Grundlagenkurs C++

Übungsblatt 8

Aufgabe 8-1 *DTMF-Generator* **(Präsenzaufgabe)**

In der Vermittlungstechnik wird zur Signalisierung auf dem Audiokanal (Inband-Signalisierung) für jedes zu übertragende Zeichen eine Kombination von zwei Tonfrequenzen verwendet. Das Signalisierungsverfahren ist als „dual tone multiple frequency“ (DTMF) Verfahren bekannt.
Es werden zwei Frequenzgruppen definiert, eine „obere“ Frequenzgruppe und eine untere Frequenzgruppe, jeweils mit vier Frequenzen. Die Signalisierung eines Zeichens erfolgt durch gleichzeitiges Abspielen zweier Töne, einer aus der oberen und einer aus der unteren Frequenzgruppe. Somit können 16 verschiedene Zeichen signalisiert werden, wie in nachfolgender Tabelle dargestellt:

	1209 Hz	**1336 Hz**	**1477 Hz**	**1633 Hz**
697 Hz	„1“	„2“	„3“	„A“
770 Hz	„4“	„5“	„6“	„B“
852 Hz	„7“	„8“	„9“	„C“
941 Hz	„*“	„0“	„#	„D“

Die typische Dauer eines DTMF Zeichens soll 50ms betragen, gefolgt von 50 ms Pause.
Es können also maximal 10 Zeichen pro Sekunde übertragen werden.

a) Sinusgenerator

Ein einfacher Sinusgenerator kann durch die folgende Differenzengleichung beschrieben werden:

```
y(t) =  2*cos(2*π*f/fs) * y(t-1) - y(t-2)
   mit folgenden Initialisierungswerten:
   y(-1) = 0 ;
   y(-2) = -A*sin(2*π*f/fs) ;
   A : Amplitude
```

Diese Gleichung ist für jeden Zeitpunkt `t` auszuwerten.
Zu jedem Zeitpunkt `t` hängt also der Ausgangwert des Sinusgenerators nur von den zwei unmittelbar vorangehenden Ausgangswerten `y(t-1)` und `y(t-2)` ab.

- Schreiben Sie eine Klasse `CSinusGenerator`, deren Konstruktor die Werte für die Abtastrate `fs` und die gewünschte Tonfrequenz `f` als Parameter erhält. Die Amplitude wird als dritter Parameter übergeben und hat den Defaultwert 1.0.

- Über die Methode `iInit(...)` kann der Generator jederzeit neu initialisiert werden. Die Parameter entsprechen denen des Konstruktors.

- Die Klasse soll die Methode `lfGetNextSample()` besitzen, die den neuen Wert nach obiger Formel aus den beiden vorangegangenen Werten berechnet und als `double` Wert zurückgibt.

b) DTMF-Generator

Zur Erzeugung eines DTMF Zeichens müssen zwei Sinusgeneratoren mit den nötigen Anfangswerten initialisiert werden. Die Einzelsignale werden dann in jedem Zeitschritt addiert.

- Schreiben Sie eine Klasse `CDTMFGenerator`, deren Konstruktor die erste und zweite Tonfrequenz sowie die Abtastrate übergeben werden. Die Amplitude kann als vierter, optionaler Parameter übergeben werden.

- Über die Methode `iInit(...)` kann der Generator jederzeit neu initialisiert werden. Die Parameter entsprechen denen des Konstruktors.

- Die Klasse soll die Methode `lfGetNextSample()` besitzen, die den neuen Wert als Summe der beiden Sinus-Samples berechnet und diesen als `double` Wert zurückgibt.
  Beachten Sie, dass die Amplituden der summierten Einzelschwingungen halb so groß sein muss wie die geforderte Gesamtamplitude des DTMF Signals.

Aufgabe 8-2 *DTMF-Mapping* **(Präsenzaufgabe)**

Um den DTMF-Generator komfortabel initialisieren zu können, soll eine Klasse für das Mapping der zu übertragenden Zeichen auf die Frequenzen der unteren und oberen Frequenzgruppe realisiert werden.

Schreiben Sie eine Klasse `CDTMFMapper`, die die oben angegebene Matrix der 16 Zeichen als `static const member` enthält. Die Klasse soll zusätzlich jeweils eine Frequenztabelle für die untere und obere Frequenzgruppe als `static const member enthalten.`

Hinweis: Verwenden Sie zur Definition der Dimensionen der Matrix und der Frequenzgruppen -Arrays geeignete `enum` Werte (`NOF_LOW_FREQ` und `NOF_HIGH_FREQ`). Vorteil: Die `enum` Werte können direkt innerhalb der Klassendeklaration initialisiert werden.

Folgende Methoden sollen als statische Methoden zur Verfügung gestellt werden:

- `static int iChar2GroupIndex(const char c, int& riLowIdx, int& riHighIdx);`

- `static int iChar2DTMF(const char c,double& rlfLowFreq, double& rlfHighFreq);`

- `static char cGroupIndex2Char(int iLowIdx, int iHighIdx);`

- `static int iIndex2DTMF(int iLowIdx, int iHighIdx, double& rlfLowFreq, double& rlfHighFreq);`

Mit dieser Klasse kann dann die Initialisierung des DTMF-Generators z.B. wie folgt aussehen:

```
...
        CDTMFGenerator cDTMF;
        double lfLow, lfHigh;
        CDTMFMapper::iChar2DTMF('A',lfLow,lfHigh);
        //init DTMF generator
        cDTMF.iInit(lfLow,lfHigh,8000,30000);
...
```

Aufgabe 8-3 *DTMF Wählverfahren* **(Hausaufgabe)**

Schreiben Sie ein Programm, das eine Telefonnummer als String einliest und die der Telefonnummer entsprechende DTMF-Tonfolge erzeugt und in ein Wave-File schreibt (Format: 8kHz, 16 Bit).

Die Signallänge je Zeichen soll 250 ms betragen, also 125 ms DTMF Ton gefolgt von 125 ms Pause.

Verwenden Sie dazu die Klassen `CWaveFile` und `CDTMFGenerator` sowie `CDTMFMapper`.
Nutzen Sie dabei die Fähigkeit von `CWaveFile`, auch mehrere Signalstücke hintereinander in das geöffnete Wave File schreiben zu können.

Geben Sie Ihre Lösung zu dieser Aufgabe in einer Datei `8-3.cpp` ab.