Linköpings Tekniska Högskola
Institutionen för Datavetenskap (IDA)
Torbjörn Jonsson, Erik Nilsson 2015-05-26

Lab 4: Hantering av strömmar

I denna laboration kommer du att lära dig hur man läser och skriver till filer.

Mål

I denna laboration kommer du att lära dig:

- hur man läser och skriver till textfiler.
- hur man delar upp sin kod i fler header- och implementationsfiler.

Uppgift

Du skall i denna uppgift skriva ett match-making program för legendariska hjältar. Programmet skall låta användaren mata in ett antal intressen och skall sedan leta igenom ett register med hjältar och skriv ut hjältar med matchande intressen. Intressen representeras i denna uppgift med heltal i intervallet [1, 15]. Registret med hjältar ligger på en separat textfil, REGISTER.TXT. Utdatat från programmet skall skrivas till en ny fil, RESULT.TXT.

Del A: Hjältedatastrukturen

Skapa först en datastruktur som representerar en hjälte. Som du kanske kommer ihåg så har hjältar följande karaktäristika:

- Ett namn
- En ålder
- Ett kön
- En vikt
- En hårfärg
- En art (något av: Human, Elf, Orc, Halfling, Ogre, Lizardman)
- En ögonfärg (något av: Blue, Green, Brown, Gray, Yellow, Red, Black, Crazy)

Som i Ada-delen är det frivilligt att implementera de två sista delarna av posten. Datastrukturen och tillhörande underprogramsdeklarationer skall placeras på en separat header-fil (t.ex. "hero.h"). Underprogramsdefinitionerna skall placeras i en tillhörande cc-fil ("hero.cc").

Till ovanstående tillkommer dock även att varje hjälte har ett fält av intressen (d.v.s heltal i intervallet [1, 15]). Rimligtvis använder du en vector för detta.

TIPS: Skriv ett litet huvudprogram som testar att läsa in data till din datastruktur och skriver ut den igen på skärmen. Detta för att separat kunna testa denna del istället för att vänta till slutet.

Linköpings Tekniska Högskola
Institutionen för Datavetenskap (IDA)
Torbjörn Jonsson, Erik Nilsson 2015-05-26

Del B: Registerdatastrukturen

Du skall nu skapa en ny headerfil som innehåller datatypen som representerar alla registrerade hjältar. Du skall även här använda dig av vector för att implementera denna datatyp. Skriv ett underprogram som läser in till datastrukturen från en filström. Skriv även ett underprogram som skriver ut hela registret till en fil. Deklarationerna för dessa underprogram lägger du givetvis i headerfilen och själva underprogramsdefinitionerna läggs i den tillhörande implementationsfilen.

TIPS: Tänk till när du gör din implementation. Det är meningen att du skall använda dig av så mycket som möjligt av det som du gjorde i del A, utan att klippa in någon kod.

Del C: Huvudprogrammet

Huvudprogrammet skall bestå av en meny med två alternativ. Antingen kan användaren välja att mata in en ny hjälte, som då skall läggas till i registret (som skall vara sorterat på något sätt), (och även i registerfilen).

Det andra alternativet är att få mata in ett antal intressen (maximalt 10 stycken). Endast tal i intervallet [1, 15] skall accepteras, övriga skall ignoreras. Inmatningen skall upphöra då en nolla (0) matas in av användaren.

Efter att inmatningen har skett skall programmet läsa in registerfilen, REGISTER.TXT, och gallra ut alla hjältar som inte har några matchande intressen. Därpå skall registret skrivas ut till utdatafilen RESULT.TXT.

Körexempel:

```
Välkommen till Hero Match-Maker 3000!
Välj ett alternativ:
  A) Mata in en ny hjälte
  B) Hitta matchande hjältar
Välj ett alternativ: B

Mata in dina intressen: 1 2 14 0
Klart! Nu finns de matchande hjältarna på filen RESULTAT.TXT
```

Del D: Huvudprogrammet

Lägg in en fördröjning i ditt underprogram som skriver till registerfilen efter att varje hjältes data skrivs. Kör sedan ditt program igen, lägg till en ny hjälte, men avbryt programmet med Ctrl-C medan hjältarna skrivs till filen.

Vad hände med registerfilen?

Går detta att undvika på något sätt?