

# Inlämningsuppgift – CDregister

## Kortfattad beskrivning

Uppgiften går ut på att skriva ett program för att hantera ett register med kompaktskivor (CD). Grundläggande funktionalitet skall finnas med som t.ex. lägga till, söka efter, uppdatera och ta bort poster, lista alla befintliga poster osv. Programmet skall också kunna spara registret på fil samt läsa in sparad register från fil. Filformatet kan t.ex. vara något enkelt textfilsformat.

## 1 Programskrivning

Programmet består av två delar. En del som utgör en modell av registret och en del som utgör ett grafiskt användargränssnitt.

Det viktigaste är modellen. En del av denna kan lösas och testas först. Därefter kan man implementera det grafiska användargränssnittet. Innan man har ett sådant kan man testa modellen, t.ex genom att programmet läser in en fil för att skapa registret i minnet, skriver ut registret på fil igen och jämför resultatet.

### 1.1 Modellen för cdregistret

Ett cdregister består av en mängd CD. För varje CD lagras uppgift om artist, titel, antal spår, total speltid i minuter och inspelningsår. Följande klass kan beskriva en CD:

```
class CDNode
{
private:
    std::string artist;
    std::string titel;
    unsigned int speltid, antalspr, ar;
public:
    // Konstruktörer och destruktörer samt medlemsfunktioner
}
```

Uppgifterna om en CD kan byggas på med t.ex. genre och betyg.

I klassen ska det bl.a. finnas en konstruktor och funktioner för att sätta respektive hämta värdet för respektive fält.

I klassen som representerar registret skall finnas funktioner för att inläsning och utskrift av registret från och till fil. Funktioner skall också finnas för att lägga till, söka efter, hämta data för, uppdatera och ta bort poster.

## 1.2 Grafiskt användargränssnitt

Programmet ska ha ett grafiskt användargränssnitt programmerat i wxWidgets. De ingående posterna i registret presenteras i form av en lista där användaren överskådligt kan se informationen om alla sina CD. Det skall gå att via listan markera en CD antingen för att ta bort posten eller för att få uppdatera de ingåendefälten i posten. Ändringar som görs via grafiken skall medföra att modellen uppdateras på motsvarande sätt. Det skall också gå att lägga till en ny post på motsvarande sätt.

Möjliga grafiska element att använda för listan är dels a) wxGrid tillsammans med wxGridStringTable eller annan subclass till wxGridTableBase och dels b) wxDataViewCtrl med tillhörande hjälpklasser. Bägge dessa alternativ ger möjlighet till editering direkt i listan, men om man har problem att få detta att fungera, så kan inmatningen istället ske i separata textfält (wxTextCtrl). Inmatade värden i textfälten förs sedan över till listpresentationen och modellen.

Programmet ska vara bekvämt att använda. Det ska inte krascha om användaren skriver in konstiga värden i inmatningsfälten.

## 2 Hantering av textfil i programmet

### indata

Ett enkelt textfilformat kan användas, där varje post (CD) motsvarar en eller flera rader på filen. Filen placeras lämpligen i projektkatalogen. Filen läses in vid programmets uppstart. Uppdateringar av registret sparas via menykommando. Om det finns osparade ändringar när programmet avslutas, så skall en dialogruta visas där användaren får välja om ändringarna skall sparas eller inte.

### Uppdatering av fil

Det är tillåtet att använda en textfil som datafil och att vid varje uppdatering av filen skriva över den gamla filen helt. I så fall skall den gamla filen sparas som en backup för att gardera sig mot dataförlust om programmet kraschar under sparningen av filen. Funktionerna `rename()` och `remove()` i `<cstdio>` kan användas för detta.

### Testdata vid inlämning

En datafil med lämpligt innehåll för testning skall bifogas vid inlämningen av uppgiften.