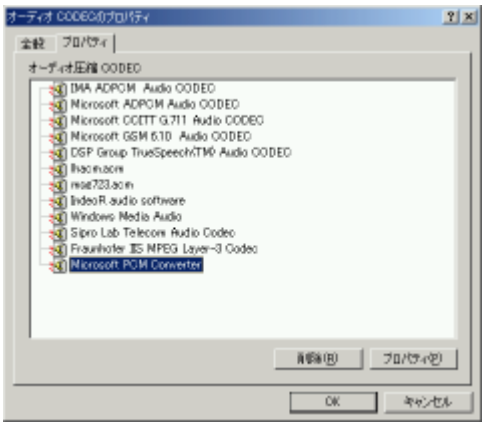
# オブジェクト指向と ゲームプログラミング

## ▶ DirectX Audio編 ― 第８回　ACMを用いたWaveファイルの読み込み ◀

### ＡＣＭ

ACM(Audio Compression Manager)とは、音声データを圧縮および伸張するAPI群のことをいいます。利用可能な圧縮形式は環境により異なり、マルチメディアのプロパティで確認することができます。




ACMに対応したコーデック(CODEC:COmpression/DECompression[圧縮および伸張を行うプログラム])をインストールしたり、読み込んだりすることにより、扱える圧縮形式を増やすことができます。

### 圧縮されたWaveファイル

圧縮されたWaveファイルは、すべてのデータが圧縮されているのではなく、dataチャンクだけ圧縮されています。つまり、ヘッダなどは通常のWaveファイルで、イメージだけ圧縮されているのです。圧縮されたWaveファイルを読み書きする場合は、ヘッダなどは通常どおり扱い、イメージだけACMを利用します。

イメージだけ圧縮したWaveファイルは、サウンドレコーダや音声編集ツールなどで作成することができます。

### ACMを利用したWaveファイルの読み込み

DirectSoundは、リニアPCMしかサポートしていないため、圧縮されたイメージをサウンドバッファに読み込んでも、正しく再生することはできません。圧縮されたイメージは、ACMを利用してリニアPCM形式に変換してからサウンドバッファに転送する必要があります。

ACMを利用した変換手順は、以下のようになります。

| | |
|---|---|
| ①変換後推奨フォーマットの取得 | acmFormatSuggest関数 |
| ②ACMストリームを開く | acmStreamOpen関数 |
| ③変換後サイズの取得 | acmStreamSize関数 |
| ④変換のためのヘッダの準備 | acmStreamPrepareHeader関数 |
| ⑤目的の形式に変換 | acmStreamConvert関数 |
| ⑥④のヘッダを破棄 | acmStreamUnprepareHeader関数 |
| ⑦ACMストリームを閉じる | acmClose関数 |

