

Christer Jakobsson
dv12cjn
2014-01-09

Umeå Universitet
Applikationsprogrammering I Java
VT-13

Uppgift 2: TravelInfo

dv12cjn/edu/apjava/lab2/

Johan Eliasson
Jan-Erik Moström
Eric Jönsson

Innehållsförteckning

Programbeskrivning.....	3
Åtkomst och användarhandledning.....	3
Kompilering och körning.....	3
Grafiskt Gränssnitt.....	4
Sökning	5
Uppdateringsintervall	6
Manuell Uppdatering.....	6
Systembeskrivning.....	7
UML.....	7
TravelInfoController.....	8
LinkListener.....	8
TravelInfoGui.....	8
TravelInfo.....	9
TravelInfoConstants.....	9
TravelModel.....	9
TravelOffer.....	9
Begränsningar.....	9
Testkörningar.....	10
Trådsäkerhet.....	13
Designmönster och diskussion.....	13
Model, Controller, View (MVC).....	13
Diskussion.....	13
Bilagor.....	13

Programbeskrivning

Programmet TravelInfo är utformat efter en uppgiftspecifikation utifrån kursen ”Applikationsprogrammering i Java”, programmets uppgift är att läsa data ifrån en xml fil som fritidsresor tillhandahåller vilket innehåller information om resor som fritidsresor har. Detta ska visas i ett grafiskt gränssnitt och användaren kan klicka på programmet för att få veta mera om resan eller klicka sig vidare och komma till bokningssidan. Programmet har en inställningsbar uppdateringsintervall, vilket kan ställas in i 30 minuters intervaller. Programmet sparar information mellan körningar om vad användaren har satt uppdateringsintervallet och storleken på fönstren, detta för att användaren ska kunna välja själv vilken fönsterstorlek som passar bäst till sig själv.

Programmet består av tre stycken paket, TravelInfoController, TravelInfoGui och TravelInfoModel. TravelInfoController tar hand om att skapa alla de andra objekten och kontrollerar vad som ska hända när användaren trycker på en knapp i Gränssnittet. TravelInfoGui har hand om all grafisk presentation och funktionalitet. TravelModel har hand om att läsa in datat ifrån hemsidan och skapar en AbstractTableModel med datat.

Åtkomst och användarhandledning

Kompilering och körning

Koden till programmet ligger i dv12cjn/edu/apjava/lab2/src
En kompilerad version ligger i dv12cjn/edu/apjava/lab2/bin
Rapporten finns i pdf format i dv12cjn/edu/apjava/lab2/report

Programmet startas genom att köra kommandot ”java Main”.

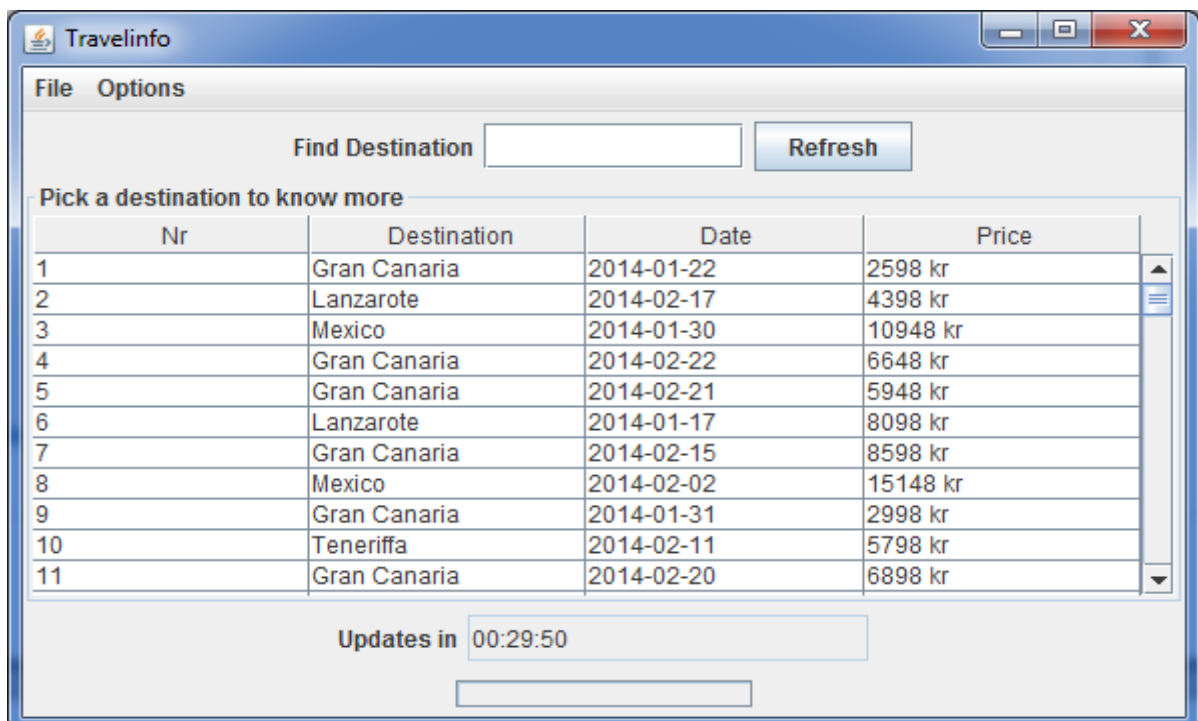
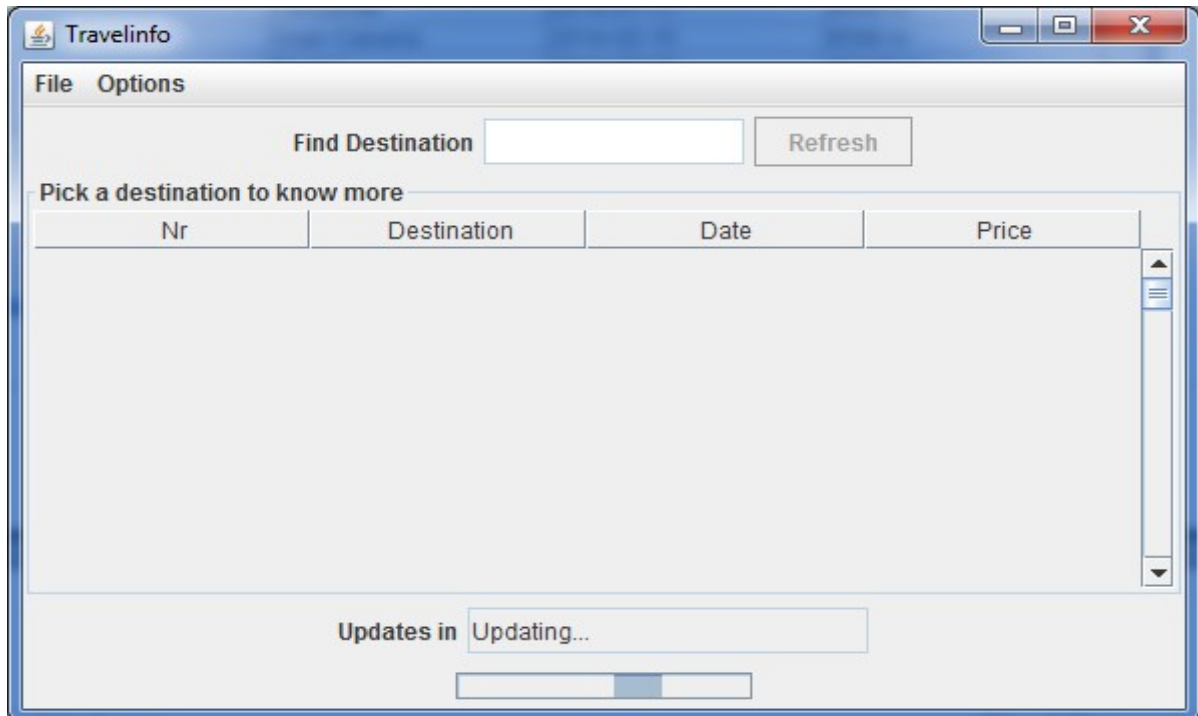
De filer som kommer att finnas tillgängliga är

