

Gruppuppgift parprogrammering

XML och JSON är två vanligt använda portabla dataformat som gör att data kan utbytas mellan olika applikationer, tjänster och hårdvaror. Under den här examinationsuppgiften kommer ni bygga en interaktiv webbplats/-tjänst, som på olika sätt presenterar innehållet i öppna dataset.

Lärandemål och examination

Följande lärandemål examineras genom den här examinationen:

- 1) Diskutera hur information bör struktureras och organiseras med syfte att utväxla information mellan olika webbtjänster, applikationer och informationssystem,
- 2) Redogöra för och jämföra två dataformat (XML och JSON) för strukturering, organisering och utbyte av information på webben
- 3) Praktiskt tillämpa dataformat och tekniker för dataöverföring mellan olika webbtjänster, applikationer och informationssystem.

Följande examinationsformer används och kan ge betyget:

- Kod och rapport (U/G/VG)
- Code review (U/G/VG)

Lärandemål 1) och 2) examineras genom rapport samt code review. Lärandemål 3) examineras genom inlämnad kod samt code review.

På den här kursen skriver ni ingen salstentamen. Istället examineras ni individuellt genom en muntlig code review. Detta för att kunna säkerställa att ni skrivit koden själva, samt att ni båda aktivt varit med och bidragit till lösningen.

Ni väljer själva om ni vill satsa på G eller VG. För att få VG på hela kursen krävs VG på den här examinationen samt VG på code review.

Uppnås G men inte VG, ges betyget G. Vid betyget U lämnar studenten in en ny lösning (byggd på nya dataset och datakällor) vid nästa examinationstillfälle.

Er uppgift

Ni ska bygga en interaktiv webbplats/-tjänst, som använder och presenterar innehållet i öppna dataset på ett interaktivt, användbart och självförklarande sätt. Beroende på vilket dataset (se mer nedan) ni väljer att jobba med behöver er lösning se olika ut.

Webbplatsens olika sidor och innehåll ska hänga samman på ett självförklarande sätt.

Vilken sorts webbplats ni bygger sätter bara fantasin, vald data (och i viss mån betygskriterierna) gränser för! Exempelvis kan ni bygga:

- en webbplats som presenterar miljörisker (översvämningrisk, brandrisker, etc.)
- en webbplats som innehåller en vädertjänst
- en e-handel för böcker/filmer/kläder/etc.
- en webbplats med filmrecensioner/bokrecensioner/kursrecensioner
- en webbplats där användaren kan söka fram aktuella kulturevenemang i specifik stad/land
- en webbplats för ett universitet/utbildningsverksamhet/etc.
- en webbplats med resor och restider (försäljning, förseningar, etc.)

Idén är att ni skall förvalta de kunskaper ni tillägnat er under vårens kurser (HTML5, CSS, JavaScript samt att utveckla för webbmiljön) och bygga en lösning med JSON och/eller XML. Förslagsvis använder ni även kunskaperna ni får från Multimedia för webben för att grafiskt presentera er data.

Arbetssätt

I den här uppgiften vill vi att ni tillämpar parprogrammering (pair programming eller pairing på engelska). Det innebär att ni arbetar två-och-två framför en dator. Den ene av er skriver kod, den andre av er granskar koden som skrivs och planerar för lösningar av kommande problem. Byt roller då och då. Tanken med parprogrammering är att lösningen granskas hela tiden medan den som kodar kan koncentrera sig på uppgiften. Förespråkare av parprogrammering menar att kvaliteten således kommer bli högre – färre buggar, fler lösningsförslag som diskuteras och provas, bättre planerad lösning som resulterar i en kod som är enklare att underhålla och bygga ut, m.m. (Agile Alliance u.å.). Dessutom lär ni er jättemycket av varandra! Parprogrammering är vanligt inom agila utvecklingsmetoder som Extreme programming (Extreme Programming u.å.).

Källkod (implementation)

Välj vilket/vilka dataset ni vill använda. Beroende på om ni väljer G- eller VG-nivå arbetar ni med JSON och/eller XML. Webbplatsen/tjänsten ska återspegla valet av dataset. Beroende på vilken sorts data ni väljer att arbeta med styr det hur er webbplats bör se ut och hur data bör presenteras. Vissa data visas bäst upp i text-form medan annan data behöver grafiska element eller till exempel kartor för att bli så självförklarande och användbart som möjligt. I Bilaga 1 finns exempel på hur data kan visas upp på grafiska sätt. Välj ett eller flera lämpliga dataset (XML eller JSON) till exempel från någon av nedanstående källor:

- <https://oppnadata.se/>
- <https://data.gov.uk/>
- <http://data.norge.no/>
- <https://www.data.gov/>

Utöver dessa datakällor finns det massor av API:er att välja på, här är ett axplock som kanske kan inspirera till vad ni kan bygga för slags webbplats/-tjänst:

- Public APIs (JSON) <https://github.com/toddmotto/public-apis>
- Open Movie Data Base (XML och JSON) <http://www.omdbapi.com/>
- Food2Fork recept-API <http://food2fork.com/about/api>
- Kart-API:er <http://bit.ly/2paGzs0>
- Pokémon (JSON) <https://pokeapi.co/>
- Last.fm musikdatabas (XML och JSON) <https://www.last.fm/sv/api>

Er lösning ska använda HTML5, CSS och JavaScript. Webbstandarder för er kod skall uppfyllas (koden ska validera).

