpp	31
5.13 main.cpp-Dateireferenz	32
5.13.1 Ausführliche Beschreibung	32
5.13.2 Makro-Dokumentation	32
5.13.2.1 _CRT_SECURE_NO_WARNINGS	32
5.13.3 Dokumentation der Funktionen	32
5.13.3.1 main()	32



# Kapitel 1

## Hierarchie-Verzeichnis

### 1.1 Klassenhierarchie

Die Liste der Ableitungen ist -mit Einschränkungen- alphabetisch sortiert:

gatter . . . . .	10
binaerGatter . . . . .	7
oderGatter . . . . .	14
undGatter . . . . .	19
unaerGatter . . . . .	17
nichtGatter . . . . .	13
quelle . . . . .	15
verbindung . . . . .	21





## Kapitel 2

# Klassen-Verzeichnis

### 2.1 Auflistung der Klassen

Hier folgt die Aufzählung aller Klassen, Strukturen, Varianten und Schnittstellen mit einer Kurzbeschreibung:

binaerGatter	7
gatter	10
nichtGatter	13
oderGatter	14
quelle	15
unaerGatter	17
undGatter	19
verbindung	21



## Kapitel 3

# Datei-Verzeichnis

### 3.1 Auflistung der Dateien

Hier folgt die Aufzählung aller Dateien mit einer Kurzbeschreibung:

<a href="#">gatter_logik.cpp</a>	
Klasse für die Verwirklichung von logischen Funktionen der Gatter . . . . .	23
<a href="#">gatter_logik.hpp</a> . . . . .	24
<a href="#">gatter_print.cpp</a>	
Klasse für die Darstellung Logikgatter . . . . .	25
<a href="#">gatter_print.hpp</a> . . . . .	26
<a href="#">gatter_sammlung.cpp</a>	
Klasse für die Speicherung von Verbindungen . . . . .	27
<a href="#">gatter_sammlung.hpp</a> . . . . .	29
<a href="#">gatter_steuer.cpp</a>	
Klasse für die Behandlung Verbindungen und Quellen . . . . .	30
<a href="#">gatter_steuer.hpp</a> . . . . .	31
<a href="#">main.cpp</a>	
LogikGatter - Grundlagen der Programmierung 2 . . . . .	32



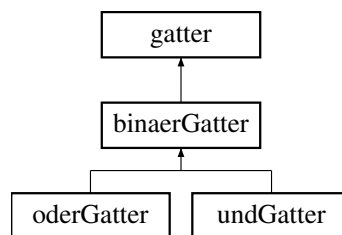
## Kapitel 4

# Klassen-Dokumentation

### 4.1 binaerGatter Klassenreferenz

```
#include <gatter_logik.hpp>
```

Klassendiagramm für binaerGatter:



#### Öffentliche Methoden

- `binaerGatter` (const char \*name, const int pin=2)  
*Default Konstruktor.*
- virtual bool `getEingang` (const int n)
- virtual void `setEingang` (const int n, const bool wert)  
*Einstellung des Eingangs des Gatters.*
- bool `getVerbunden` (const int n)
- int `getPin` ()
- `~binaerGatter` ()

#### Geschützte Attribute

- int `pin`
- bool \* `eingang`
- bool \* `verbunden`

#### 4.1.1 Beschreibung der Konstruktoren und Destruktoren

#### 4.1.1.1 binaerGatter()

```
binaerGatter::binaerGatter (
    const char * name,
    const int pin = 2 )
```

Default Konstruktor.

##### Parameter

<i>name</i>	Name des Gatters, erscheint bei Darstellung
<i>pin</i>	Anzahl der Eingänge

#### 4.1.1.2 ~binaerGatter()

```
binaerGatter::~binaerGatter ( )
```

Destruktor

### 4.1.2 Dokumentation der Elementfunktionen

#### 4.1.2.1 getEingang()

```
bool binaerGatter::getEingang (
    const int n ) [virtual]
```

##### Parameter

<i>n</i>	Nummer des bis-Gatter-Pins
----------	----------------------------

##### Rückgabe

true, falls am Eingang 1 ist  
false, falls am Eingang 0 ist

#### 4.1.2.2 getPin()

```
int binaerGatter::getPin ( )
```

##### Rückgabe

Anzahl der Pins des Gatters

#### 4.1.2.3 getVerbunden()

```
bool binaerGatter::getVerbunden (
    const int n ) [virtual]
```

##### Parameter

<i>n</i>	Nummer des Gatter-Pins
----------	------------------------

##### Rückgabe

true, falls der Pin Verbunden ist

false, falls der Pin nicht verbunden ist

Implementiert [gatter](#).