- Main, Används för att skapa en TravelController som sedan exekverar resten av programmet.
- TravelInfoController Package
 - TravelInfoController
- TravelInfoGui Package
 - LinkListener
 - TravelInfoGui
- TravelInfoModel Package
 - TravelInfo
 - TravelInfoConstants
 - TravelModel
 - TravelOffer

Det som programmet behöver för att fungera är en internet uppkoppling så att det kan hämta hem xml filen ifrån hemsidan och sedan visa resultatet.

Grafiskt Gränssnitt

När programmet startar så kommer ett gränssnitt att visas som ser ut som nedan.

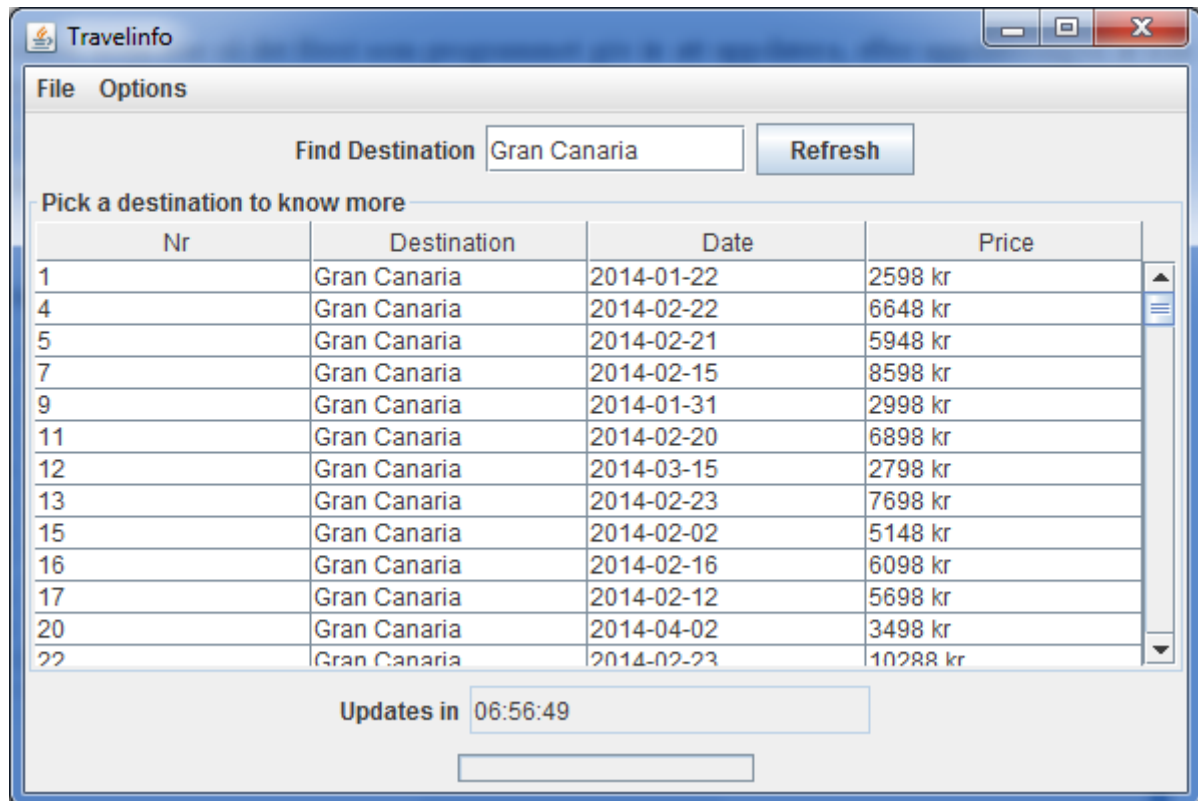


Som bilden indikerar så uppdaterar programmet det första det gör, efter uppdateringen är klar så visas en tabell med alla resor som finns.

