Applikationsutveckling i Java

Obligatorisk uppgift 2 – Trådar och XML

**Utförd av:** William Larsson **Datum:** 2020-01-29

Andra inlämningen

**Kursansvarig:** Johan Eliasson  
**Övriga lärare:** Elias Åström, Joel Sandman,

Klas Af Geijerstam Unger, Anders Broberg.  
  
 Godkänd

Retur  
  
**Rättad av:** **Datum:**

Innehållsförteckning

[Inledning 3](#_Toc31271309)

[Problembeskrivning 3](#_Toc31271310)

[Användarhandledning 3](#_Toc31271311)

[Att starta programmet 3](#_Toc31271312)

[Kompilering av systemet 3](#_Toc31271313)

[Systembeskrivning 4](#_Toc31271314)

[Programbeskrivning 4](#_Toc31271315)

[Grafiskt användargränssnitt 4](#_Toc31271316)

[Implementation 5](#_Toc31271317)

[Implementation av systembeskrivningen 5](#_Toc31271318)

[Trådsäkerhet 5](#_Toc31271319)

[Designmönster 5](#_Toc31271320)

[Adapter-mönster 5](#_Toc31271321)

[Proxy-mönster 5](#_Toc31271322)

[Observer/Observable-mönster 5](#_Toc31271323)

[Template-mönster 5](#_Toc31271324)

[Gränssnitt 6](#_Toc31271325)

[Kanalyvyn 6](#_Toc31271326)

[Tablåvyn 7](#_Toc31271327)

[Menyn 7](#_Toc31271328)

[Lösningens begränsningar 8](#_Toc31271329)

[Vidareutveckling 8](#_Toc31271330)

# Inledning

Syftet med detta projekt är att ge grundläggande kunskaper och färdigheter i applikationsutveckling med hjälp av Java och Swing. Detta genom att skapa en applikation som implementerar klasser och metoder för att tolka XML-dokument, skriva trådsäker kod och skapa ett eget användargränssnitt. Utöver detta så ska ytterligare färdigheter inom att följa ett givet dataformat, följa kodkonventioner för Java samt rapportskrivning erhållas.

# Problembeskrivning

En applikation ska utvecklas med genom biblioteket Swing som tillhandahåller verktyg och komponenter för att skapa grafiska gränssnitt inom programmeringsspråket Java. Applikationen i sig ska implementera Swing och använda dess funktionaliteter för trådar och trådsäkerhet och grafiska komponenter för att skapa en applikation som inte låser sig när applikationen gör tunga och tidskrävande operationer. Inom applikationen ska en meny implementeras för att utföra olika operationer på programmet och en tabell ska skapas för att visa information till användaren. Applikationen ska även kunna förmedla informerande beskrivningar till användaren och visa upp bilder. Programmet ska även läsa in XML från en utomstående källa och denna information kommer att variera. Därmed ska det även vara möjligt att uppdatera informationen som visas inom applikationen.

# Användarhandledning

För att kunna följa denna beskrivning så antas användaren använda *Java 11* eller senare versioner. Om inte så kan det hända att följande instruktioner inte fungerar som de är tänkt.

## Att starta programmet

För att starta applikationen, öppna terminal/kommandotolken och navigera fram till mappen dit .zip-filen har packats upp. Navigera till undermappen ”sökväg till Jar”. I denna mapp återfinns den körbara .jar-filen.

Därefter kan applikationen köras genom att skriva följande kommande in i terminalen:

>java -jar NAME.jar

Applikationen kommer då att öppna i en ny programruta på skärmen. För förklaring av användargränssnittet, se Systembeskrivning – Grafiskt användargränssnitt.

## Kompilering av systemet

Eftersom detta projekt är byggt med Maven så är det rekommenderat att låta en IDE göra kompileringen av filerna åt dig. Detta då det vanliga kompileringsmetoden med Javac inte kommer att vara medveten om pom.xml-filen när den bygger projektet, vilket göra att programmet inte kommer att bli tilldelad alla förnödenheter på ett korrekt sätt.

