

```
+-----cursor parking lot-----+
|                                   |
|                                   |
|                                   |
+-----+
+-----+
+-----+
```

Viel Glück :)

Abgabe 1

MCI- und WPF-Anwendungen

Ist MCI ein Zustandsautomat? Wenn ja, warum?

Lösung: (Folie 2)

Ein MCI-Gerät ist eine Zustandsmaschine, ähnlich einem Videorecorder.

Man schickt symbolische Kommandos an ein Gerät:

OPEN, PLAY, SEEK, STOP, RECORD, CLOSE, ...

Welche Medientypen kann man mit MCI wiedergeben?

Lösung:

-Sound: wav, midi, mp3, wma

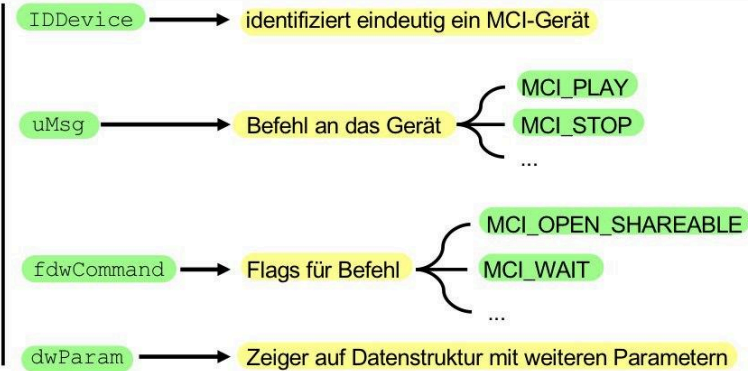
-Video: avi, mpg, mp4, wmv

-CD: Audio-CD, Video-CD

Was sind die Parameter von mciSendCommand?

Lösung:

```
MCIERROR mciSendCommand ( MCIDeviceID IDDevice,  
                          UINT uMsg,  
                          DWORD fdwCommand,  
                          DWORD dwParam );
```



die zweitwichtigste MCI-Funktion ->

Wie setzt man das time format und wann/wo?

in `openFile` oder `OpenAudioCD` mithilfe von `mciSendCommand`. Unterschied zwischen CD und anderen Dateien: Millisekunden oder

.

```

bool CMCIObject::OpenFile(LPCWSTR pszFileName)
{
    ... // wie oben
    MCI_SET_PARMS t;
    t.dwTimeFormat=MCI_FORMAT_MILLISECONDS;
    if ((m_Result = mciSendCommand(m_op.wDeviceID,
        MCI_SET,MCI_SET_TIME_FORMAT, (DWORD_PTR)&t)) != 0) {
        MCIError();
        return false;
    }
    return true;
}

bool CMCIObject::OpenAudioCD(LPCWSTR drive, BYTE &tracks)
{
    ... // wie oben
    // Set the time format to track, min, sek, frame
    MCI_SET_PARMS t;
    t.dwTimeFormat=MCI_FORMAT_TMSF;
    if ((m_Result = mciSendCommand(m_op.wDeviceID,
        MCI_SET,MCI_SET_TIME_FORMAT, (DWORD_PTR)&t)) != 0) {
        MCIError();
        return false;
    }
    return true;
}

```

<

“Erkläre Schrittweise die Methode getTrackLength();

Ziel → die Länge eines (bzw. aller) Audio-Tracks ermitteln ...

```

bool CMCIObject::GetTrackLength(BYTE track, BYTE &min, BYTE &sek, BYTE &frame) {

    if (m_op.wDeviceID == 0) return false; // Not open

    MCI_STATUS_PARMS length;           // ask for Track length
    length.dwTrack = track;
    length.dwItem = MCI_STATUS_LENGTH;
    if ((m_Result=mciSendCommand(m_op.wDeviceID, MCI_STATUS,
        MCI_STATUS_ITEM | MCI_TRACK,
        (DWORD_PTR)(LPMCI_STATUS_PARMS)&length)) != 0) {
        MCIError();
        return false;
    }
    // calculate return value using MSF-Format
    m_Result = length.dwReturn & 0x000000FF; min = (BYTE) m_Result;
    m_Result = length.dwReturn & 0x0000FF00; sek = (BYTE) (m_Result >> 8);
    m_Result = length.dwReturn & 0x00FF0000; frame = (BYTE) (m_Result >> 16);
    return true;
}