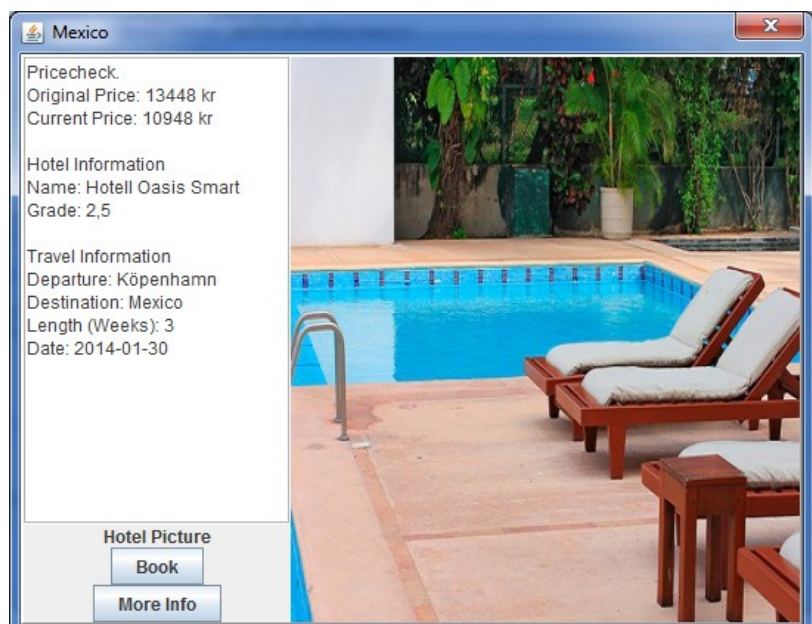
Här visas en tabell med alla resor som man kan skrolla igenom för att bläddra bland resorna.

Sökning

I fältet "Find Destination" så kan man mata in en textrad som man vill söka efter och bara visa resultat som innehåller den exakta textsträngen. Genom att använda "|" operatoren så kan man lägga till fler sökord text Gran|Thai kommer att visa alla destinationer som innehåller de båda strängarna.

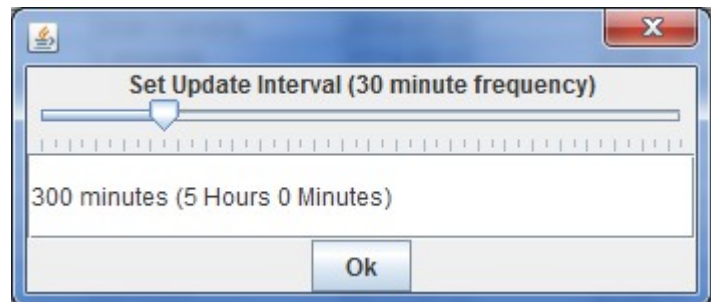


Mera information om ett resmål kan man få genom att klicka på det med musen, då kommer ett fönster med informationen att visas. I detta fönster finns det 2 knappar, en för att komma till bokningssidan och en för att öppna en sida som innehåller ytterligare information. Om någon av dessa blir klickade på så kommer programmet kolla om det kan öppna sidan i användarens standard webbläsare ex Chrome. Om detta inte stöds så kommer programmet att öppna sidan i ett fönster som agerar som en minimal webbläsare.

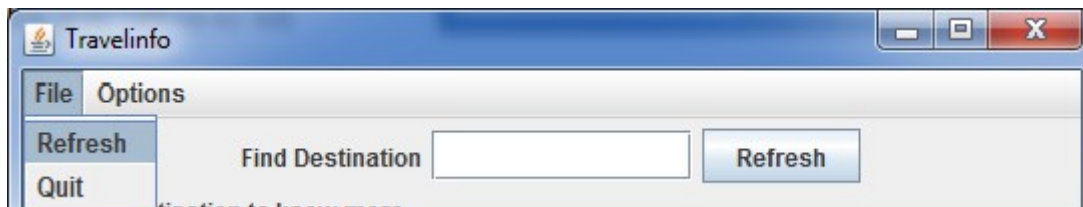


Uppdateringsintervall

Genom att i huvudfönstret klicka på Options → Set Update Interval så ställer man in uppdateringsintervallet med 30 minuters intervaller genom att reglera en slider till önskvärd uppdateringstid. Genom att klicka på "Ok" knappen eller krysset så har man valt en ny tid.