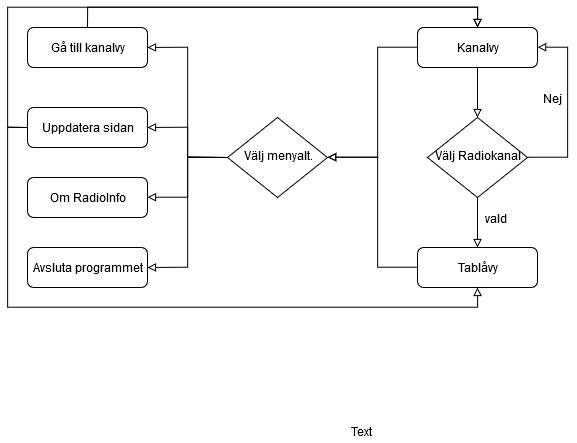
# Systembeskrivning

## Programbeskrivning

Denna applikation är ett program som tillhandahåller information och data om kanaler och programtablåer för radioprogram som sänds från Sverige Radio. Genom applikationen kan användaren välja en radiokanal från SR:s utbud och se aktuella programtablåer över de radioprogram som har sänds, sänds just nu och kommer att sändas inom ramen av ett dygn.

## Grafiskt användargränssnitt

Det grafiska användargränssnittet som användaren bemöter och använder sig av är väldigt avskalat och innehåller endast de mest väsentliga byggstenarna för att få applikationens flöde och funktionalitet att fungera på ett smidigt och intuitivt sätt.



**Figur 1:** *Flödesschema över applikationen.*

Applikationen har i stort sett endast två olika typer av vyer där informationen kommer att visas till användaren, se Figur 1. För övrigt finns det en meny högst upp i fönstret som ständigt är synlig och innehåller funktionalitet för att bland annat uppdatera aktuell tablå.

# Implementation

## Implementation av systembeskrivningen

Applikationen är uppbyggd genom programmeringsspråket Java och dess inbyggda grafiska bibliotek Swing.

Under arbetets gång så har applikationen skapats på en dator med Windows 10, men har även testkörts på en dator med det Linuxbaserade operativsystemet Debian för att säkerställa kompabilitet med datorer på Umeå Universitets labbsalar. Det finns därmed ingen garanti för att programmet även kommer att fungera felfritt på OSX eller andra operativsystem.

## Trådsäkerhet

Biblioteket Swing är i sig inte trådsäkert, men kan behöva användning av många separata trådar för att kunna hantera data som måste beräknas, samtidigt som användargränssnittet fortfarande måste vara interaktivt till användarinput. Av den anledningen är programmet uppdelat så att gränssnittet och uppdateringar av denna körs enbart på den så kallade EDT-tråden som Swing tillhandahåller, medan tunga uppdateringar så som att läsa in XML och visa upp den informationen görs i bakgrunden med SwingWorkers. Bla bla mer om trådar när jag lyckas med update

## Designmönster

### Adapter-mönster

Det allra första som implementerades under arbetets gång var en adapterklass vid namn XMLParser. Denna klass har till uppgift att läsa in extern XML och formatera om dess innehåll till ett format som är användbart och förståeligt för andra delar av systemet. Detta särskilt för att förenkla kompabilitet med Swing skapandet av JTables exempelvis.

### Proxy-mönster

När man endast vill gå från en kanaltablå tillbaka till kanalvyn så behöver inte det betyda att man måste uppdatera informationen för kanalerna då denna data sällan ändras. Därför sparas informationen om kanalerna kvar så länge inte en uppdatering sker, så att man slipper tillkalla den utomstående databasen för att återgå till kanalvyn.

### Observer/Observable-mönster

Inom MVC-mönstret och Swing då används actionlisteners som en form av observer på olika GUI-komponenter, eller observables. På så vis kan man prenumerera på interaktioner som användaren utför på respektive UI-komponent och utföra beräkningar utifrån dessa inom observer-metoderna.

### Template-mönster

Både för SwingWorkers samt uppbyggnad av JTables för tablåvyn så använts Interface och/eller abstrakt klass för att definiera ett skal av algoritmer som ska implementeras i underliggande klasser.

# Gränssnitt

## Kanalyvyn

Det första sida som syns då programmet startar är kanalvyn. Här finns det är en tre-delad lista över olika radiokanaler från Sveriges Radios utbud, se Figur 2. Från denna vy är funktionaliteten begränsad till att använda olika menyalternativ, eller att klicka med muspekaren på en av de listade radiokanalerna. Om den kanal man trycker på har en aktuell programtablå så kommer gränssnittet att skifta till tablåvyn, samtidigt som muspekaren byts ut till en loading-pekare för att indikera att systemet arbetar med den givna förfrågan. Om det inte finns en aktuell tablå så kommer en dialogruta att visas som informerar om detta.

En bild som visar skärmbild

Automatiskt genererad beskrivning

**Figur 2:** *Kanalvyn som visas när programmet först startas.*

## Tablåvyn

Den andra grundläggande vyn är tablåvyn som är nästkommande vy efter kanalvyn, se Figur 3. Från denna vy ser man högst upp på sidan en bild och en liten förklarande text tillhörande den kanal som man klickat på. Under detta finns programtablån som visar alla radioprogram som har sänds eller kommer sändas inom 12 timmar från den minuten som användaren klickar sig in på sidan.

