

## Die Klasse Console V1.1

### Ziel, Inhalt

- Dieses Dokument beschreibt die Klasse *Console*. Mit dieser Klasse ist es möglich im Konsolenfenster gewisse Attribute (Farben etc.) zu setzen.

Die Klasse Console V1.1	1
Ziel, Inhalt	1
Die Klasse Console V1.1	2
Einführung	2
Die Schnittstelle	2
Einbinden in das Projekt	3
Anwendung und Beispiel	4
Erzeugen	4
Grösse ändern	4
Titel setzen	4
Farbe ändern	5
Cursor setzen	5
Komplettes Beispiel	6

## Die Klasse Console V1.1

### Einführung

Die Klasse Console dient dazu das das Konsolenfenster in einem Konsolenprojekt den eigenen Bedürfnissen anzupassen. Es ist möglich die Farbe des Textes und des Hintergrundes zu setzen, den Titel und die Grösse des Konsolenfensters (mit Einschränkungen) zu bestimmen und den Cursor gezielt an eine Position zu setzen.

### Die Schnittstelle

Die Schnittstelle der Klasse *Console* sieht so aus:

```
typedef unsigned int Color; // definiere einen Datentypen Color
                             // der einem UINT entspricht

const Color Black = 0x00;
const Color Blue  = 0x01;
const Color Green = 0x02;
const Color Red   = 0x04;
const Color Intensity = 0x08;
const Color White = Blue | Green | Red;

////////////////////////////////////

class Console
{
public:
    Console();    // Konstruktor

    ~Console();  // Destruktor

    void locate(int zeile, int spalte);
    void clear();
    // Die Farben können mit dem bitweisen oder-operator |
    // kombiniert werden
    void setColor(Color textColour,
                  Color backgroundColour);

    // den Titel des Fensters setzen
    void setTitle(const std::string& title);

    // die maximale Grösse des Fensters in
    // Anzahl Zeichen setzen
    void setSize(int width, int height);

private:
    HANDLE          _handle; // member variable
    unsigned int    _width;
    unsigned int    _height;
};
```

Wie man sieht gibt es nur wenige Farben. Es sind dies die drei Grundfarben. Durch „bitweise oder“ ist es möglich die Farben zu mischen, wie es für die Farbe weiss gemacht wurde. Eine weitere Möglichkeit ergibt sich aus dem zusätzlichen Flag Intensity, das auch mit oder verknüpft werden kann. Die Klasse besitzt einen Konstruktor und einen Destruktor. Ein Objekt der Klasse Console kann ganz einfach als Variable erzeugt werden, wie man im Beispiel weiter unten sehen wird.

Mit *locate* wird der Cursor an eine bestimmte Position gesetzt. So ist es möglich den Textbildschirm aufzuteilen, und ähnlich wie ein Formular mit Informationen zu füllen. Nach einem *locate* wird die nächste Ausgabe mittels *cout* an dieser Position stattfinden.

Die Methode *clear* löscht den Bildschirm.

Mit *setColor* werden die Farben für den Text und den Hintergrund gesetzt. Die Farben müssen vom Datentyp Color sein, von denen es nur die drei Grundfarben und ein Flag für die Intensität gibt. Die drei Grundfarben und das Intensitäts-Flag können mit dem oder operator (*|*) kombiniert werden. Siehe dazu das Beispiel weiter unten.

Mit *setTitle* kann der Titel des Fensters geändert werden.