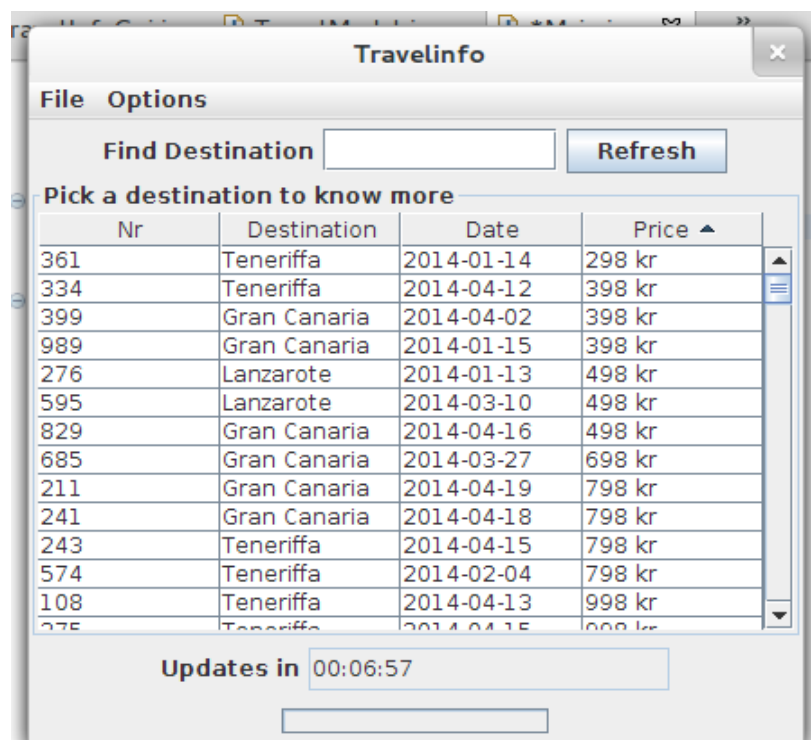


Manuell Uppdatering



Genom att antingen klicka på "Refresh" knappen på huvudfönstret eller genom att välja File → Refresh så kommer programmet manuellt att uppdateras och visa en uppdaterad tabell.

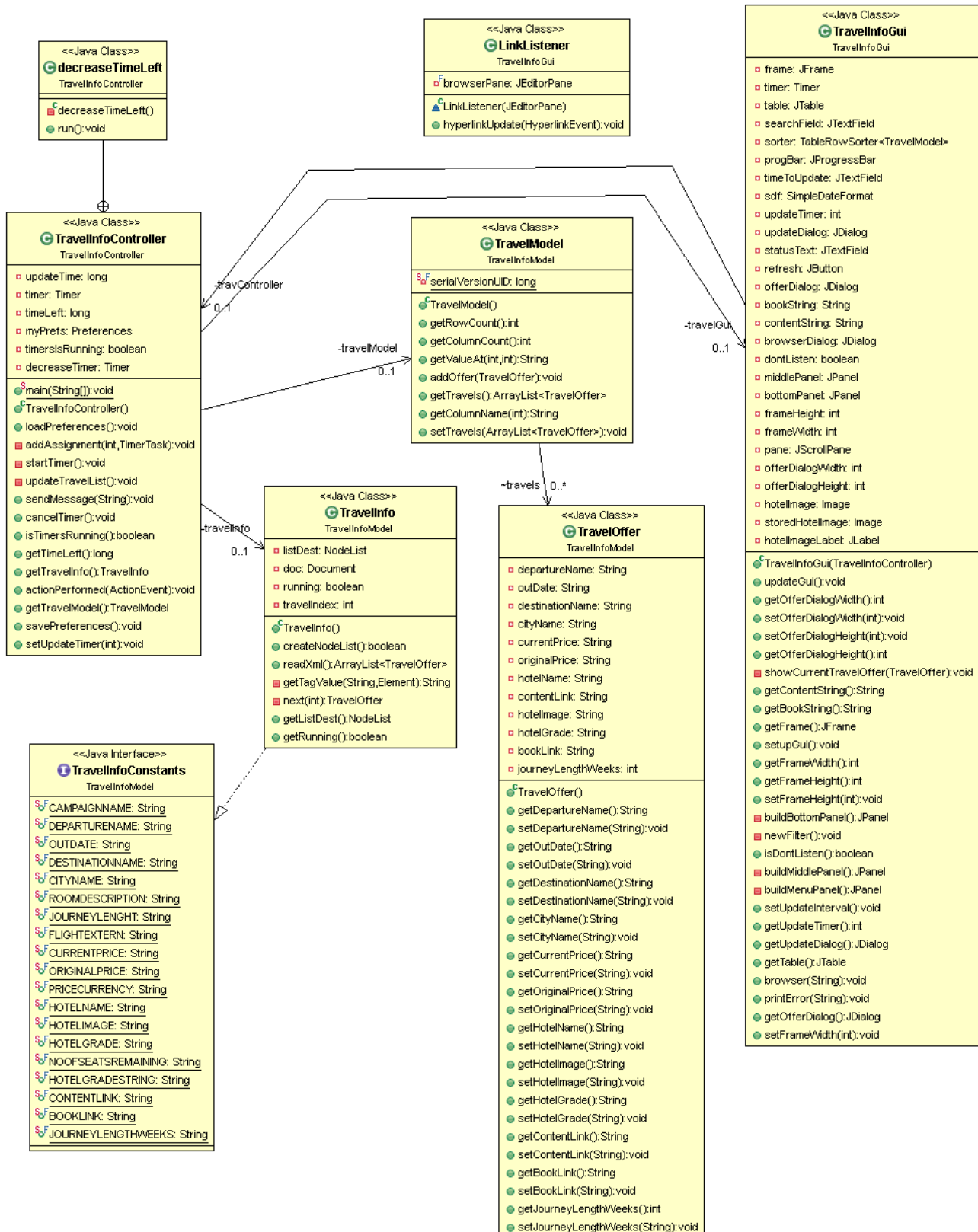
Användaren kan sortera tabellen efter valfri kolumn. Exempel Sorterat efter pris:



När användaren vill avsluta programmet så sparas inställningarna och programmet stängs ner.