```

Erläuterung:

GetTrackLength(Übergabe Tracknummer, Zeiger auf Minute, Zeiger auf Sekunde, Zeiger auf Frame)

```

//Track wird auf Struktur (length) gepackt
MCI_STATUS_PARMS length; // ask for Track length
length.dwTrack = track;

//Das Item, was bei MCI_STATUS_ITEM abgefragt wird
length.dwItem = MCI_STATUS_LENGTH;

//mciSendCommand wieder Mustererkennung/Zugang zum Media Control Interface
if ((m_Result=mciSendCommand(m_op.wDeviceID, MCI_STATUS,
MCI_STATUS_ITEM | MCI_TRACK,
(DWORD_PTR)(LPMCI_STATUS_PARMS)&length)) != 0) {
MCIError();
return false;

//bitweise werden Minuten und Sekunden ermittelt (Frame wird nicht gebraucht)
// calculate return value using MSF-Format
m_Result = length.dwReturn & 0x000000FF; min = (BYTE) m_Result;
m_Result = length.dwReturn & 0x0000FF00; sek = (BYTE) (m_Result >> 8);
m_Result = length.dwReturn & 0x00FF0000; frame = (BYTE) (m_Result >> 16);
return true;
}

```

-> Woher weiß die Funktion, dass die Ausgabe im Byte Format ist (und nicht im Min/Sek Format)? L: wird festgelegt durch open audio cd (tracks = (BYTE)status.dwReturn;)

DirectSound

GenerateSound() ist in 3 Teile unterteilt, welche sind das?

Lösung: Lock, Sinuswellenformel, Unlock

mit LockBuffer und UnlockBuffer:

```
void *lpvPtr1, *lpvPtr2; DWORD dwBytes1, dwBytes2;
if (!m_ds.LockBuffer(lpDSBSecondary, 0, 2, // we use the first 2 seconds
                    &lpvPtr1,           // get pointer 1
                    &dwBytes1,           // get bytes available there
                    &lpvPtr2,           // get pointer 2 (the buffer is circular)
                    &dwBytes2))         // get bytes available there
    return false;

// write a sinus sound now (88040/63 = 1397 Hz)
DWORD i; short int *t; // points to 16 Bit
for (i=0, t=(short int*)lpvPtr1; i<(dwBytes1+dwBytes2); i+=4, t+=2) {
    if (i==dwBytes1) t=(short int*)lpvPtr2;
    // 2 channels with 16 Bit each
    *t = *(t+1) = (short int) (sin(i/10.0)*30000);
}

if (!m_ds.UnlockBuffer(lpDSBSecondary,
                       lpvPtr1,           // pointer 1
                       dwBytes1,         // bytes written there
                       lpvPtr2,         // pointer 2
                       dwBytes2))       // bytes written there
    return false;
```

Wie erstellt man ein object/instanz? (COM Objekt? und Interface?)

1.COM Interface im Konstruktor von CDirectSound

2. Objekt: in der Create Methode in 2 Schritten: DirectSound und Sound Buffer

if (!m_ds.Create(this))

OnCancel();

1. Erzeuge ein DirectSound - Interface ("repräsentiert die Soundkarte")

```
if (CoCreateInstance(CLSID_DirectSound, NULL, CLSCTX_ALL,
                    IID_IDirectSound, (LPVOID*) &lpds) != DS_OK) {
    AfxMessageBox(L"DirectSound object not created.");
    return false;
}
if (lpds->Initialize(0) != DS_OK) {
    AfxMessageBox(L"Error: Initialize DirectSound-Interface");
    return false;
}
```

Bemerkungen:

- lpds zeigt nun auf das initialisierte Interface-Objekt
- Sie haben die Wahl zwischen verschiedenen DSound-Versionen!!!

```
IID_IDirectSound (LPDIRECTSOUND)
IID_IDirectSound8 (LPDIRECTSOUND8)
```

Welche Parameter nimmt die CreateSoundBuffer (die für sekundäre DSB) Funktion?

