 - Lernmittel für moderne Technologien -	C/C++ - Programmierung	© Udo John www.lmt-verlag.de
	Konstruktoren	Seite 1 von 4

Werden Objekte einer Klasse angelegt sind die Elemente zunächst unbestimmt. Das heißt: Sie haben keinen konkreten sondern einen zufälligen Wert.


Das folgende Beispiel mit der Klasse CRechteck verdeutlicht dieses.

```
//Klasse ohne Konstruktor
#include<iostream>
#include<conio.h>
using namespace std;
class CRechteck
{
private:
    double laenge;
    double breite;
public:
    void setlaenge();
    void setbreite();
    double getflaeche();
};
void CRechteck::setlaenge()
{
    cout << "\nLaenge: ";
    cin >> laenge;
}
void CRechteck::setbreite()
{
    cout << "\nBreite: ";
    cin >> breite;
}
double CRechteck::getflaeche()
{
    return (laenge*breite);
}
int main()
{
    CRechteck wiese;
    cout<<"Flaeche: "<<wiese.getflaeche()<<endl;
    _getch();
    return(0);
}
```

Im Hauptprogramm wird für die Fläche ein unsinniger bzw. unbestimmter Wert ausgegeben!

Die Initialisierung der Elemente erfolgt über einen so genannten **Konstruktor**. Der Konstruktor ist eine Funktion, welche beim Anlegen des Objektes automatisch aufgerufen wird.

Der Name des Konstruktors muss identisch mit dem Klassennamen sein.

 - Lernmittel für moderne Technologien -	C/C++ - Programmierung	© Udo John www.lmt-verlag.de
	Konstruktoren	Seite 2 von 4


Außerdem darf der Konstruktor keinen Rückgabewert besitzen, auch nicht *void*!

Der Konstruktor muss unter den public-Methoden in der Klassenbeschreibung deklariert und wie die anderen Methoden initialisiert werden.

Man unterscheidet zwischen Standardconstructoren und allgemeinen Constructoren mit Übergabe- bzw. Vorgabewerten.

Das folgende Beispiel zeigt die Verwendung eines Standardconstructors ohne Parameter bzw. Übergabewerte. Die Initialisierung der Elemente erfolgt durch den Konstruktor, welcher beim Anlegen des Objektes automatisch aufgerufen wird.

```
//Klasse mit Standardkonstruktor
#include<iostream>
#include<conio.h>
using namespace std;
class CRechteck
{
private:
    double laenge;
    double breite;
public:
    CRechteck();           //Standardkonstruktor ohne Parameter
                          //Der Name ist der Klassenname - Kein Rückgabetyt
    void setlaenge();
    void setbreite();
    double getflaeche();
};
CRechteck::CRechteck()  //Initialisierung des Standardconstructors
{
    laenge=5; breite=3; //laenge und breite werden initialisiert
}
/***** Initialisierung der weiteren Methoden *****/
double CRechteck::getflaeche()
{
    return (laenge*breite);
}
int main()
{
    CRechteck wiese;
    cout<<"Flaeche: "<<wiese.getflaeche()<<endl;
/*
Der Aufruf des Constructors erfolgt automatisch - Es wird der Wert 15 für die
Fläche ausgegeben
*/
    _getch();
    return(0);
}
```


 - Lernmittel für moderne Technologien -	C/C++ - Programmierung	© Udo John www.lmt-verlag.de
	Konstruktoren	Seite 3 von 4

Gelegentlich ist es erwünscht, dass die Elemente der angelegten Objekte mit unterschiedlichen Werten initialisiert werden.

Zu diesem Zweck werden sogenannte allgemeine Konstruktoren verwendet. Allgemeine Konstruktoren besitzen wie Funktionen Übergabeparameter. Wie bei Funktionen müssen Reihenfolge und definierte Typen beim Anlegen des Objektes eingehalten werden.

In folgendem Beispiel wird ein Konstruktor angelegt, welcher zwei double-Werte für laenge und breite beim Anlegen des Objektes erwartet.

```
//Klasse mit allgemeinem Konstruktor und Übergabewerten
#include<iostream>
#include<conio.h>
using namespace std;
class CRechteck
{
private:
    double laenge;
    double breite;
public:
    CRechteck(double, double); //Allgemeiner Konstruktor mit Übergabewerten
    void setlaenge();
    void setbreite();
    double getflaeche();
};
//Initialisierung des Standardkonstruktors
CRechteck::CRechteck(double l,double b)
{
    laenge=l; breite=b; //laenge und breite werden initialisiert
}
/***** Initialisierung der weiteren Methoden *****/
double CRechteck::getflaeche()
{
    return (laenge*breite);
}
int main()
{
    CRechteck wiese(4,6); //Elemente werden initialisiert: laenge=4 breite=6
    cout<<"Flaeche: "<<wiese.getflaeche()<<endl;
    //Es wird der Wert 24 für die Fläche ausgegeben
    _getch();
    return(0);
}
```

 - Lernmittel für moderne Technologien -	C/C++ - Programmierung	© Udo John www.lmt-verlag.de
	Konstruktoren	Seite 4 von 4

Noch komfortabler lassen sich Objekte anlegen, wenn ein Konstruktor mit sogenannten Vorgabewerten erstellt wird.

Bei diesen Konstruktoren werden nur die Elemente mit den Übergabewerten initialisiert die angegeben werden. Alle anderen Elemente werden mit Vorgabewerten initialisiert. Die Vorgabewerte müssen bei der Deklaration des Konstruktors in der Klassenbeschreibung angegeben werden.

In folgendem Beispiel kann das Anlegen von Objekten ohne, mit einem oder mit zwei Übergabeparametern erfolgen. Fehlende Parameter werden mit den Vorgabewerten initialisiert.

```
//Klasse mit allgemeinem Konstruktor und Vorgabewerten
#include<iostream>
#include<conio.h>
using namespace std;
class CRechteck
{
private:
    double laenge;
    double breite;
public:
//Allgemeiner Konstruktor mit Vorgabewerten
    CRechteck(double l=5, double b=4);
    void setlaenge();
    void setbreite();
    double getflaeche();
};
//Initialisierung des Standardkonstruktors
CRechteck::CRechteck(double l,double b)    {
    laenge=l; breite=b; //laenge und breite werden initialisiert
}
/***** Initialisierung der weiteren Methoden *****/
double CRechteck::getflaeche()
{
    return (laenge*breite);
}
int main()
{
    CRechteck wiese1;           //Ohne Übergabewert
    CRechteck wiese2(6);       //Mit einem Parameter laenge=6
    CRechteck wiese3(7,3);     //Mit zwei Parametern laenge=7 breite=3
    cout<<"Flaeche1: "<<wiese1.getflaeche()<<endl; //Fläche = 20
    cout<<"Flaeche2: "<<wiese2.getflaeche()<<endl; //Fläche = 24
    cout<<"Flaeche3: "<<wiese3.getflaeche()<<endl; //Fläche = 21
    _getch();
    return(0);
}
```