Systembeskrivning

UML



TravelInfoController

Denna klass skapar de objekt som behövs för att kunna hämta data ifrån webbsidan och visa det i ett grafiskt gränssnitt. Klassen implementerar ActionListener och skapar gränssnittet som den lyssnar efter knapptryckningar ifrån. Innehåller två metoder med två TimerTasks varav den ena tar hand om att läsa in data från webbsidan med det specificerade uppdateringsintervallet, och den andra TimerTasken räknar ner en long variabel med en sekund per sekund, variabeln representerar tiden kvar till nästa uppdatering. Så detta sker därmed i två egna trådar så att TravelInfoGui och TravelInfoControllers tråd som dom använder inte låses medans man uppdaterar tabellen.

När användaren stänger av programmet på ett korrekt sätt så kommer vissa inställningar sparas i registret med hjälp av Preferences, och när programmet startas så försöker det hämta de sparade inställningarna och om det inte skulle gå så sätts de till standardvärden.

LinkListener

Denna klass implementerar HyperLinkListener och används när användaren klickar på en länk i den egenskapade minimalistiska webbläsaren vilken använder sig av en JEditorPane, när användaren trycker på en länk i JEditorPanen så anropas dennes setPage metod med HyperLinkListeners events medföljande URL. Om Sidan inte hittas så visas "Error 404" i webbläsaren.

TravelInfoGui

Denna klass skapar ett grafiskt gränssnitt som används för att visa en tabell med alla resor och låta användaren se mera information om en specifik resa och ändra uppdateringsintervallet. Klassen skapar en JFrame som innehåller tre paneler, övre, mittre, och nedre del av framen, den övre innehåller framens meny där användaren har olika operationer denne kan göra.

Den mittersta panelen innehåller en JLabel med förklarande text till vad användaren ska lägga in i JTextField:en som ligger i samma panel och till höger om JLabel:n. Detta textfält används för att filtera tabellen med den aktuella strängen som finns i fältet, om fältet är tomt så visas alla resor. Det finns även en knapp som användaren kan trycka på för att manuellt uppdatera tabellen, och själva tabellen ligger också i den mittersta panelen. Tabellen går att sortera efter valfri kolumn genom att klicka på kolumn namnet.

Den nedre panelen innehåller en JLabel med förklarande text till ett textfält som visar en timer som räknar ner tiden till en uppdatering kommer att ske. Panelen innehåller även en indeterminate JProgressBar som körs när programmet uppdaterar för att visa det för användaren.

Klassen innehåller en metod som innehåller en Timer med en TimerTask som körs varje sekund, om programmet uppdaterar tabellen så visas det genom att progressbaren rör sig, textfältet till uppdateringen visar "Updating..." och muspekaren ändras till "WAIT_CURSOR".

Tabellen inaktiveras av säkerhetsskäl och den döljs även, även "Search" och "Refresh" knappen inaktiveras.

När programmet uppdaterat klart så aktiveras alla dessa funktioner igen och progress baren slutar att röra sig och textfältet med tiden till nästa uppdatering visar den aktuella tiden kvar.

Om användaren trycker på en cell i tabellen så kommer det att visas ett fönster som har mera information om den markerade resan, det finns även en knapp för att komma till bokningssidan och en knapp för att komma till en sida som innehåller ännu mera information om resan.

TravelInfo

Denna klass läser in data ifrån fritidsresors hemsida i forma av ett xml dokument som innehåller information om 1000 stycken resor som fritidsresor vill sälja. Klassen läser in datat igenom att skapa ett dokument som innehåller indata och sedan skapas en NodeList och läser av varje "Offer" som då är ett resmål. Klassen hämtar sedan den information den behöver ifrån noden i en och sparar det i en TravelOffer som sedan läggs till i en ArrayList. Detta sker för varje offer som finns i dokumentet och därefter så returneras ArrayListen som sedan TravelModel kommer att använda sig av för att skapa tabellen.

TravelInfoConstants

En klass som innehåller konstanter för att läsa in ifrån XML dokumentet, konstanterna är vad varje tag har för namn och de används när det plockas ut element ifrån xml dokumentet.

TravelModel

Denna klass implementerar en AbstractTableModel, klassen innehåller en ArrayList med alla resors information lagrade i en TravelOffer klass. Tabellen som skapas i Gränssnittet (JTable) får denna model som inparameter och skapar sedan en tabell utifrån de implementerade metoderna i TravelModel.

TravelOffer

Denna klass är en hjälpklass, dess syfte är att lagra information om en enskild resas information tex destination, pris, vilket hotell som ingår. Denna klass innehåller bara metoder med variabler och en getter och setter till varje variable så att det går att sätta värdet och hämta värdet.

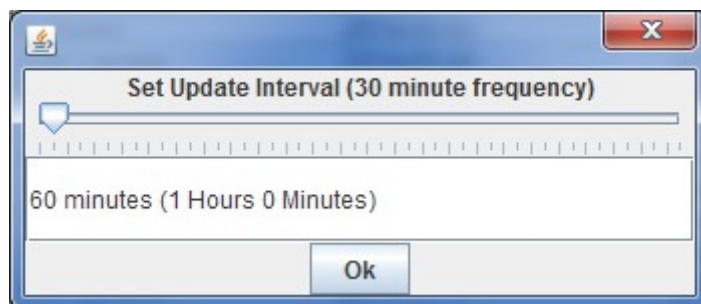
Begränsningar

- TravelInfo använder sig inte av validering när det läser in datat, detta för att ett schema inte tillhandahålls.
- Användaren har bara möjlighet att välja uppdateringsintervall upp till tjugofyra timmar, Detta är en medveten begränsning för att jag upplever inte att det behövs kunna vara högre.
- Om användaren har sträng tillåtelse för program att arbeta med systemets register så kommer Preferences inte kunna spara ner inställningarna mest troligt.

Testkörningar

Test 1: Kontrollerar så att tiden som visas för nästa uppdatering stämmer efter att uppdateringsintervallet ändrats.

Utskrift:



Ställer in tiden till 60 minuter och trycker på ok.