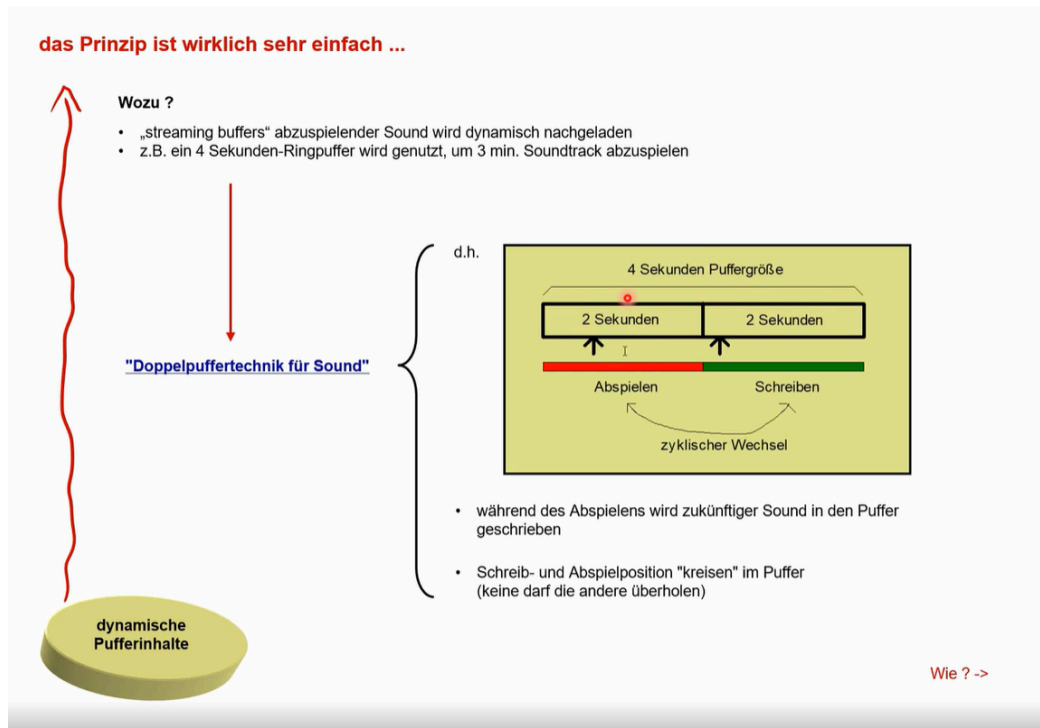
Lösung:

CreateSoundBuffer(WORD Channels, WORD BitsPerSample, DWORD SamplesPerSec, WORD seconds)

(WORD Channels: Kanäle -> 1 oder 2 (Mono / Stereo)
WORD seconds: Länge)

Wie kann man das "streamen" von einem audiobuffer der viel größer ist als der DSB realisieren?

Doppelpuffertechnik



Wenn es um viele Sekunden geht (also nicht nur 4), sollte es nicht statisch gespeichert werden d.h. dynamische Pufferinhalte mit der Doppelpuffertechnik (ig?). So wie man auch PCM realisiert hat.

Abgabe 2/3

Können Sie was zu Zeitformate sagen?

Nanosekunden-Maßstab beim Video?