#### 4.1.2.4 setEingang()

```
void binaerGatter::setEingang (
    const int n,
    const bool wert ) [virtual]
```

Einstellung des Eingangs des Gatters.

##### Parameter

<i>n</i>	Nummer des Gatter-Pins
<i>wert</i>	Logischer Wert der Eingang

Implementiert [gatter](#).

### 4.1.3 Dokumentation der Datenelemente

#### 4.1.3.1 eingang

```
bool* binaerGatter::eingang [protected]
```

Logischer Wert an den Eingängen

#### 4.1.3.2 pin

```
int binaerGatter::pin [protected]
```

Anzahl der Pins

#### 4.1.3.3 verbunden

```
bool* binaerGatter::verbunden [protected]
```

Die Eingänge sind verbunden

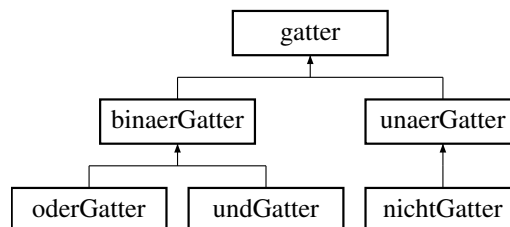
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

- [gatter\\_logik.hpp](#)
- [gatter\\_logik.cpp](#)

## 4.2 gatter Klassenreferenz

```
#include <gatter_logik.hpp>
```

Klassendiagramm für gatter:



### Öffentliche Methoden

- [gatter](#) (const char \*name)  
*Default Konstruktor.*
- char \* [getName](#) ()
- char [getNameChar](#) (const int n)
- bool [getAusgang](#) ()
- virtual bool [getVerbunden](#) (const int n)=0
- virtual bool [gatterLogik](#) ()=0
- virtual void [setEingang](#) (const int n, const bool wert)=0
- [~gatter](#) ()

### Geschützte Attribute

- char \* [name](#)
- bool [ausgang](#)

### 4.2.1 Beschreibung der Konstruktoren und Destruktoren

#### 4.2.1.1 gatter()

```
gatter::gatter (
    const char * name )
```

Default Konstruktor.



## Parameter

<i>name</i>	Name des Gatters, erscheint bei Darstellung
-------------	---

**4.2.1.2 ~gatter()**

```
gatter::~~gatter ( ) [inline]
```

**4.2.2 Dokumentation der Elementfunktionen****4.2.2.1 gatterLogik()**

```
virtual bool gatter::gatterLogik ( ) [pure virtual]
```

Implementiert in [undGatter](#), [oderGatter](#) und [nichtGatter](#).

**4.2.2.2 getAusgang()**

```
bool gatter::getAusgang ( )
```

## Rückgabe

Aktueller Ausgang des Gatters

**4.2.2.3 getName()**

```
char * gatter::getName ( )
```

## Rückgabe

Name des Gatters

**4.2.2.4 getNameChar()**

```
char gatter::getNameChar (
    const int n )
```

**Parameter**

<i>n</i>	
----------	--

**Rückgabe**

n-ter Charakter des Names

**4.2.2.5 getVerbunden()**

```
virtual bool gatter::getVerbunden (
    const int n ) [pure virtual]
```

Implementiert in [binaerGatter](#) und [unaerGatter](#).

**4.2.2.6 setEingang()**

```
virtual void gatter::setEingang (
    const int n,
    const bool wert ) [pure virtual]
```

Implementiert in [unaerGatter](#) und [binaerGatter](#).