Programmer uppdaterar igen och visar sedan tiden:

11	Gran Canaria	2014-01-29	1998 kr
12	Gran Canaria	2014-02-20	6898 kr
13	Gran Canaria	2014-03-15	2798 kr

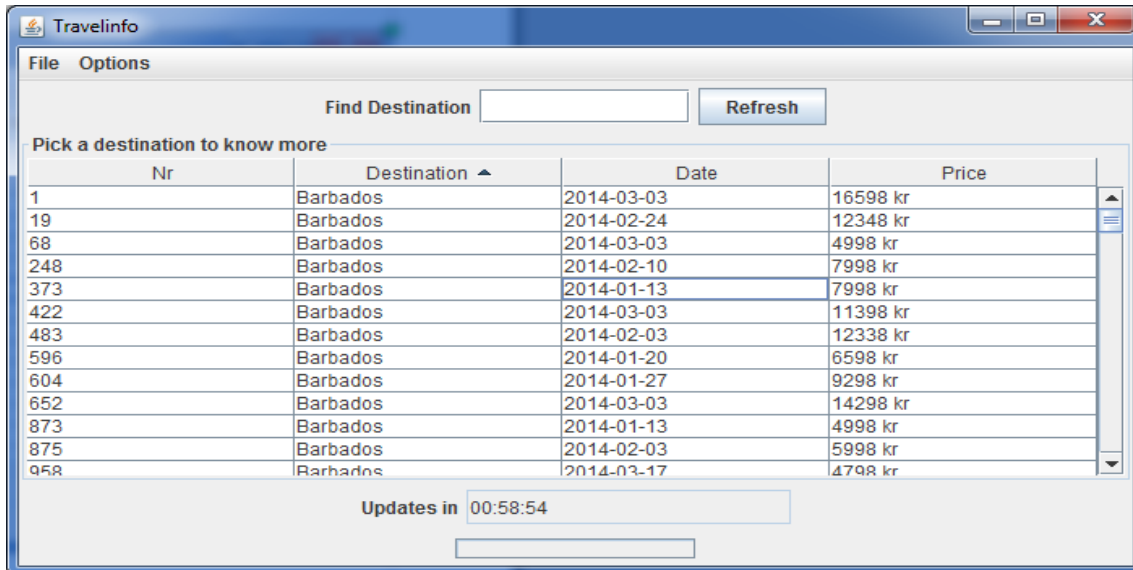
Updates in 00:59:54

Kommentar: Programmet fungerar som tänkt.

Test 2: Testar så att det fungerar att sortera listan och sedan uppdatera den.

Utskrift:

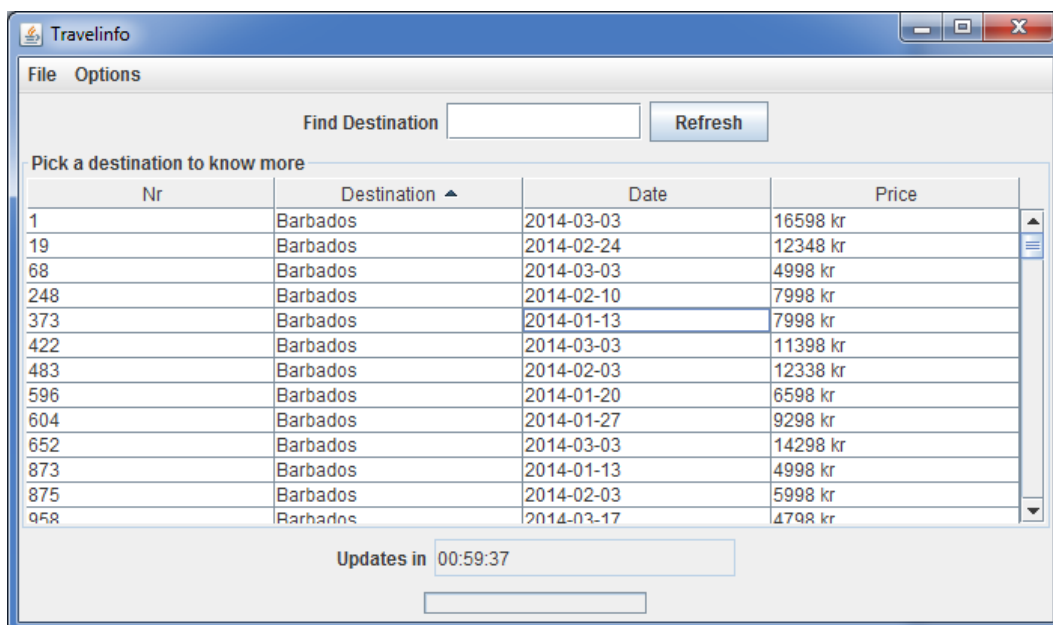
Valt att sortera efter Destination.



The screenshot shows the 'Travelinfo' application window. At the top, there is a 'Find Destination' text box and a 'Refresh' button. Below this is a section titled 'Pick a destination to know more'. The main part of the window is a table with four columns: 'Nr', 'Destination', 'Date', and 'Price'. The table contains 15 rows of data, all with 'Barbados' as the destination. The 'Nr' column lists various numbers, and the 'Price' column shows prices in Swedish Krona (kr). At the bottom of the window, there is a status bar that says 'Updates in 00:58:54'.