Welche com-interfaces haben Sie verwendet?

die benötigten COM-Interfaces:

- IQueryBuilder
 - IMediaControl
 - IMediaEvent
- zum Aufbau eines Filtergraphen
- als Interface für den „gesamten“ Filtergraphen
- zur Behandlung der Ereignisse des Filtergraphen
- = Speicherung des Filtergraphen.*
- In*

die prinzipielle Vorgehensweise am Beispielcode:

erzeugt Filtergraph // Interface 1

1. der Filtergraph muss initialisiert werden:

```
IGraphBuilder *pGraph; // ein Zeiger auf das COM-Interface
CoInitialize(NULL);    // zur Initialisierung des COM-Interfaces
CoCreateInstance(CLSID_FilterGraph, NULL, CLSCTX_INPROC_SERVER,
                IID_IGraphBuilder, (void **) &pGraph);
```

↳ Unser Konstruktor

2. nun können die beiden anderen Interfaces initialisiert werden:

```
IMediaControl *pMediaControl; // Interface 2
IMediaEvent   *pEvent;        // Interface 3
pGraph->QueryInterface(IID_IMediaControl, (void **) &pMediaControl);
pGraph->QueryInterface(IID_IMediaEvent, (void **) &pEvent);
```

3. das neue Interface muss initialisiert werden

```
IVideoWindow *pVidWin = NULL; // Viertes Interface vom COM Objekt
pGraph->QueryInterface(IID_IVideoWindow, (void **) &pVidWin); // pGraph ist ein initialisierter Filtergraph
```

