



AKADEMIN FÖR TEKNIK OCH MILJÖ
Avdelningen för industriell utveckling, IT och samhällsbyggnad

Programvaruteknik - utveckling och underhåll av programvara

**15 HP
VT16**

Inlämningsuppgift 2

av

Viktor Hanstorp

Akademien för teknik och miljö

Högskolan i Gävle

S-801 76 Gävle, Sweden

Innehållsförteckning

Inledning	1
Metod	1
Resultat	2
Diskussion	2
Referenser	3
Bilaga 1 – Utskrift vid körning av ”FootballAndWeatherCombiner”	6

Inledning

Inlämningsuppgiften är en fortsättning på ett arbete med att kombinera data från olika data källor. I denna del så skall man hämta data via **REST** [1] från olika websidor.

De sidor som man skall använda är **Everysport** [2] och **SMHI** [3], där det finns ganska bra dokumentation för **everysport** [4], dock så är det inte lika bra för **SMHI**.

Målet är att kombinera antal gjorda mål med den temperatur som var.

Metod

För att generera datakällor från de olika websidorna så valdes det att skapa "**Builders**" [5] med en blandning av "**Factory Methods**" [6].

Basen för dessa datakällgenererare är den abstrakta klassen

"**DataSourceBuilder**" [7]. Denna klass har en abstrakt metod som heter "**generateData**", vilken skall ge den data som skall stoppas in i den "**DataSource**" som byggs. Det finns också funktionalitet att sätta "**name**", "**unit**" samt en filtreringsfunktion. Den publika metoden "**build**" samlar all data från "**name**", "**unit**" och "**generateData**" efter filtrering med filtreringsfunktionen och skapar en "DataSource" som den returnerar.

"**DataSourceBuilder**" ärvs sedan av "**JSONDataSourceBuilder**" [8] och "**CSVDataSourceBuilder**" [9]. Dessa klasser implementerar metoden "**generateData**" vilken genererar data från den datatyp den är skapad för att hantera. Klasserna kan ges funktioner som extraherar data ur de datastrukturer som de är skapta för att hantera. Exempel på sådana funktioner är funktioner som extraherar "**name**" och "**unit**" ur den data man ger dem. Båda klasserna skall ges en "**Supplier<String>**" [10] som tillhandahåller den data som skall tolkas.

För att underlätta detta så finns klassen "**DataSupplierFactory**" [11], vilken tillhandahåller metoder för att generera "**Supplier<String>**" som kan läsa filter eller data från en **URL** [12]. Metoden som läser från **URL** cachar [13] resultatet som standard (man kan säga till att den inte skall göra det) då det inte är så bra att skicka så mycket trafik, samt för att göra det snabbare.

JSON- [14] och **CSV** [15] byggarna är skapta så att de kan läsa vilken **JSON** och **CSV** data som helst, under förutsättning att man ger dem korrekta extraktionsfunktioner. För att göra byggarna snabbare så cachas data så att när man ger nya extraktionsfunktioner så uppdateras bara den data som behöver uppdateras.

JSON byggaren utnyttjar biblioteket "**Genson**" [16] för att översätta en **JSON** textfil till en "**Map**" [17], som inspirerat av tillhandahållen kod.

JSON byggaren och **CSV** byggaren ärvs av "**FootballSourceBuilder**" [18] respektive "**SMHISourceBuilder**" [19].

"**FootballSourceBuilder**" definierar standardextraktionsfunktioner som är anpassade för att tolka data från den API [20] som **everysport** tillhandahåller. För att läsa värden från datan så kan de statiska funktionerna "**HOME_GOALS_EXTRACTOR**", "**AWAY_GOALS_EXTRACTOR**" eller "**SPECTATORS_EXTRACTOR**" användas. Man kan också välja om man vill hämta data med den

tillhandahållna **API**-nyckeln, en egen nyckel eller en valfri **"Supplier"**. Om man väljer att hämta data med hjälp av **API**:et så kan man även välja vilka parametrar man skall använda.