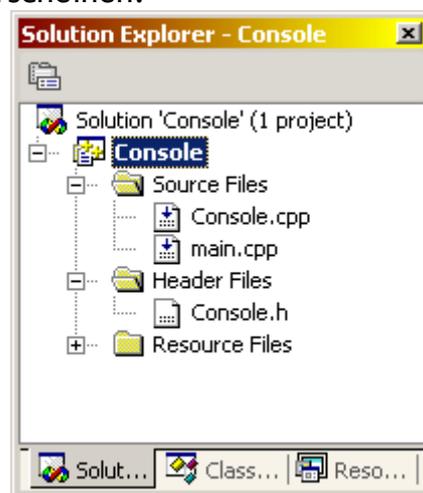
Die Methode *setSize* kann unter gewissen Einschränkungen verwendet werden die Fenstergrösse zu ändern. Dabei gibt es eine untere Grössenbeschränkung. Auch ist es nicht immer möglich das Fenster zu verkleinern.

## Einbinden in das Projekt

Um die Klasse in einem Konsolen-Projekt zu verwenden muss man die Header- und die Quellcode-Datei in das Projektverzeichnis kopieren. Die Dateien findest du auf

<http://www.devmentor.ch/teaching/additional/Console/Console.zip>.

Füge nun die Dateien im Visual Studio in das Projekt ein, so dass sie in der Dateien Baumansicht erscheinen:



## Anwendung und Beispiel

### Erzeugen

Um die Konsoleneigenschaften zu ändern muss man ein Objekt der Klasse `Console` erzeugen:

```
#include "Console.h"
#include <iostream>

using namespace std;

int main()
{
    Console theConsole;
```

Hier heisst das Objekt *theConsole*, ihr könnt es nennen wie ihr wollt. Natürlich muss die Datei `Console.h` inkludiert werden, damit die Klassendefinition bekannt ist. Für die Ausgabe wird weiterhin das Objekt *cout* verwendet!

### Grösse ändern

Um die Grösse des Fensters zu ändern wird die Methode `setSize` verwendet:

```
theConsole.setSize(100,40);
```

Das erste Argument gibt die Breite in Anzahl Zeichen an, währenddem das zweite Argument die Höhe in Anzahl Zeichen bestimmt.

Wie gesagt, kann die Grösse nicht beliebig geändert werden. Die Fenstergrösse des Konsolenfensters kann auch kleiner sein als die Anzahl darstellbarer Zeichen, was man an den Scrollbalken erkennen kann. Die Fenstergrösse kann auch mit der Maus geändert werden, aber nicht die Anzahl Zeichen im Fenster.

### Titel setzen

Der Titel des Fensters ist leicht setzbar:

```
theConsole.setTitle("Hallo");
```

### Farbe ändern

Die Farbe des Textes und die Hintergrundfarbe des Textes kann mit *setColor* gesetzt werden.

Die neuen gesetzten Farben gelten erst ab dem neu geschriebenen Text!

```
cout << "Alte Farben" << endl;
theConsole.setColor(Black, White | Intensity);
cout << "Neue Farben" << endl;
```



Um das ganze Fenster in der Hintergrundfarbe anzupassen kannst du einfach *clear* aufrufen!

### Cursor setzen

Mit *locate* lässt sich der Cursor setzen von wo aus die nächste Ausgabe mit *cout* stattfindet.

```
cout << "Alte Farben" << endl;
theConsole.setColor(Black, White | Intensity);
theConsole.locate(5, 10);
cout << "Neue Farben" << endl;
```



## Komplettes Beispiel

Hier ist ein vollständiges main. Arbeite das Programm mit dem Debugger Schritt für Schritt ab und schau dir die Auswirkung auf das Konsolenfenster an.

```
#include "Console.h"
#include <iostream>

using namespace std;

int main()
{
    Console theConsole;

    theConsole.setSize(100,40);

    theConsole.setTitle("Hallo");

    cout << "Alte Farben" << endl;
    theConsole.setColor(Black, White | Intensity);
    theConsole.locate(5, 10);
    cout << "Neue Farben" << endl;

    theConsole.clear();

    cout << "Test" << endl;

    int test;
    cin >> test;

    return 0;
}
```

Diese Datei kannst du auch vom Internet laden:

<http://www.devmentor.ch/teaching/additional/Console/main.cpp>.