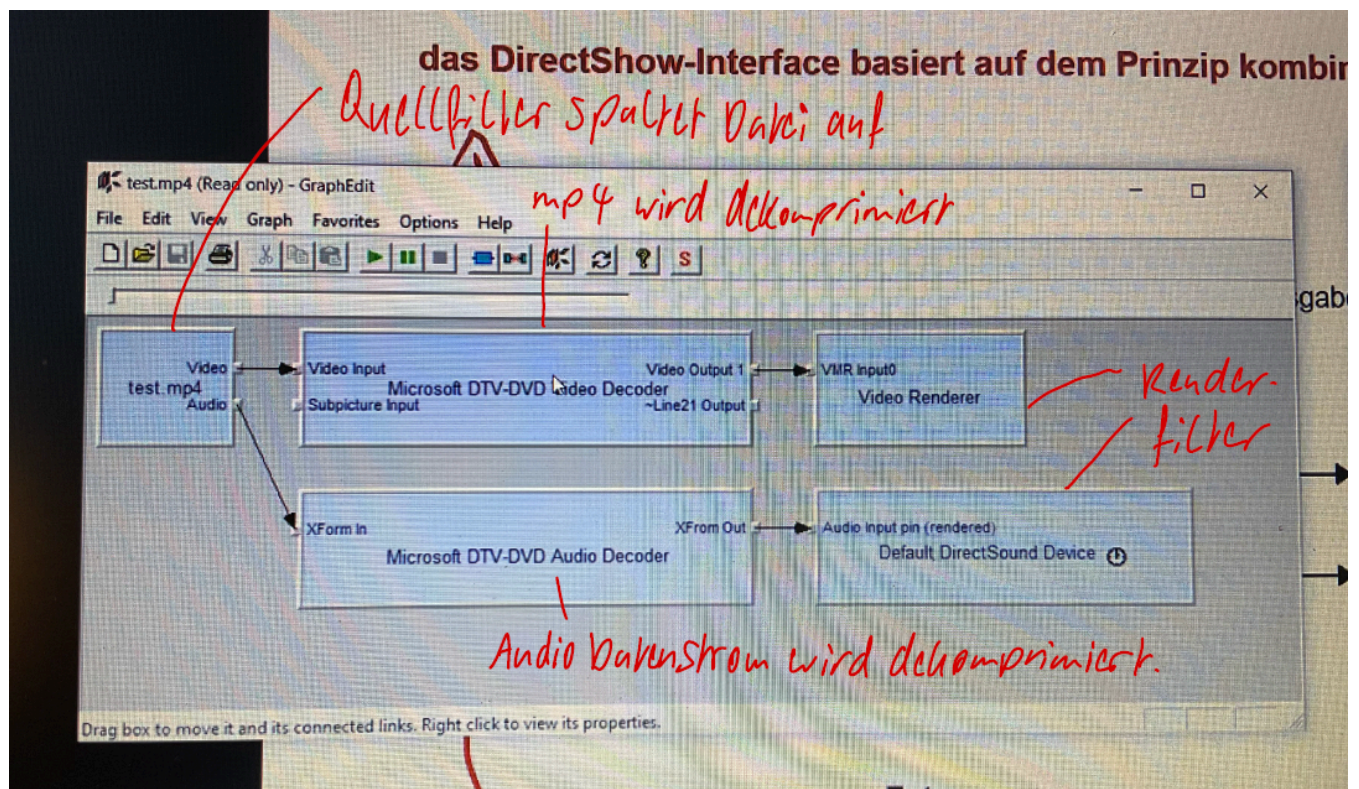
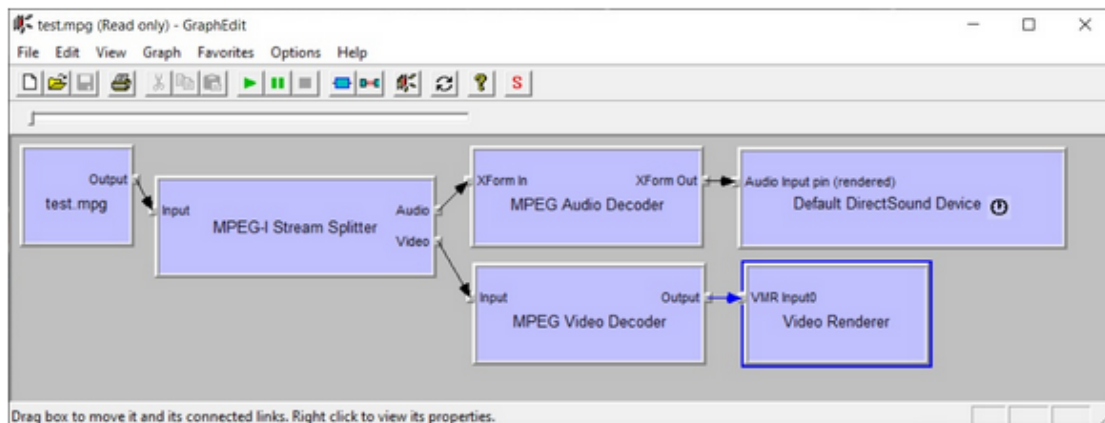
merke: wir initialisieren dieses Interface
genauso wie die bisher bekannten Interfaces

IMediaSeeking ist ein weiteres Interface.

```
IMediaSeeking *pSeek = NULL;
pGraph->QueryInterface(IID_IMediaSeeking, (void **) &pSeek);
```


wo wird der filtergraph aufgebaut?

diesen Filtergraphen plausibel und testen Sie diesen Graphen (Play, Pause, Stop).

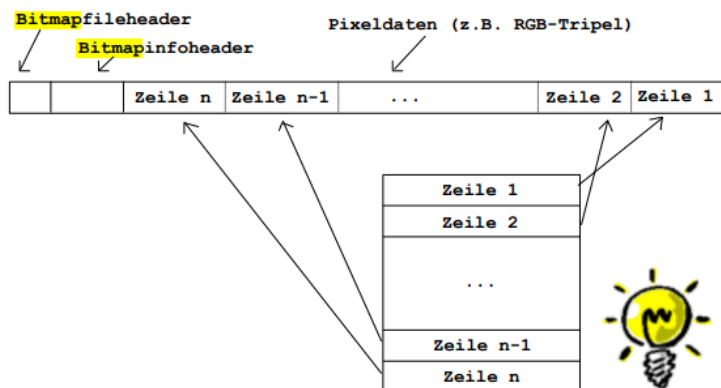


Im Code wird der Filtergraph an der Stelle
`pGraph->RenderFile(L"test.mp4", NULL);`
 aufgebaut.