"SMHISourceBuilder" hämtar data från **SMHI**s **API**. Man kan välja vilken station man vill hämta data ifrån med hjälp av enumet **"SMHILocation"** [21], där alla stationer som **SMHI** erbjuder finns med. Stationen matas till byggaren tillsammans med den period man är intresserad av (korrekterad arkiv, senaste fyra månaderna eller de senaste dagarna).

Funktionalitet för att läsa regn istället för temperatur är implementerad, dock så är det inte testat.

För att kombinera data (antal mål med temperatur) så skapades klassen **"FootballAndWeatherCombiner"** [22] vilken tar emot en mapping för (arena-id : **SMHI** station) och bygger ihop data-par med (totalt antal mål : temperatur).

Tanken med uppgiften var att man skulle köra **TDD** [23], dock så lyckades inte det så bra. Det kan ha att göra med att vissa delar var redan påbörjade när **TDD** introducerades, samt att resultatet som skulle genereras inte var riktigt fastställt.

Resultat

Det resulterande programmet fungerar, enligt testerna och iakttagelser av körningar.

Det som efterfrågas i uppgiften (antal mål vid temperatur) har genererats (se Bilaga 1 – Utskrift vid körning av "FootballAndWeatherCombiner").

Diskussion

Användningen av **SMHI**s **API** blev ganska bökigt. Som det ser ut nu så inaktiverades möjligheten att använda annan data än det från det korrigerade arkivet, dock så finns den mesta funktionalitet som behövs redan i koden. Annat som är bökigt är att de olika **SMHI** stationerna har olika längd på insamlad data. Detta gör det svårt att mappa mot annan data. På grund av tidsbrist så kunde detta inte lösas, dock så bör det i framtiden åtgärdas med funktioner som kan hitta den närmsta station till en position (long lat) under vissa datum, och eventuellt också kombinera mätdata från olika stationer om det behövs. Det kan vara så att denna funktionalitet faktiskt finns i **API**n, dock så har det ej hittats.

Andra datakällor av **JSON** och **CSV** stuk bör vara ganska enkla att implementera, då det finns en ganska bra grund. Andra dataformat bör också vara relativt enkla att införa.

Är missnöjd med att det inte riktigt lyckades med att köra **TDD**, men det lyckades åtminstone få med test.

Referenser

- [1] Wikipedia, "Representational state transfer," 25 2 2016. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Representational_state_transfer&oldid=706807616. [Använd 27 2 2016].
- [2] EverySPORT, "EverySPORT," EverySPORT, [Online]. Available: <http://everySPORT.com>. [Använd 27 2 2016].
- [3] SMHI, "Väder Väderprognoser Klimat- & Vädertjänster i Sverige | SMHI," SMHI, [Online]. Available: <http://www.smhi.se>. [Använd 28 2 2016].
- [4] menmo, "Documentation for the EverySPORT AP," [Online]. Available: <https://github.com/menmo/everySPORT-api-documentation>. [Använd 27 2 2016].
- [5] SourceMaking, "Builder Design Pattern," SourceMaking, [Online]. Available: https://sourcemaking.com/design_patterns/builder. [Använd 28 2 2016].
- [6] SourceMaking, "Factory Method Design Pattern," SourceMaking, [Online]. Available: https://sourcemaking.com/design_patterns/factory_method. [Använd 28 2 2016].
- [7] V. Hanstorp, "GitHub - DataSourceBuilder," 27 2 2016. [Online]. Available: https://github.com/HIG-Student/Programvaruteknik_Lab2/blob/6f7739c9e4656f63f4026ec420cbc41dbe08f0c5/Programvaruteknik_Lab2/src/se/hig/programvaruteknik/model/DataSourceBuilder.java. [Använd 28 2 2016].
- [8] V. Hanstorp, "GitHub - JSONDataSourceBuilder," 26 2 2016. [Online]. Available: https://github.com/HIG-Student/Programvaruteknik_Lab2/blob/91c849fed6bbcd754d38d3a7257ab68e1d5ab7e8/Programvaruteknik_Lab2/src/se/hig/programvaruteknik/data/JSONDataSourceBuilder.java. [Använd 28 2 2016].
- [9] V. Hanstorp, "GitHub - CSVDataSourceBuilder," 26 2 2016. [Online]. Available: https://github.com/HIG-Student/Programvaruteknik_Lab2/blob/91c849fed6bbcd754d38d3a7257ab68e1d5ab7e8/Programvaruteknik_Lab2/src/se/hig/programvaruteknik/data/CSVDataSourceBuilder.java. [Använd 28 2 2016].
- [10] Oracle, "Supplier (Java Platform SE 8)," Oracle, 12 1 2016. [Online]. Available: <https://docs.oracle.com/javase/8/docs/api/java/util/function/Supplier.html>.

pplier.html. [Använd 28 2 2016].