Nr	Destination	Date	Price
1	Barbados	2014-03-03	16598 kr
19	Barbados	2014-02-24	12348 kr
68	Barbados	2014-03-03	4998 kr
248	Barbados	2014-02-10	7998 kr
373	Barbados	2014-01-13	7998 kr
422	Barbados	2014-03-03	11398 kr
483	Barbados	2014-02-03	12338 kr
596	Barbados	2014-01-20	6598 kr
604	Barbados	2014-01-27	9298 kr
652	Barbados	2014-03-03	14298 kr
873	Barbados	2014-01-13	4998 kr
875	Barbados	2014-02-03	5998 kr
958	Barbados	2014-03-17	4798 kr

Trycker sedan på uppdatera knappen och efter att programmet uppdaterat ser det ut sm följer:



This screenshot shows the 'Travelinfo' application window after an update. The interface is identical to the previous screenshot, but the status bar at the bottom now shows 'Updates in 00:59:37', indicating that time has passed since the last update.

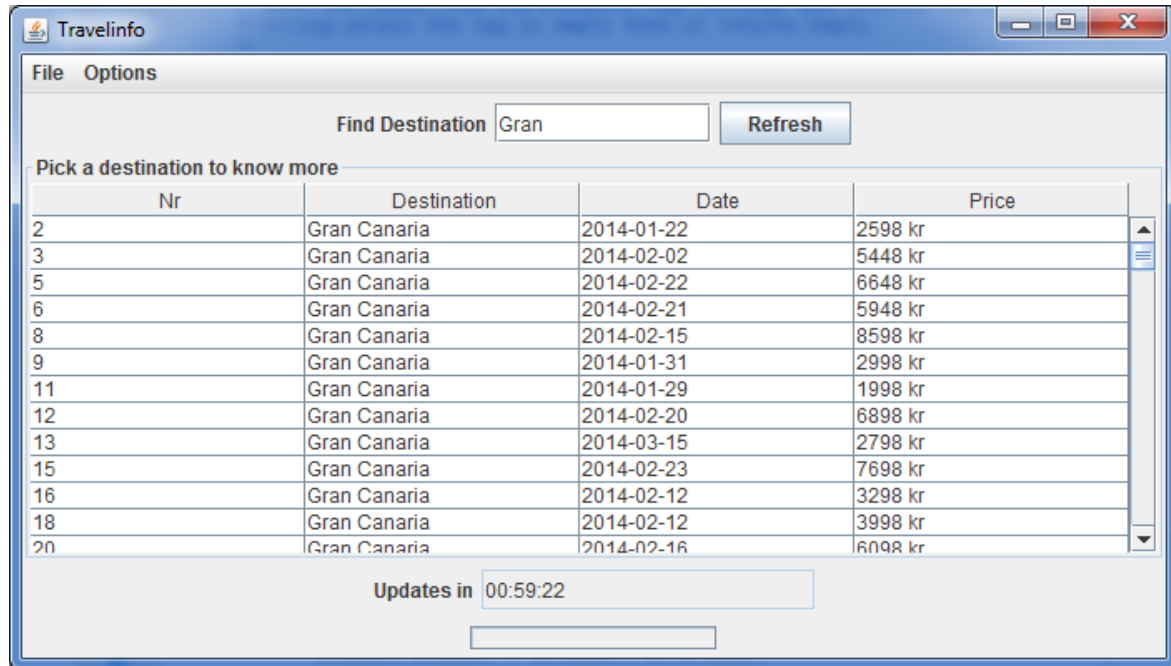
Nr	Destination	Date	Price
1	Barbados	2014-03-03	16598 kr
19	Barbados	2014-02-24	12348 kr
68	Barbados	2014-03-03	4998 kr
248	Barbados	2014-02-10	7998 kr
373	Barbados	2014-01-13	7998 kr
422	Barbados	2014-03-03	11398 kr
483	Barbados	2014-02-03	12338 kr
596	Barbados	2014-01-20	6598 kr
604	Barbados	2014-01-27	9298 kr
652	Barbados	2014-03-03	14298 kr
873	Barbados	2014-01-13	4998 kr
875	Barbados	2014-02-03	5998 kr
958	Barbados	2014-03-17	4798 kr

Kommentar: Fungerar som tänkt, det är sorterat efter samma kolumn efter en uppdatering är gjord.

Test 3: Filtrering fungerar även efter att listan har uppdaterats.

Utskrift:

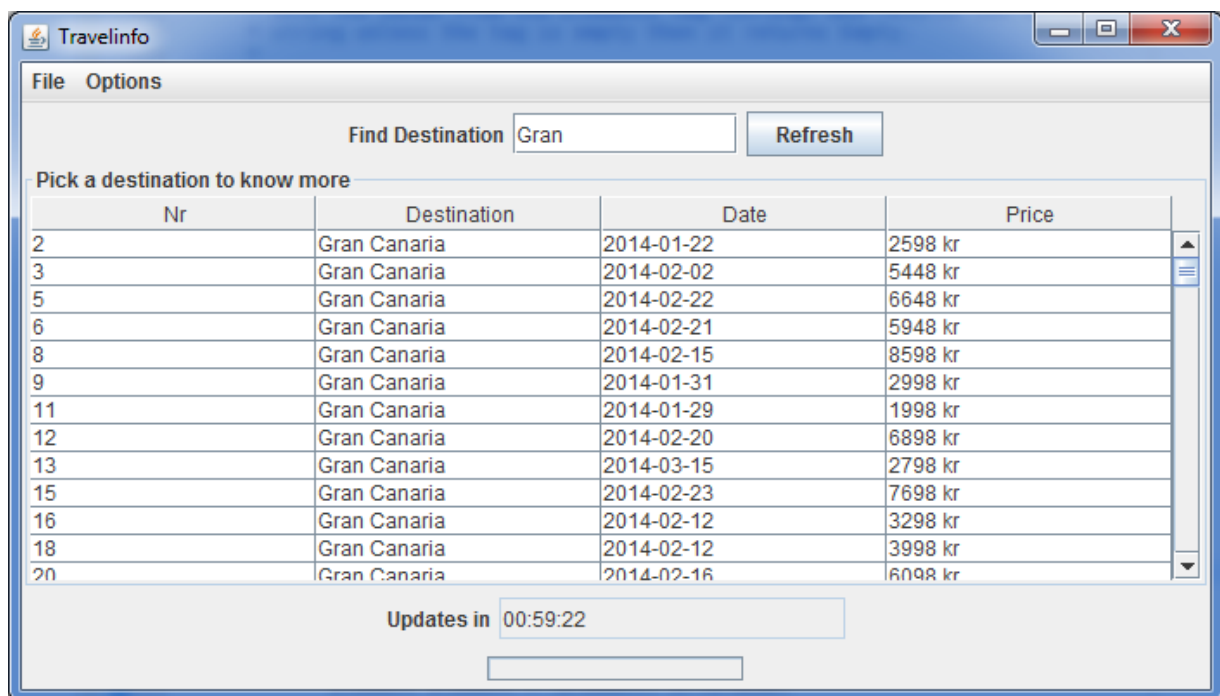
Skriver in "Gran" och ser då bara de Resmål som innehåller den exakta textsträngen.