An welcher Stelle legen wir die Kommunikation zu dem Filtergraphen fest, damit wir die Events (Play/Pause usw) von dem Filtergraph mitbekommen? (er hat bei mir nicht erwähnt, dass es das Event ist, auf das wir die Ressourcen dann wieder freigeben, das hat mich verwirrt. War ne fiese Frage vom Roeper, weiß nicht ob die öfters kommt)

`pEvent->SetNotifyWindow(parent, WM_GRAPHNOTIFY, 0);`

Wie ist eine Bitmap aufgebaut?



was mussten sie beim vollbildmodus beachten

eine neue Methode dient dem Schalten in und aus dem Vollbildmodus

```
void CxxxDlg::Vollbild(bool v) {
    if (pGraph) {
        IVideoWindow *pVidWin1 = NULL;
        pGraph->QueryInterface(IID_IVideoWindow, (void **)&pVidWin1);
        pVidWin1->put_FullScreenMode(v?OATRUE:OAFALSE);
        pVidWin1->Release();
    }
}
```

merke:

- vorher muss der Filtergraph aufgebaut worden sein, d.h. `pGraph->RenderFile()` wurde gerufen
- das `IVideoWindows`-Interface kann sofort wieder freigegeben werden, es diente uns "nur als Interface"

merke:

- Maus- und Tastaturereignisse im Vollbildmodus müssen über das `IVideoWindow`-Interface an die Dialogklasse geleitet werden !!!!!
- nach dem Initialisieren des `IVideoWindow`-Interfaces:

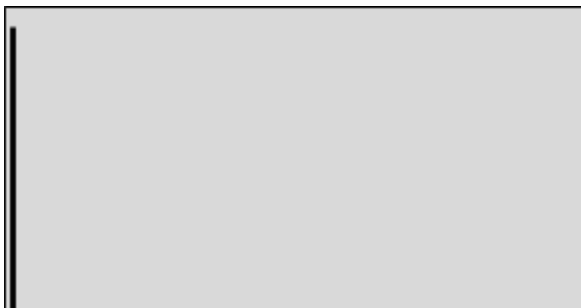
-> neues `IVideoWindow` Video- Interface muss zusätzlich erzeugt werden, neben dem vorhandenen Video Window in dem Dialog Fenster.

Wie wird das Histogramm erzeugt?

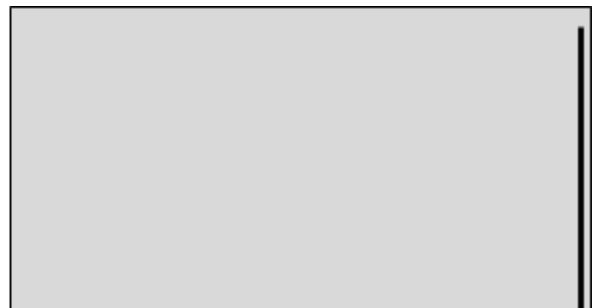
```
float h[256] = { 0.f }; int x = 10, y = 105; //
CClientDC dc(this);
m_dib.histogram(h, 20.f);
dc.MoveTo(x, y); dc.LineTo(x + 255 + 2, y); // Rahmen zeichnen
dc.LineTo(x + 255 + 2, y - 101); dc.LineTo(x, y - 101); dc.LineTo(x, y);
CPen p(PS_SOLID, 1, RGB(255, 255, 0)); dc.SelectObject(&p); //Farbeinstellung (Gelb)
for (int i = 0; i < 255; i++) { // Histogramm zeichnen
    dc.MoveTo(x + i + 1, y - 1);
    dc.LineTo(x + i + 1, y - 1 - (100 * h[i])); //gehen h durch; je nach häufigkeit des Wertes wird Linie dementsprechend "lang"
}
```

```
void CDIB::histogramm(float* h, float zoom) {
    if ((m_pBMFH == 0) || (m_pBMI->bmiHeader.biBitCount != 24))
        return;
    BYTE* t; BYTE g; int width = DibWidth() * 3;
    float step = 1.f / (DibHeight() * DibWidth());
    for (int i = 0; i < 255; i++) h[i] = 0.f; // init
    for (int i = 0; i < DibHeight(); i++) {
        t = (BYTE*)GetPixelAddress(0, i);
        for (int j = 0; j < width; j += 3) {
            g = (BYTE)(0.1145 * (*(t + j)) + 0.5866 * (*(t + j + 1)) + 0.2989 * (*(t + j + 2)));
            h[g] += step; // count g = Helligkeit; h=256mögliche Helligkeitswerte
        }
    }
    if (zoom != 0.0f) // zoom für Unanhängigkeit der Anzahl von Bildpunkten
        for (int i = 0; i < 255; i++) {
            h[i] *= zoom;
            if (h[i] > 1.f) h[i] = 1.f;
        }
}
```

