Objektorienterad mjukvaruutveckling  
Laboration 4

Modell till kod

# Målsättning

Förstå hur man omvandlar modeller till kod

# Förberedelse

Gå igenom föreläsning 12 ”UML som kod” och koden till applikationen som finns i filen Bhockey.zip.

# Examination

En vecka efter övningen startade skall programkoden presenteras och testas. Man skall då kunna besvara frågor angående programkoden.

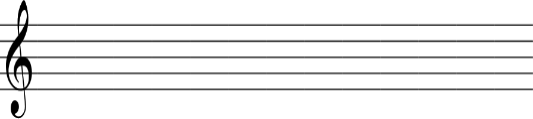
Genomförandet

Laborationen utförs i en grupp om två personer. Man skall skriva all programkod själva. Koden får inte vara likadan som någon annans kod.

Uppgifter

1. UML-diagram skall användas för att skapa ett kompositörsprogram. Man markera en not och klickar seden någonstans på notsystemet tills man är nöjd. Då kan man klicka på spela så spelas melodin upp, se nedanstående bild. 

Figur 1 Förslag till användarinterface för kompositören.

1. Börja med att skapa användarinterfacet. Skapa nu ett QT-project med namnet kompositor. Leta upp mappen Form Files i Solution Explorer. Öppna filen kompositor som finns där med hjälp av Qt Designer. Det borde räcka med att dubbelklicka på filnamnet, annars kan man göra Qt Designer som standardval.
   1. Placera alla bild-filerna i projektets Resources-katalog. Notsymbolerna skapas genom att välja Label och sedan välja respektive bildfil som pixmap bland egenskaperna. Placera sedan bilden på lämpligt ställe.   
       half.png quater.png eight.pngquaterest.pngeightrest.pngnotblad.png
   2. Knapparna erhålls genom att välja Radio Button
   3. Till spela-knappen används en Push Button och rubriken kan skapas m h a en Label
   4. Notera namnet på de ovanstående komponenterna som blir objekt med namnet ”ui.” följt av namnet i programmet. Du kan se hur de skapas i filen ui\_”projektnamnet”.h i mappen Generated Files.
2. Implementera nu klasserna i nedanstående klassdiagram så att de fungerar som nedanstående sekvensdiagram. Endast en typ av not skall användas än så länge.  
   



Om notbladet placeras med X=100,Y=300,Width=931 och Height=143 kan nedanstående kod användas.  
  
int Kompositor::pixelToHöjd(int ypixel)\*

{

float enivå = 413;

float step = 8;

int höjd = -1;

for (int i = 0; i < 12; i++)

{

if (ypixel <= (enivå + (2 - i)\*step + step / 2) && ypixel > (enivå + (2 - i)\*step - step / 2))

höjd = i;

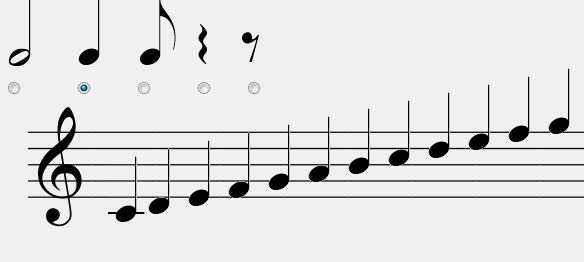
}

return höjd;

}

1. Fortsätt sedan att rita ut noterna enligt nedan



1. Nedan visas olika nothöjder. Noten med höjd 0 skall visas med ett streck igenom. Dessa bildfiler heter halfC.png, quaterC.png och eightC.png.   
     
   

Höjd: 0 1 2 3 4 5 6 7 8 9 10 11

Namn: C D E F G A B C D E F G  
  
 Fig 2 Nothöjder

1. Lägga till att kunna spela upp kompositionen, se diagram nedan. Synthesizer-klassen bifogas.





1. Bonusuppgift:  
   Avsluta med att lägga till olika noter och pauser. Pauserna quaterest.pngeightrest.png skall inte ändra höjd.   
   

Fig 3 Pauser alltid på samma höjd.