**4.2.3 Dokumentation der Datenelemente****4.2.3.1 ausgang**

```
bool gatter::ausgang [protected]
```

Logischer Wert am Ausgang

**4.2.3.2 name**

```
char* gatter::name [protected]
```

Name des Gatters

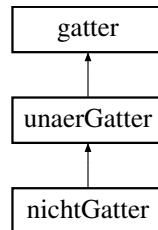
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

- [gatter\\_logik.hpp](#)
- [gatter\\_logik.cpp](#)

## 4.3 nichtGatter Klassenreferenz

```
#include <gatter_logik.hpp>
```

Klassendiagramm für nichtGatter:



### Öffentliche Methoden

- nichtGatter (const char \*name)
- virtual bool gatterLogik ()
- ~nichtGatter ()

### Weitere Geerbte Elemente

#### 4.3.1 Beschreibung der Konstruktoren und Destruktoren

##### 4.3.1.1 nichtGatter()

```
nichtGatter::nichtGatter (
    const char * name ) [inline]
```

##### 4.3.1.2 ~nichtGatter()

```
nichtGatter::~~nichtGatter ( ) [inline]
```

#### 4.3.2 Dokumentation der Elementfunktionen

#### 4.3.2.1 gatterLogik()

```
bool nichtGatter::gatterLogik ( ) [virtual]
```

Nicht Gatter - Logik

##### Rückgabe

true, falls am Ausgang 1 ist  
false, falls am Ausgang 0 ist

Implementiert [gatter](#).

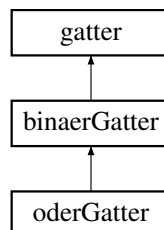
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

- [gatter\\_logik.hpp](#)
- [gatter\\_logik.cpp](#)

## 4.4 oderGatter Klassenreferenz

```
#include <gatter_logik.hpp>
```

Klassendiagramm für oderGatter:



### Öffentliche Methoden

- [oderGatter](#) (const char \*[name](#))
- [oderGatter](#) (const char \*[name](#), const int [pin](#))
- virtual bool [gatterLogik](#) ()
- [~oderGatter](#) ()

### Weitere Geerbte Elemente

#### 4.4.1 Beschreibung der Konstruktoren und Destruktoren

#### 4.4.1.1 oderGatter() [1/2]

```
oderGatter::oderGatter (
    const char * name ) [inline]
```

#### 4.4.1.2 oderGatter() [2/2]

```
oderGatter::oderGatter (
    const char * name,
    const int pin ) [inline]
```

#### 4.4.1.3 ~oderGatter()

```
oderGatter::~oderGatter ( ) [inline]
```

### 4.4.2 Dokumentation der Elementfunktionen

#### 4.4.2.1 gatterLogik()

```
bool oderGatter::gatterLogik ( ) [virtual]
```

Oder Gatter - Logik

##### Rückgabe

true, falls am Ausgang 1 ist  
false, falls am Ausgang 0 ist

Implementiert [gatter](#).

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

- [gatter\\_logik.hpp](#)
- [gatter\\_logik.cpp](#)

## 4.5 quelle Klassenreferenz

```
#include <gatter_steuer.hpp>
```

## Öffentliche Methoden

- `quelle (gatter *bis, const int n=1, const bool wert=true)`  
*Konstruktor.*
- `gatter & getBis ()`
- `quelle & operator! ()`
- `~quelle ()`

### 4.5.1 Beschreibung der Konstruktoren und Destruktoren

#### 4.5.1.1 `quelle()`

```
quelle::quelle (
    gatter * bis,
    const int n = 1,
    const bool wert = true )
```

Konstruktor.

Erzeugt eine Quelle

Parameter

<i>bis</i>	Pointer des bis-Gatters
<i>n</i>	Nummer des bis-Gatter-Pins, wo die Verbindung angeschlossen werden soll
<i>wert</i>	Logischer Wert des Gatters

#### 4.5.1.2 `~quelle()`

```
quelle::~~quelle ( )
```

Destruktor

### 4.5.2 Dokumentation der Elementfunktionen

#### 4.5.2.1 `getBis()`

```
gatter & quelle::getBis ( )
```

Rückgabe

Pointer auf das von-Gatter der Verbindung

#### 4.5.2.2 operator"!( )

```
quelle & quelle::operator! ( )
```

Negiert den aktuellen Wert der Quelle

#### Rückgabe

Referenz auf die originelle Quelle

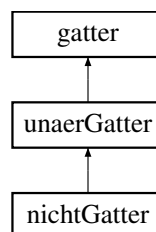
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

- [gatter\\_steuer.hpp](#)
- [gatter\\_steuer.cpp](#)

## 4.6 unaerGatter Klassenreferenz

```
#include <gatter_logik.hpp>
```

Klassendiagramm für unaerGatter:



### Öffentliche Methoden

- [unaerGatter](#) (const char \*[name](#))
- bool [getEingang](#) ()
- virtual bool [getVerbunden](#) (const int n=1)
- virtual void [setEingang](#) (const int n, const bool wert)  
*Stellt den Eingang des Gatters ein.*
- [~unaerGatter](#) ()

### Geschützte Attribute

- bool [eingang](#)
- bool [verbunden](#)

### 4.6.1 Beschreibung der Konstruktoren und Destruktoren

#### 4.6.1.1 unaerGatter()

```
unaerGatter::unaerGatter (
    const char * name ) [inline]
```

#### 4.6.1.2 ~unaerGatter()

```
unaerGatter::~~unaerGatter ( ) [inline]
```

### 4.6.2 Dokumentation der Elementfunktionen

#### 4.6.2.1 getEingang()

```
bool unaerGatter::getEingang ( )
```

##### Rückgabe

Eingang des Gatters mit einem Eingang

#### 4.6.2.2 getVerbunden()

```
bool unaerGatter::getVerbunden (
    const int n = 1 ) [virtual]
```

##### Parameter

<i>n</i>	Nummer des Gatter-Pins - automatisch 1 für unäre Gatter
----------	---

##### Rückgabe

true, wenn der Eingang des Gatters verbunden ist  
false, wenn der Eingang des Gatters nicht verbunden ist

Implementiert [gatter](#).



#### 4.6.2.3 setEingang()

```
void unaerGatter::setEingang (
    const int n,
    const bool wert ) [virtual]
```

Stellt den Eingang des Gatters ein.

##### Parameter

<i>n</i>	Nummer des Gatter-Pins - automatisch 1 für unäre Gatter
<i>wert</i>	Logischer Wert des Eingangs

Implementiert [gatter](#).

### 4.6.3 Dokumentation der Datenelemente

#### 4.6.3.1 eingang

```
bool unaerGatter::eingang [protected]
```

Logischer Wert am Eingang

#### 4.6.3.2 verbunden

```
bool unaerGatter::verbunden [protected]
```

Der Eingang ist verbunden

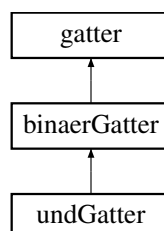
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

- [gatter\\_logik.hpp](#)
- [gatter\\_logik.cpp](#)

## 4.7 undGatter Klassenreferenz

```
#include <gatter_logik.hpp>
```

Klassendiagramm für undGatter:



## Öffentliche Methoden

- [undGatter](#) (const char \*[name](#))
- [undGatter](#) (const char \*[name](#), const int [pin](#))
- virtual bool [gatterLogik](#) ()
- [~undGatter](#) ()

## Weitere Geerbte Elemente

### 4.7.1 Beschreibung der Konstruktoren und Destruktoren

#### 4.7.1.1 undGatter() [1/2]

```
undGatter::undGatter (
    const char * name ) [inline]
```

#### 4.7.1.2 undGatter() [2/2]

```
undGatter::undGatter (
    const char * name,
    const int pin ) [inline]
```

#### 4.7.1.3 ~undGatter()

```
undGatter::~undGatter ( ) [inline]
```

### 4.7.2 Dokumentation der Elementfunktionen

#### 4.7.2.1 gatterLogik()

```
bool undGatter::gatterLogik ( ) [virtual]
```

Und Gatter - Logik

##### Rückgabe

true, falls am Ausgang 1 ist  
false, falls am Ausgang 0 ist

Implementiert [gatter](#).

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

- [gatter\\_logik.hpp](#)
- [gatter\\_logik.cpp](#)

## 4.8 verbindung Klassenreferenz

```
#include <gatter_steuer.hpp>
```

### Öffentliche Methoden

- `verbindung ()`  
*Default Konstruktor.*
- `verbindung (gatter *von, gatter *bis, const int n=1)`  
*Konstruktor.*
- `void aktual ()`
- `gatter * getVon ()`
- `gatter * getBis ()`
- `int getPin ()`
- `~verbindung ()`  
*Destruktor.*

### 4.8.1 Beschreibung der Konstruktoren und Destrukturen

#### 4.8.1.1 verbindung() [1/2]

```
verbindung::verbindung ( )
```

Default Konstruktor.

#### 4.8.1.2 verbindung() [2/2]

```
verbindung::verbindung (
    gatter * von,
    gatter * bis,
    const int n = 1 )
```

Konstruktor.