Se betygskriterierna nedan.

Rapport (dokumentation)

Beskriv och analysera valt dataset (ett eller flera) i rapporten. Identifiera hur de olika datadelarna är representerade i er källa. Analysera hur väl filerna är strukturerade/formulerade utifrån rådande standarder – finns några avvikelser från standarder, exempelvis? Dokumentationen för er lösning ska kunna användas som underlag för underhållsarbete och ska därför vara tydlig. Se betygskriterierna nedan.

Betygskriterier

För G krävs att följande kriterier uppfylls:

- Lösningen (koden):
 - Är responsiv (dvs. fungerar bra på både små och stora skärmar)
 - Fungerar i senaste stabila versionen av Chrome och/eller Firefox
 - Är fullständig (alla efterfrågade delar i kod samt rapport ingår i inlämningen)
 - Validerar enligt rådande standarder för CSS och HTML5
 - Använder JSON-data
 - Presenterar data på ett lämpligt (grafiskt) sätt (t.ex. tabeller, grafer, kartor)
 - Är en helhet (sidor och innehåll hänger samman)
 - Är interaktiv
 - Är tydligt kommenterad – tänk er att någon/ni själva ska använda, vidareutveckla och/eller underhålla lösningen i framtiden. Se även informationen om fusk och plagiat i bilaga 2!
- Rapporten (dokumentation):
 - Beskriver vad det är ni har gjort (vad är det för webbplats/tjänst/applikation ni byggt? Vilka tekniker och datatyper har ni använt? Hur valde ni just det dataset ni valt? Varför valde ni just det dataset ni valt (fördelar, nackdelar)? Se lärandemål!)
 - Beskriver och jämför XML och JSON (likheter/skillnader, användningsområden, fördelar/nackdelar)
 - Beskriver hur det dataset ni valt är strukturerat
 - Beskriver hur ni använder valt dataset
 - Beskriver hur er lösning är uppbyggd

För VG krävs att kriterierna för G uppfylls, samt att:

- Ni använder XML-data utöver JSON-data (dvs. minst två olika dataset från två olika datakällor används i lösningen)
- Rapporten (dokumentation) innehåller analys av båda dataset (se kriterierna för G)

Betyget U ges om kriterierna för G ej uppfylls.

Inlämning

I lösningen skall följande ingå:

1. Källkod (och övriga filer som krävs för att köra er lösning). Lämnas in som ett .zip- eller .rar-arkiv.
2. Rapport (dokumentation). Som .pdf, .doc eller .docx. Plagiatkontrolleras.
3. Tydlig steg-för-steg-guide för att köra er lösning (annars kan det vara svårt för oss att granska era lösningar).

Källor

Agile Alliance. (u.å.). *Pair Programming* [Online]. Available: <https://www.agilealliance.org/glossary/pairing/> [Accessed 28 April 2017].

Bokfynd. (u.å.). *Bokfynd* [Online]. Available: <http://www.bokfynd.nu/9113045334.html> [Accessed 28 April 2017].

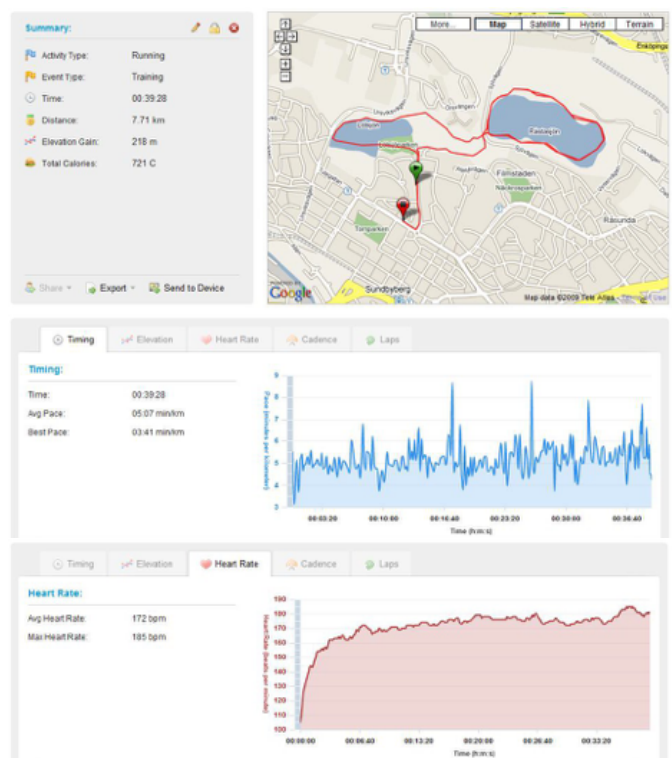
Extreme Programming. (u.å.). *Pair Programming* [Online]. Available: <http://www.extremeprogramming.org/rules/pair.html> [Accessed 28 April 2017].