The screenshot shows the Travelinfo application window. At the top, there is a menu bar with 'File' and 'Options'. Below the menu bar, there is a search section with a label 'Find Destination', a text input field containing 'Gran', and a 'Refresh' button. Below this, there is a section titled 'Pick a destination to know more' which contains a table. The table has four columns: 'Nr', 'Destination', 'Date', and 'Price'. The table lists 18 destinations, all of which are 'Gran Canaria'. The prices range from 1998 kr to 6648 kr. At the bottom of the window, there is a status bar that says 'Updates in 00:59:22'.

Nr	Destination	Date	Price
2	Gran Canaria	2014-01-22	2598 kr
3	Gran Canaria	2014-02-02	5448 kr
5	Gran Canaria	2014-02-22	6648 kr
6	Gran Canaria	2014-02-21	5948 kr
8	Gran Canaria	2014-02-15	8598 kr
9	Gran Canaria	2014-01-31	2998 kr
11	Gran Canaria	2014-01-29	1998 kr
12	Gran Canaria	2014-02-20	6898 kr
13	Gran Canaria	2014-03-15	2798 kr
15	Gran Canaria	2014-02-23	7698 kr
16	Gran Canaria	2014-02-12	3298 kr
18	Gran Canaria	2014-02-12	3998 kr
20	Gran Canaria	2014-02-16	6098 kr

Uppdaterar.....



This screenshot is identical to the one above, showing the Travelinfo application window with the search field set to 'Gran' and a list of 18 destinations. The status bar at the bottom still shows 'Updates in 00:59:22'.

Nr	Destination	Date	Price
2	Gran Canaria	2014-01-22	2598 kr
3	Gran Canaria	2014-02-02	5448 kr
5	Gran Canaria	2014-02-22	6648 kr
6	Gran Canaria	2014-02-21	5948 kr
8	Gran Canaria	2014-02-15	8598 kr
9	Gran Canaria	2014-01-31	2998 kr
11	Gran Canaria	2014-01-29	1998 kr
12	Gran Canaria	2014-02-20	6898 kr
13	Gran Canaria	2014-03-15	2798 kr
15	Gran Canaria	2014-02-23	7698 kr
16	Gran Canaria	2014-02-12	3298 kr
18	Gran Canaria	2014-02-12	3998 kr
20	Gran Canaria	2014-02-16	6098 kr

Kommentar: Fungerar som tänkt, det är filtrerat efter samma sak efter en uppdatering.

Trådsäkerhet

Det är ett krav på att programmet ska vara trådsäkert, detta försöker jag åstadkomma med olika tillvägagångssätt.

Jag har gjort min TravelModel till volatile för ta bort trådproblem om en tråd ändrar värdet och en annan vill hämta värdet och detta sker samtidigt. Det enda tillfället TravelModel ändras på är när programmet ska uppdateras och då så tas allt i ArrayListen bort sen läggs det till nya TravelOffers ifråntråden som TimerTasken kör som i sin tur kör metoden readXml. När programmet uppdaterar ArrayListen så inaktiverar jag alla komponenter som kan hämta listan så det ska ej gå att kunna manipulera listan medans modellen uppdateras, om gränssnittets tråd skulle bli långsam skulle användaren rent tekniskt kunna trycka på "Refresh" knappen innan den inaktiveras men jag har försökt att åstadkomma detta men ej lyckats vara så snabb. Gränssnittet hämtar en boolean ifrån TravelInfo varje sekund som säger om TravelModel uppdateras eller inte, denna boolean är volatile för att TravelInfo ändrar dess värde och gränssnittet hämtar den och detta kan ske samtidigt.

Designmönster och diskussion

Model, Controller, View (MVC)

För klassernas relation till varandra så försöker jag att använda mig av ett MVC designmönster där TravelInfoController är Controller, TravelModel är modellen och TravelInfoGui är View.

Diskussion

Jag tycker att denna laboration har varit intressant, kul att arbeta med fritidsresors riktiga data och bygga en program som man kan koppla till verkligheten. Trådsäkerheten har varit klurig att förstå vid vissa tillfällen och att använda datatypen Jtable har varit spännande för den har varit lite mera avancerad att lära sig hur den fungerar. Det svåraste med denna uppgift har varit det faktum att den bedöms med poäng detta gör det svårt att känna att man är klar med uppgiften, i och med att det finns ett grafiskt gränssnitt så är det många funktioner man skulle vilja lägga till och många olika sätt att presentera informationen på som kan vara bättre än det man har gjort men man måste bestämma sig att man har finjusterat klart någon gång.

Bilagor

Bilaga 1:

Laborationsspecifikation: <http://www8.cs.umu.se/kurser/SDV135/HT13/labbar/lab2/index.html>