Stellt eine Verbindung zwischen zwei Gattern

#### Parameter

<i>von</i>	Pointer des von-Gatters
<i>bis</i>	Pointer des bis-Gatters
<i>n</i>	Nummer des bis-Gatter-Pins, wo die Verbindung angeschlossen werden soll

#### 4.8.1.3 ~verbindung()

```
verbindung::~~verbindung ( )
```

Destruktor.

### 4.8.2 Dokumentation der Elementfunktionen

#### 4.8.2.1 aktual()

```
void verbindung::aktual ( )
```

Aktualisiert das Ende der Verbindung. Muss abgerufen werden, wenn es eine Änderung am Eingang gibt

#### 4.8.2.2 getBis()

```
gatter * verbindung::getBis ( )
```

##### Rückgabe

Pointer auf das von-Gatter der Verbindung

#### 4.8.2.3 getPin()

```
int verbindung::getPin ( )
```

##### Rückgabe

Nummer des von-Gatter-Pins, wozu die Verbindung angeschlossen ist

#### 4.8.2.4 getVon()

```
gatter * verbindung::getVon ( )
```

##### Rückgabe

Pointer auf das von-Gatter der Verbindung

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

- [gatter\\_steuer.hpp](#)
- [gatter\\_steuer.cpp](#)

## Kapitel 5

# Datei-Dokumentation

### 5.1 gatter\_logik.cpp-Dateireferenz

Klasse für die Verwirklichung von logischen Funktionen der Gatter.

```
#include <iostream>
#include <cstring>
#include "gatter_logik.hpp"
```

#### Makrodefinitionen

- `#define _CRT_SECURE_NO_WARNINGS`

#### 5.1.1 Ausführliche Beschreibung

Klasse für die Verwirklichung von logischen Funktionen der Gatter.

Autor

Ferenc *Andras* Kiss

Datum

Mai 2022

#### 5.1.2 Makro-Dokumentation

##### 5.1.2.1 \_CRT\_SECURE\_NO\_WARNINGS

```
#define _CRT_SECURE_NO_WARNINGS
```

## 5.2 gatter\_logik.hpp-Dateireferenz

### Klassen

- class [gatter](#)
- class [unaerGatter](#)
- class [binaerGatter](#)
- class [undGatter](#)
- class [oderGatter](#)
- class [nichtGatter](#)

## 5.3 gatter\_logik.hpp

[gehe zur Dokumentation dieser Datei](#)

```
1 #ifndef GATTER_LOGIK_H
2 #define GATTER_LOGIK_H
3
4 using namespace std;
5
6 class gatter {
7 protected:
8     char* name;
9     bool ausgang;
10 public:
11     gatter(const char* name);
12     char* getName();
13     char getNameChar(const int n);
14     bool getAusgang();
15     virtual bool getVerbunden(const int n) = 0;
16     virtual bool gatterLogik() = 0;
17     virtual void setEingang(const int n, const bool wert) = 0;
18     ~gatter() { delete[] name; }
19 };
20
21 class unaerGatter : public gatter {
22 protected:
23     bool eingang;
24     bool verbunden;
25 public:
26     unaerGatter(const char* name) : gatter(name) {
27         verbunden = false;
28     }
29     bool getEingang();
30     virtual bool getVerbunden(const int n = 1);
31     virtual void setEingang(const int n, const bool wert);
32     ~unaerGatter() {}
33 };
34
35 class binaerGatter : public gatter {
36 protected:
37     int pin;
38     bool* eingang;
39     bool* verbunden;
40 public:
41     binaerGatter(const char* name, const int pin = 2);
42     virtual bool getEingang(const int n);
43     virtual void setEingang(const int n, const bool wert);
44     bool getVerbunden(const int n);
45     int getPin();
46     ~binaerGatter();
47 };
48
49 class undGatter : public binaerGatter {
50 public:
51     undGatter(const char* name) : binaerGatter(name) {}
52     undGatter(const char* name, const int pin) : binaerGatter(name, pin) {}
53     virtual bool gatterLogik();
54     ~undGatter() {}
55 };
56
57 class oderGatter : public binaerGatter {
58 public:
59     oderGatter(const char* name) : binaerGatter(name) {}
60     oderGatter(const char* name, const int pin) : binaerGatter(name, pin) {}
61     virtual bool gatterLogik();
```

```

62     ~oderGatter() {}
63 };
64
65 class nichtGatter : public unaerGatter {
66 public:
67     nichtGatter(const char* name) : unaerGatter(name) {};
68     virtual bool gatterLogik();
69     ~nichtGatter() {}
70 };
71
72 #endif

