D0036D Nätverksprogrammering

Laboration 2 – Väderdataklient

Uppgift

I denna labb ska du träna på GUI programmering i Java, hämta data via ett REST API samt att ”parsa” ut data ur xml filer.
Du ska skapa en väderklient som kommunicerar med YR (yr.no). Din applikation ska hämta och visa, prognostiserad, temperatur för en viss ort och timme. Man ska kunna mata in ort och timme eller välja från lista. Applikationen skall visa Temperatur för användarens val eller felmeddelande om något gått fel.
För att begränsa belastningen på servern rekommenderas det att man ”cachar” väderdata och bara begär ny data från servern när den gamla blivit för gammal. Din applikation skall göra detta. Se till att det går att ställa in ”åldringstiden” utan att programmet måste kompileras om.

YR erbjuder data via ett API. Du hittar dokumentation om detta API på:
http://api.yr.no/weatherapi/locationforecast/1.9/documentation
Detta API är baserat på HTTP meddelandet GET. Man skickar en förfrågan om en viss URL och får tillbaka ett meddelande kodat i xml.

I filen ”places.xml” placeras koordinaterna för de orter man vill kunna begära data om. Några orter ligger redan där men det är också ok att lägga till egna orter.

För läsning och parsning av xml rekomenderas
http://docs.oracle.com/javase/7/docs/api/org/w3c/dom/package-summary.html och
http://docs.oracle.com/javase/7/docs/api/javax/xml/parsers/package-summary.html.

Du ska designa din applikation enligt Model-View-Controler design pattern.
(läs mer på http://en.wikipedia.org/wiki/Model-view-controller)
”Model” hanterar själva datat (tex. i listor).
”View” visar upp delar av datat i ett lämpligt format för användaren.
”Controler” sköter hanteringen av user input och binder ihop View och Model.
Om man inte använder denna modell så blir det lätt så att de visuella komponenterna även ansvarar för ”lagringen” av datat…
Du är såklart välkommen att bygga in fler finesser i din applikation, t.ex. att andra väderparametrar visas, ju mer generell din design och implementation är desto lättare skall det vara att göra detta.

Programmet skall bygga på en objektorienterad design och skall också ha en objektorienterad implementation.

Din kod ska vara kommenterad och koddokumentation genererad med javadoc.
http://jautodoc.sourceforge.net/,
http://www.oracle.com/technetwork/java/javase/documentation/javadoc-137458.html

Begränsningar

Programmet behöver bara kunna hantera en 24 timmars period.
Programmet behöver bara kunna hantera de orter som finns i filen ”places.xml”.
Noggrannhet för ”åldringstid” vid caching behöver inte vara noggrannare än en minut.

Examination

Redovisa din lösning för laborationshandledare vid anvisat tillfälle. Lämna in kod (.txt), Dokumentation och klassdiagram (.zip).