- [11] V. Hanstorp, "GitHub - DataSupplierFactory," 27 2 2016. [Online]. Available: https://github.com/HIG-Student/Programvaruteknik_Lab2/blob/6f7739c9e4656f63f4026ec420cbc41dbe08f0c5/Programvaruteknik_Lab2/src/se/hig/programvaruteknik/data/DataSupplierFactory.java. [Använd 28 2 2016].
- [12] Wikipedia, "Uniform Resource Locator --- Wikipedia{,} The Free Encyclopedia," Wikipedia, 19 2 2016. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Uniform_Resource_Locator&oldid=705717141. [Använd 28 2 2016].
- [13] Wikipedia, "Cache (computing) --- Wikipedia{,} The Free Encyclopedia," Wikipedia, 3 2 2016. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=Cache_\(computing\)&oldid=703118808#Web_cache](https://en.wikipedia.org/w/index.php?title=Cache_(computing)&oldid=703118808#Web_cache). [Använd 28 2 2016].
- [14] Wikipedia, "JSON --- Wikipedia{,} The Free Encyclopedia," Wikipedia, 27 2 2016. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=JSON&oldid=707212849>. [Använd 28 2 2016].
- [15] Wikipedia, "Comma-separated values --- Wikipedia{,} The Free Encyclopedia," Wikipedia, 11 2 2016. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Comma-separated_values&oldid=704338130. [Använd 28 2 2016].
- [16] owlike, "Genson - Fast and easy to use Java and Scala to Json conversion library," [Online]. Available: <https://owlike.github.io/genson/>. [Använd 28 2 2016].
- [17] Oracle, "Map (Java Platform SE 8)," Oracle, 12 1 2016. [Online]. Available: <https://docs.oracle.com/javase/8/docs/api/java/util/Map.html>. [Använd 28 2 2016].
- [18] V. Hanstorp, "GitHub - FootballSourceBuilder," 28 2 2016. [Online]. Available: https://github.com/HIG-Student/Programvaruteknik_Lab2/blob/3e7dd3984712d0cf7b1197d8429d1dbc3bc488fe/Programvaruteknik_Lab2/src/se/hig/programvaruteknik/data/FootballSourceBuilder.java. [Använd 28 2 2016].
- [19] V. Hanstorp, "GitHub - SMHISourceBuilder," 28 2 2016. [Online]. Available: https://github.com/HIG-Student/Programvaruteknik_Lab2/blob/3e7dd3984712d0cf7b1197d8429d1dbc3bc488fe/Programvaruteknik_Lab2/src/se/hig/programvaruteknik/data/SMHISourceBuilder.java. [Använd 28 2 2016].
- [20] Wikipedia, "Application programming interface --- Wikipedia{,} The Free Encyclopedia," Wikipedia, 20 2 2016. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Application_programmi

ng_interface&oldid=705963925. [Använd 28 2 2016].

- [21] V. Hanstorp, "GitHub - SMHILocation," 27 2 2016. [Online]. Available: https://github.com/HIG-Student/Programvaruteknik_Lab2/blob/6f7739c9e4656f63f4026ec420cbc41dbe08f0c5/Programvaruteknik_Lab2/src/se/hig/programvaruteknik/data/SMHILocation.java. [Använd 28 2 2016].
- [22] V. Hanstorp, "GitHub - FootballAndWeatherCombiner," 27 2 2016. [Online]. Available: https://github.com/HIG-Student/Programvaruteknik_Lab2/blob/6f7739c9e4656f63f4026ec420cbc41dbe08f0c5/Programvaruteknik_Lab2/src/se/hig/programvaruteknik/data/FootballAndWeatherCombiner.java. [Använd 28 2 2016].
- [23] Wikipedia, "Test-driven development --- Wikipedia, The Free Encyclopedia," Wikipedia, 17 2 2016. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Test-driven_development&oldid=705411040. [Använd 28 2 2016].