```

## 5.4 gatter\_print.cpp-Dateireferenz

Klasse für die Darstellung Logikgatter.

```

#include <iostream>
#include "gatter_print.hpp"
#include "gatter_logik.hpp"
#include "gatter_steuer.hpp"
#include "gatter_sammlung.hpp"

```

### Funktionen

- ostream & `operator<<` (ostream &out, `binaerGatter` &g)
- ostream & `operator<<` (ostream &out, `unaerGatter` &g)

### 5.4.1 Ausführliche Beschreibung

Klasse für die Darstellung Logikgatter.

Autor

Ferenc *Andras* Kiss

Datum

Mai 2022

### 5.4.2 Dokumentation der Funktionen

#### 5.4.2.1 `operator<<()` [1/2]

```

ostream & operator<< (
    ostream & out,
    binaerGatter & g )

```

Überladung des << Operators, um Gatter mit mehreren Eingängen darzustellen

**Parameter**

<i>out</i>	ostream Referenz
<i>g</i>	Referenz aus das zu darstellene Gatter

**Rückgabe**

ostream Referenz

**5.4.2.2 operator<<() [2/2]**

```
ostream & operator<< (
    ostream & out,
    unaerGatter & g )
```

Überladung des << Operators, um Gatter mit einem Eingang darzustellen

**Parameter**

<i>out</i>	ostream Referenz
<i>g</i>	Referenz aus das zu darstellene Gatter

**Rückgabe**

ostream Referenz

**5.5 gatter\_print.hpp-Dateireferenz**

```
#include <iostream>
#include "gatter_logik.hpp"
```

**Funktionen**

- ostream & **operator<<** (ostream &out, binaerGatter &g)
- ostream & **operator<<** (ostream &out, unaerGatter &g)

**5.5.1 Dokumentation der Funktionen****5.5.1.1 operator<<() [1/2]**

```
ostream & operator<< (
    ostream & out,
    binaerGatter & g )
```

Überladung des << Operators, um Gatter mit mehreren Eingängen darzustellen



## Parameter

<i>out</i>	ostream Referenz
<i>g</i>	Referenz aus das zu darstellene Gatter

## Rückgabe

ostream Referenz

**5.5.1.2 operator<<() [2/2]**

```
ostream & operator<< (
    ostream & out,
    unaerGatter & g )
```

Überladung des << Operators, um Gatter mit einem Eingang darzustellen

## Parameter

<i>out</i>	ostream Referenz
<i>g</i>	Referenz aus das zu darstellene Gatter

## Rückgabe

ostream Referenz

**5.6 gatter\_print.hpp**

[gehe zur Dokumentation dieser Datei](#)

```
1 #ifndef GATTER_PRINT_H
2 #define GATTER_PRINT_H
3
4 #include <iostream>
5
6 #include "gatter_logik.hpp"
7
8 ostream& operator<<(ostream& out, binaerGatter& g);
9 ostream& operator<<(ostream& out, unaerGatter& g);
10
11 #endif
```

**5.7 gatter\_sammlung.cpp-Dateireferenz**

Klasse für die Speicherung von Verbindungen.

```
#include <cstdint>
#include "gatter_logik.hpp"
#include "gatter_steuer.hpp"
#include "gatter_sammlung.hpp"
```

## Funktionen

- void `add` (const `verbindung` &*v*)
- `gatter` \* `find` (const `gatter` &*g*, const int *n*)
- void `deleteSammlung` ()

## Variablen

- `verbindung` \* `gesamt`
- int `element`

### 5.7.1 Ausführliche Beschreibung

Klasse für die Speicherung von Verbindungen.