Zu welchen Bildern gehören die folgenden Histogramme?



komplett schwarzes Bild



komplett weißes Bild

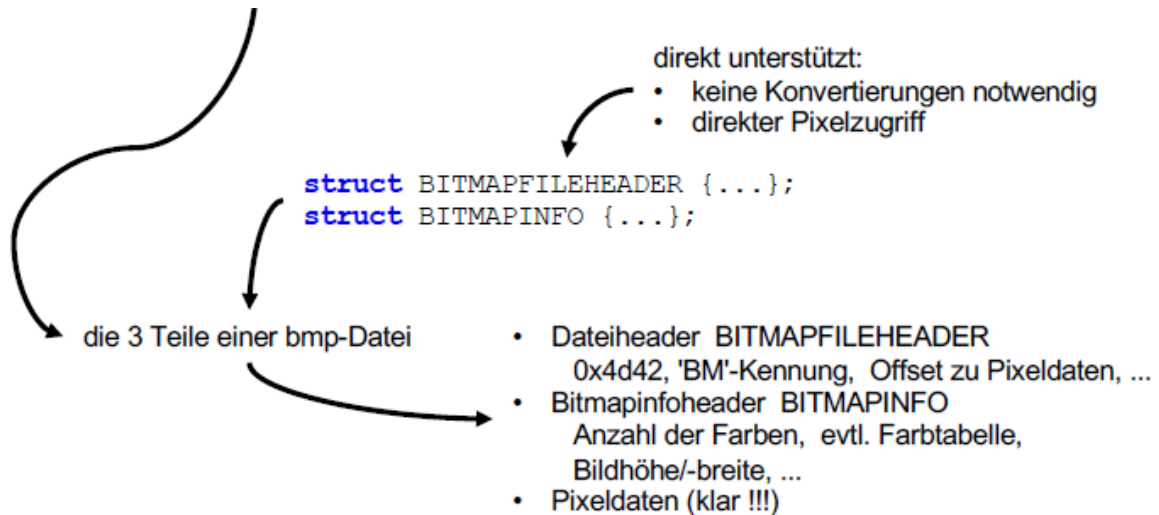
Wie funktioniert der Brightness Filter? Erklären, ohne den Code zu haben, er hat mit den Händen visualisiert, wo 255 ist, wo der aktuelle Farbwert ist und dass wir um 50% anheben wollen.

Abstand vom aktuellen Wert bis 255, den mit Skalierungsfaktor 0,5 multiplizieren und auf den aktuellen Wert drauf addieren.

Hier noch random Fragen aus vorherigen Semestern:

Wie ist BMP aufgebaut? (die Kanäle)

maybe das hier?



Aufhellen-Funktion erklären bei

Wie funktioniert der Pixel Filter?

Erkläre die genutzten Interfaces?

Wo wird der Filter Graph bei dem Video aufgebaut?