Bilaga 1 — Utskrift vid körning av "FootballAndWeatherCombiner"

Goals made at temperatures
(Goals : Temperature)

(2.0 : 7.9)	(3.0 : 6.4)	(3.0 : 7.4)
(3.0 : 7.4)	(2.0 : 9.0)	(3.0 : 6.2)
(5.0 : 13.3)	(1.0 : 14.1)	(4.0 : 13.1)
(5.0 : 6.5)	(5.0 : 7.9)	(1.0 : 7.3)
(3.0 : 11.1)	(4.0 : 14.6)	(3.0 : 13.7)
(3.0 : 6.3)	(3.0 : 6.7)	(3.0 : 6.2)
(2.0 : 3.7)	(3.0 : 9.3)	(0.0 : 7.9)
(4.0 : 11.5)	(2.0 : 6.4)	(0.0 : 7.2)
(3.0 : 6.4)	(2.0 : 7.9)	(4.0 : 10.8)
(2.0 : 9.0)	(3.0 : 7.4)	(2.0 : 9.1)
(2.0 : 7.9)	(5.0 : 13.3)	
(3.0 : 7.4)	(5.0 : 6.5)	
(5.0 : 13.3)	(3.0 : 11.1)	
(5.0 : 6.5)	(3.0 : 6.3)	
(3.0 : 11.1)	(2.0 : 3.7)	
(3.0 : 6.3)	(4.0 : 11.5)	
(2.0 : 3.7)	(3.0 : 6.4)	
(4.0 : 11.5)	(2.0 : 9.0)	
(3.0 : 6.4)	(3.0 : 14.5)	
(2.0 : 9.0)	(2.0 : 8.5)	
(1.0 : 14.1)	(2.0 : 7.3)	
(5.0 : 7.9)	(3.0 : 3.8)	
(4.0 : 14.6)	(4.0 : 11.6)	
(3.0 : 6.7)	(2.0 : 7.4)	
(3.0 : 9.3)	(2.0 : 10.0)	
(2.0 : 6.4)	(3.0 : 9.3)	
(2.0 : 7.9)	(3.0 : 5.5)	
(3.0 : 7.4)	(5.0 : 8.8)	
(5.0 : 13.3)	(3.0 : 6.2)	
(5.0 : 6.5)	(1.0 : 5.4)	
(3.0 : 11.1)	(4.0 : 7.0)	
(3.0 : 6.3)	(1.0 : 3.8)	
(2.0 : 3.7)	(5.0 : 14.2)	
(4.0 : 11.5)	(4.0 : 14.4)	
(3.0 : 6.4)	(2.0 : 7.1)	
(2.0 : 9.0)	(6.0 : 8.3)	
(2.0 : 7.9)	(3.0 : 14.5)	
(3.0 : 7.4)	(2.0 : 8.5)	
(5.0 : 13.3)	(2.0 : 7.3)	
(5.0 : 6.5)	(3.0 : 3.8)	
(3.0 : 11.1)	(4.0 : 11.6)	
(3.0 : 6.3)	(2.0 : 7.4)	
(2.0 : 3.7)	(4.0 : 2.6)	
(4.0 : 11.5)	(5.0 : 13.3)	
(3.0 : 6.4)	(1.0 : 6.6)	
(2.0 : 9.0)	(1.0 : 14.1)	
(2.0 : 7.9)	(5.0 : 7.9)	
(3.0 : 7.4)	(4.0 : 14.6)	
(5.0 : 13.3)	(3.0 : 6.7)	
(5.0 : 6.5)	(3.0 : 9.3)	
(3.0 : 11.1)	(2.0 : 6.4)	
(3.0 : 6.3)	(5.0 : 14.1)	
(2.0 : 3.7)	(4.0 : 4.7)	
(4.0 : 11.5)	(2.0 : 5.8)	