#### Autor

Ferenc *Andras* Kiss

#### Datum

Mai 2022

### 5.7.2 Dokumentation der Funktionen

#### 5.7.2.1 `add()`

```
void add (
    const verbindung & v )
```

Fügt eine neue Verbindung hinzu

#### Parameter

<i>v</i>	Verbindung
----------	------------

#### 5.7.2.2 `deleteSammlung()`

```
void deleteSammlung ( )
```

Löscht das Arrays von Verbindungen

### 5.7.2.3 find()

```
gatter * find (
    const gatter & g,
    const int n )
```

Entscheidet, ob ein gegebenes Pin eines Gatters mit einem anderen Gatter verbunden ist

#### Parameter

<i>g</i>	Gatter
<i>n</i>	Pin

#### Rückgabe

Speicheradresse des anderen Gatters, wenn verbunden  
0, wenn nicht verbunden

## 5.7.3 Variablen-Dokumentation

### 5.7.3.1 element

```
int element
```

### 5.7.3.2 gesamt

```
verbindung* gesamt
```

## 5.8 gatter\_sammlung.hpp-Dateireferenz

### Funktionen

- void `add` (const `verbindung` &*v*)
- `gatter * find` (const `gatter` &*g*, const int *n*=1)
- void `deleteSammlung` ()

### 5.8.1 Dokumentation der Funktionen

#### 5.8.1.1 add()

```
void add (
    const verbindung & v )
```

Fügt eine neue Verbindung hinzu

**Parameter**

<i>v</i>	Verbindung
----------	------------

**5.8.1.2 deleteSammlung()**

```
void deleteSammlung ( )
```

Löscht das Arrays von Verbindungen

**5.8.1.3 find()**

```
gatter * find (
    const gatter & g,
    const int n )
```

Entscheidet, ob ein gegebenes Pin eines Gatters mit einem anderen Gatter verbunden ist

**Parameter**

<i>g</i>	Gatter
<i>n</i>	Pin

**Rückgabe**

Speicheradresse des anderen Gatters, wenn verbunden  
0, wenn nicht verbunden

**5.9 gatter\_sammlung.hpp**

[gehe zur Dokumentation dieser Datei](#)

```
1 #ifndef GATTER_SAMMLUNG_H
2 #define GATTER_SAMMLUNG_H
3
4 void add(const verbindung& v);
5 gatter* find(const gatter& g, const int n = 1);
6 void deleteSammlung();
7
8 #endif
```

**5.10 gatter\_steuer.cpp-Dateireferenz**

Klasse für die Behandlung Verbindungen und Quellen.

```
#include <iostream>
#include "gatter_steuer.hpp"
#include "gatter_logik.hpp"
#include "gatter_print.hpp"
```

### 5.10.1 Ausführliche Beschreibung

Klasse für die Behandlung Verbindungen und Quellen.

Autor

Ferenc *Andras* Kiss

Datum

Mai 2022

## 5.11 gatter\_steuer.hpp-Dateireferenz

```
#include "gatter_logik.hpp"
```

### Klassen

- class [verbindung](#)
- class [quelle](#)

## 5.12 gatter\_steuer.hpp

[gehe zur Dokumentation dieser Datei](#)

```
1 #ifndef GATTER_STEUER_H
2 #define GATTER_STEUER_H
3
4 #include "gatter_logik.hpp"
5
6 using namespace std;
7
8 class verbindung {
9 private:
10     gatter* von;
11     gatter* bis;
12     int n;
13 public:
14     verbindung();
15     verbindung(gatter* von, gatter* bis, const int n = 1);
16     void aktual();
17     gatter* getVon();
18     gatter* getBis();
19     int getPin();
20     ~verbindung();
21 };
22
23 class quelle {
24 private:
25     gatter* bis;
26     int n;
27     bool wert;
28 public:
29     quelle(gatter* bis, const int n = 1, const bool wert = true);
30     gatter& getBis();
31     quelle& operator! ();
32     ~quelle();
33 };
34
35 #endif
```

## 5.13 main.cpp-Dateireferenz

LogikGatter - Grundlagen der Programmierung 2.

```
#include <iostream>
#include <cstring>
#include <cstdlib>
#include "gatter_logik.hpp"
#include "gatter_print.hpp"
#include "gatter_steuer.hpp"
#include "gatter_sammlung.hpp"
```

### Makrodefinitionen

- `#define \_CRT\_SECURE\_NO\_WARNINGS`

### Funktionen

- `int main ()`

#### 5.13.1 Ausführliche Beschreibung

LogikGatter - Grundlagen der Programmierung 2.