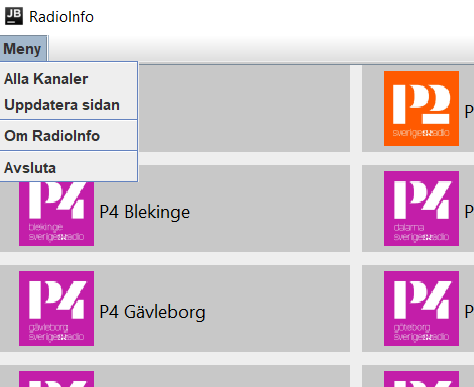
En bild som visar skärmbild

Automatiskt genererad beskrivning

**Figur 3:** *Tablåvyn som visar radioprogram som går har gått och som pågår just nu*

## Menyn

På samtliga sidor så finns det en meny, se Figur 4. Funkonaliteten är alltid densamma och inkluderar att gå tillbaka till kanalvyn, uppdatera sidan, få allmän information om applikationen samt att avsluta.



**Figur 4:** *Hur applikationens meny ser ut och var den är placerad i gränssnittet.*

# Lösningens begränsningar

När man uppdaterar applikationens information via menyalternativet ”Uppdatera sidan” eller när programmet automatiskt uppdaterar sig självt så uppstår det ibland ett litet problem där sidan har laddat färdigt, vilket indikeras av loading-muspekaren, men gränssnittet inte är direkt responsivt. Detta är väldigt märkligt med tanke på att den enda tråden som vid det tillfället körs är EDT-tråden då SwingWorkern som uppdaterade informationen just har avslutats efter att ha uppdaterat gränssnittet på ett trådsäkert och korrekt sätt. Detta sker inte vid varje uppdatering, men om det skulle inträffa och användaren klickar på en knapp inom kanalvyn så kommer det att dröja en liten men märkbar stund innan programtablån börjar läsas in. Detta problem har felsökts i stor omfattning, men i slutändan har ingen anledning eller konkret lösning lyckats implementeras. Därmed antas detta vara ett internt problem med Swing. Däremot så är det ingenting som förstår programmet eller dess funktionalitet i dess nuvarande form så det ses som ett godtyckligt fel att lämna kvar vid leverans.

En annan begränsning av lösningen är att den tidigare beskrivna ”Uppdatera sidan” funktionaliteten alltid kommer att ta användaren tillbaka till kanalvyn, även fastän användaren har en programtablå uppe. Detta valdes att implementeras på detta sätt på grund av tidsrestriktioner och är ett utvecklingsområde för applikationen. Detta anses dock inte vara ett större problem då det inte kommer vara ofta som en användare kommer att använda applikationen i en hel timme, vilket motsvarar uppdateringsintervallen.

# Vidareutveckling

För att göra denna applikation ännu bättre så finns det många områden som skulle behövas förändringar och utökningar för att kunna skapa en bättre och mer användarvänlig applikation. Dels så skulle man behöva utöka den cell-renderare som används för att ändra visuella delar av programtablåvyn. Detta för att vissa visuella hjälpmedel som färgmarkering vid tryck på en viss rad inte fungerar exempelvis. Detta är en konsekvens av att ”Avslutade”, ”Pågående” och ”Kommande” radioprogram färgkodas, samtidigt som inte ansågs nödvändigt att fixa tillbaka denna funktionalitet då andra funktioner behövde prioriteras.

För övrigt skulle man kunna utöka så att man kan använda API:er från kommersiella radiokanaler, om sådana finns, så att man skulle kunna visa upp programtabeller för flera radiokanaler och ha allting samlat på ett och samma ställe inom denna applikation.

Det skulle även vara något om man kunde streama akutell radio från internet baserad på akutell programtablå så att man även skulle kunna använda programmet för att direkt kunna lyssna på radio, istället för att bara kunna se de programmen som går.

Slutligen så skulle applikationen automatiska uppdatering samt ”Uppdatera sidan”-funktionaliteten via menyn behöva vidareutvecklas. Detta på grund av att användaren i dagsläget alltid blir tagen till kanalvyn, eller startsidan, varje gång som informationen uppdateras på detta sätt. Det hade varit fördelaktigt ur användarsynpunkt om den aktuella sidan uppdaterades samtidigt som man inte blir förflyttad från sidan man för tillfället har framme.