Havs- och vattenmyndigheten. (2014). *Geodata via karttjänster direkt från HaV* [Online]. Available: <https://www.havochvatten.se/hav/samordning--fakta/kartor--gis/karttjanster.html> [Accessed 28 April 2017].

Bilaga 1. Exempel på presentation av data

Exempel 1: En applikation som visar träningsdata.

- Datum
- Total träningstid hh:mm:ss
- Total distans km
- Maxhastighet km/h
- Snitthastighet km/h
- Maxpuls bpm
- Snittpuls bpm
- Googlekarta som visar löprundan
- Graf för hastighet eller puls (använd en HTML5 Canvas eller SVG).



Exempel 2: En webbplats som jämför och visar bokpriser från diverse e-butiker (Bokfynd u.å.).

- Information om sökt bok
- Butiker som säljer boken
- Priser
- Leveranstid
- Länk till e-handeln

The screenshot shows the Bokfynd website interface. At the top, there's a search bar with 'Sök böcker' and a date 'Fredagen den 28 april 2017'. Below the header, a message states: 'Hämtar priser och leveranstider direkt från butikerna. Var god vänta (det tar 5 - 20 sekunder)'. The main content is divided into two columns. The left column, titled 'Bokinformation', displays the book 'Det andra könet' by Simone de Beauvoir, published by Storpocket, Norstedts, in 2012-06. It lists ISBNs and provides a link to 'Lägg i Jämförelsekorgen'. The right column, titled 'Priser och leveranstider', lists four retailers: Adlibris (Sverige), Bokus (Sverige), CDON (Sverige), and Glnza (Sverige). Each entry shows the total price, delivery time, and a link to the retailer's website. At the bottom, there's a 'Sortera på:' section with options like 'totalpris', 'bokpris', 'fraktpolis', 'butiknamn', 'kortast lev.tid', and 'längst lev.tid'.

Exempel 3: Tjänst som visar information kopplat till vatten på interaktiv karta (Havs- och vattenmyndigheten 2014).

- Fiskbestånd
- Djurbestånd
- Ortnamn
- Länder
- Vattendrag
- Vattendjup

The screenshot shows an interactive map from the Havs- och vattenmyndigheten. The map displays the Baltic Sea region, including parts of Sweden, Denmark, and Germany. On the left, there's a sidebar with a search bar 'Sök lager' and a list of layers: 'Miljöövervakning - kustfisk bestånd', 'Miljöövervakning - knubbsål bestånd', and 'Valda lager (3)'. The 'Valda lager (3)' section is expanded, showing 'Miljöövervakning - knubbsål bestånd' and 'Djupskugg... översiktlig'. The map itself shows various coastal features and place names like Göteborg, Borås, Rås, Kungälv, Varberg, Falkenberg, Halmstad, Åmål, Helsingborg, Eslov, Kolding, Odense, and København. A legend at the bottom right indicates 'Havs- och vattenmyndigheten, HAV' and 'Havets kartor'.

Bilaga 2. Fusk, plagiat och upphovsrätt vid lösning av programmeringsuppgift i kursen "Portabla format"

Vid användning av kodresurser från t.ex. forum, webbplatser, kurskamrater, och litteratur ska du hänvisa till källan, precis som när du skriver en text. Gör du inte detta resulterar det i en anmälan till disciplinnämnden som beslutar om påföljd. Påföljden kan vara en varning eller värsta fall avstängning för fusk.

Har du använt kod från externa resurser så som internet eller kurslitteraturen? Om ja:

1. Har du tydligt markerat i din kod vilka delar du kopierat, vem upphovsmannen är och hur du anpassat koden efter din lösning?
2. Har du tydligt visat att du förstår vad koden gör genom kommentarer i koden?

Lägg till följande som kommentarer i koden:

- Källan och/eller upphovsmannen.
- Förklaring med egna ord hur du anpassat koden till din lösning.
- Förklaring med egna ord vad koden gör.

Om källan har kommentarer som förklarar koden, behåll kommentarerna och, om källan är på annat språk, översätt kommentarerna till svenska.