Autor

Ferenc *Andras* Kiss

Datum

Mai 2022

#### 5.13.2 Makro-Dokumentation

##### 5.13.2.1 `_CRT_SECURE_NO_WARNINGS`

```
#define _CRT_SECURE_NO_WARNINGS
```

#### 5.13.3 Dokumentation der Funktionen

##### 5.13.3.1 `main()`

```
int main ( )
```

# Index

`_CRT_SECURE_NO_WARNINGS`

gatter\_logik.cpp, 23

main.cpp, 32

`~binaerGatter`

binaerGatter, 8

`~gatter`

gatter, 11

`~nichtGatter`

nichtGatter, 13

`~oderGatter`

oderGatter, 15

`~quelle`

quelle, 16

`~unaerGatter`

unaerGatter, 18

`~undGatter`

undGatter, 20

`~verbindung`

verbindung, 21

add

gatter\_sammlung.cpp, 28

gatter\_sammlung.hpp, 29

aktual

verbindung, 22

ausgang

gatter, 12

binaerGatter, 7

`~binaerGatter`, 8

binaerGatter, 7

eingang, 9

getEingang, 8

getPin, 8

getVerbunden, 8

pin, 9

setEingang, 9

verbunden, 9

deleteSammlung

gatter\_sammlung.cpp, 28

gatter\_sammlung.hpp, 30

eingang

binaerGatter, 9

unaerGatter, 19

element

gatter\_sammlung.cpp, 29

find

gatter\_sammlung.cpp, 28

gatter\_sammlung.hpp, 30

gatter, 10

`~gatter`, 11

ausgang, 12

gatter, 10

gatterLogik, 11

getAusgang, 11

getName, 11

getNameChar, 11

getVerbunden, 12

name, 12

setEingang, 12

gatter\_logik.cpp, 23

`_CRT_SECURE_NO_WARNINGS`, 23

gatter\_logik.hpp, 24

gatter\_print.cpp, 25

operator<<, 25, 26

gatter\_print.hpp, 26

operator<<, 26, 27

gatter\_sammlung.cpp, 27

add, 28

deleteSammlung, 28

element, 29

find, 28

gesamt, 29

gatter\_sammlung.hpp, 29

add, 29

deleteSammlung, 30

find, 30

gatter\_steuer.cpp, 30

gatter\_steuer.hpp, 31

gatterLogik

gatter, 11

nichtGatter, 13

oderGatter, 15

undGatter, 20

gesamt

gatter\_sammlung.cpp, 29

getAusgang

gatter, 11

getBis

quelle, 16

verbindung, 22

getEingang

binaerGatter, 8

unaerGatter, 18

getName

gatter, 11

getNameChar

- gatter, [11](#)
- getPin
  - binaerGatter, [8](#)
  - verbindung, [22](#)
- getVerbunden
  - binaerGatter, [8](#)
  - gatter, [12](#)
  - unaerGatter, [18](#)
- getVon
  - verbindung, [22](#)
- main
  - main.cpp, [32](#)
- main.cpp, [32](#)
  - \_CRT\_SECURE\_NO\_WARNINGS, [32](#)
  - main, [32](#)
- name
  - gatter, [12](#)
- nichtGatter, [13](#)
  - ~nichtGatter, [13](#)
  - gatterLogik, [13](#)
  - nichtGatter, [13](#)
- oderGatter, [14](#)
  - ~oderGatter, [15](#)
  - gatterLogik, [15](#)
  - oderGatter, [14](#), [15](#)
- operator!
  - quelle, [16](#)
- operator<<
  - gatter\_print.cpp, [25](#), [26](#)
  - gatter\_print.hpp, [26](#), [27](#)
- pin
  - binaerGatter, [9](#)
- quelle, [15](#)
  - ~quelle, [16](#)
  - getBis, [16](#)
  - operator!, [16](#)
  - quelle, [16](#)
- setEingang
  - binaerGatter, [9](#)
  - gatter, [12](#)
  - unaerGatter, [18](#)
- unaerGatter, [17](#)
  - ~unaerGatter, [18](#)
  - eingang, [19](#)
  - getEingang, [18](#)
  - getVerbunden, [18](#)
  - setEingang, [18](#)
  - unaerGatter, [17](#)
  - verbunden, [19](#)
- undGatter, [19](#)
  - ~undGatter, [20](#)
  - gatterLogik, [20](#)
  - undGatter, [20](#)
- verbindung, [21](#)
  - ~verbindung, [21](#)
  - aktual, [22](#)
  - getBis, [22](#)
  - getPin, [22](#)
  - getVon, [22](#)
  - verbindung, [21](#)
- verbunden
  - binaerGatter, [9](#)
  - unaerGatter, [19](#)