ACMの関数を使用する場合は、ヘッダファイル<mmreg.h>および<msacm.h>と静的リンクライブラリ"msacm32.lib"が必要になります。

# 課　題

以下のプログラムは、CDXAudio8::CreateBufferFromFile関数をACMを用いるように変更したものです。MSDNライブラリなどでACM関数について調べながら、プログラムを完成させましょう。

```
/*****************************************************************************************************/
/*                                          バッファ生成                                             */
/*****************************************************************************************************/
ISoundBuffer* CDXAudio8::CreateBufferFromFile(LPSTR inFileName)
{
    HMMIO               hFile      = NULL;
    HACMSTREAM          hACMStream = NULL;    // ACMストリーム
    LPVOID              pFormat    = NULL;    // フォーマット
    LPVOID              pImage     = NULL;    // イメージ

    ISoundBuffer*       pBuffer    = NULL;
    IDirectSoundBuffer8* pDSBuffer  = NULL;

    try {
        if(m_pDSound == NULL)
            throw "*** Error - DirectX Audio未初期化(CDXAudio8_CreateBufferFromFile)\n";

        // ファイルオープン
        hFile = ::?????????(inFileName, NULL, MMIO_READ | MMIO_ALLOCBUF);
        if(hFile == NULL)
            throw "*** Error - ファイルオープン失敗(CDXAudio8_CreateBufferFromFile)\n";

        MMCKINFO   ParentChunk, ChildChunk;
        ::ZeroMemory(&ParentChunk, sizeof(ParentChunk));
        ::ZeroMemory(&ChildChunk,  sizeof(ChildChunk ));

        // 'WAVE'チャンクへ進入
        ParentChunk.fccType = mmioFOURCC('W', 'A', 'V', 'E');
        if(::mmioDescend(hFile, &ParentChunk, NULL, MMIO_FINDRIFF) != MMSYSERR_NOERROR)
            throw "*** Error - 'WAVE'チャンクが見つかりません(CDXAudio8_CreateBufferFromFile)\n";

        // 'WAVE'チャンクの'fmt 'チャンクへ進入
        ChildChunk.ckid = mmioFOURCC('f', 'm', 't', ' ');
        if(::mmioDescend(hFile, &ChildChunk, &ParentChunk, MMIO_FINDCHUNK) != MMSYSERR_NOERROR)
            throw "*** Error - 'fmt 'チャンクが見つかりません(CDXAudio8_CreateBufferFromFile)\n";

        // フォーマット情報取得
        pFormat = new BYTE[ChildChunk.cksize];
        if((DWORD)::mmioRead(hFile, (HPSTR)pFormat, ChildChunk.cksize) != ChildChunk.cksize)
            throw "*** Error - フォーマット情報取得失敗(CDXAudio8_CreateBufferFromFile)\n";

        ::mmioAscend(hFile, &ChildChunk, 0);    // 'fmt 'チャンク退出

        // 'WAVE'チャンクの'data'チャンクへ進入
        ChildChunk.ckid = mmioFOURCC('d', 'a', 't', 'a');
        if(::mmioDescend(hFile, &ChildChunk, &ParentChunk, MMIO_FINDCHUNK) != MMSYSERR_NOERROR)
            throw "*** Error - 'data'チャンクが見つかりません(CDXAudio8_CreateBufferFromFile)\n";

        // リニアPCMに変換
        WAVEFORMATEX   NewFormat;
        ::ZeroMemory(&NewFormat, sizeof(NewFormat));

        // 推奨フォーマット取得
        NewFormat.wFormatTag = WAVE_FORMAT_PCM;
        if(::acmFormatSuggest(NULL, (LPWAVEFORMATEX)pFormat, &NewFormat, sizeof(NewFormat),
                              ACM_FORMATSUGGESTF_WFORMATTAG) != 0)
            throw "*** Error - 推奨フォーマット取得失敗(CDXAudio8_CreateBufferFromFile)\n";

        // ACMストリームオープン
        if(::acmStreamOpen(&hACMStream, NULL, (LPWAVEFORMATEX)pFormat, &NewFormat, NULL, 0, 0,
                           ACM_STREAMOPENF_NONREALTIME) != 0)
            throw "*** Error - ACMストリームオープン失敗(CDXAudio8_CreateBufferFromFile)\n";

        // フォーマット変換後取得
        DWORD   NewSize;
        if(::acmStreamSize(hACMStream, ChildChunk.cksize, &NewSize, ACM_STREAMSIZEF_SOURCE) != 0)
            throw "*** Error - デコードサイズ取得失敗(CDXAudio8_CreateBufferFromFile)\n";

        // 変換元イメージ読み込み
```

```
        pImage = new BYTE[ChildChunk.cksize];
        if((DWORD)::mmioRead(hFile, (HPSTR)pImage, ChildChunk.cksize) != ChildChunk.cksize)
            throw "*** Error - 変換元イメージ読み込み失敗(CDXAudio8_CreateBufferFromFile\n";

        // セカンダリバッファ生成
        DSBUFFERDESC   dsbd;
        ::ZeroMemory(&dsbd, sizeof(dsbd));
        dsbd.dwSize          = sizeof(dsbd);
        dsbd.dwBufferBytes   = ChildChunk.cksize;
        dsbd.dwReserved      = 0;
        dsbd.lpwfxFormat     = (LPWAVEFORMATEX)pFormat;
        dsbd.dwFlags         = DSBCAPS_LOCDEFER | DSBCAPS_CTRLVOLUME | DSBCAPS_CTRLPAN |
                               DSBCAPS_CTRLFREQUENCY | DSBCAPS_GETCURRENTPOSITION2;
        dsbd.guid3DAlgorithm = GUID_NULL;

        // DirectSoundBuffer生成
        IDirectSoundBuffer*   pdsb;
        if(FAILED(※ここは各自考えましょう))
            throw "*** Error - DirectSoundBuffer生成失敗(CDXAudio8_CreateBuffer)\n";

        // DirectSoundBuffer8取得
        const HRESULT   hr = pdsb->?????????????(IID_IDirectSoundBuffer8, (LPVOID*)&pDSBuffer);
        pdsb->Release();
        if(hr != S_OK)
            throw "*** Error - DirectSoundBuffer8取得失敗(CDXAudio8_CreateBuffer)\n";

        // バッファのロック
        LPVOID    pBuf[2];
        DWORD     BufBytes[2];
        if(pDSBuffer->????(0, 0, &pBuf[0], &BufBytes[0], &pBuf[1], &BufBytes[1], DSBLOCK_ENTIREBUFFER)
           != DS_OK)
            throw "*** Error - バッファロック失敗(CDXAudio8_CreateBuffer)\n";

        // ACMストリームヘッダ設定
        ACMSTREAMHEADER   acmsh;
        ZeroMemory(&acmsh, sizeof(acmsh));
        acmsh.cbStruct    = sizeof(acmsh);
        acmsh.fdwStatus   = 0;
        acmsh.pbSrc       = (LPBYTE)pImage;         // 変換元イメージ領域
        acmsh.cbSrcLength = ChildChunk.cksize;      // 変換元イメージ領域のサイズ
        acmsh.pbDst       = (LPBYTE)pBuf[0];        // 変換後イメージ領域
        acmsh.cbDstLength = BufBytes[0];            // 変換後イメージ領域のサイズ
        if(::acmStreamPrepareHeader(hACMStream, &acmsh, 0) != 0) {
            pDSBuffer->Unlock(pBuf[0], 0, pBuf[1], 0);
            throw "*** Error - ACMストリームヘッダ設定失敗(CDXAudio8_CreateBuffer)\n";
        }

        // 変換しながらバッファへ転送
        const MMRESULT   mmr = ::acmStreamConvert(hACMStream, &acmsh, 0);

        ::acmStreamUnprepareHeader(hACMStream, &acmsh, 0);
        pDSBuffer->Unlock(pBuf[0], BufBytes[0], pBuf[1], 0);
        if(mmr != 0)
            throw "*** Error - イメージ変換失敗(CDXAudio8_CreateBuffer)\n";

        pBuffer = new CSoundBuffer(pDSBuffer);
    } catch(LPCSTR ErrorString) {
        ::OutputDebugString(ErrorString);
        // 例外が発生した場合は、NULLサウンドバッファを生成
        pBuffer = new CNullSoundBuffer();
    }

    SafeRelease(pDSBuffer);
    delete[] pImage;
    delete[] pFormat;
    if(hACMStream != NULL)  ::acmStreamClose(hACMStream, 0);;
    if(hFile      != NULL)  ::mmioClose(hFile, 0);

    // バッファリストへ追加
    m_BufferList.?????????(pBuffer);

    return pBuffer;
}
```