Muistio
Parsi-projektin 2. koodikatselmointi

Muistio laadittu: 13.5.2011

Aika: torstaina 12.5.2011 klo 10:18 - 12:08
Paikka: sovellusprojektien kokoushuone Ag C226.2

Osallistujat

Ryhmä:
- Vili Auvinen
- Olli Kauppinen
- Juho Tammela, sihteeri

Tilaajat:
- Tommi Lahtonen

Ohjaajat:
- Jukka-Pekka Santanen
- Mikko Tyrväinen, saapui kohdassa 1

1. Aloitus

Aloitettiin Parsi-projektin toinen lähdekoodikatselmointi. Ryhmä oli julkaissut katselmoitavan lähdekoodin projektiorganisaatiolle tiistaina 11.5. Lähdekoodia käytiin läpi YouSourcesta videoprojektorin avulla.

Silmäiltiin lähdekoodia Mikko Tyrväistä odotellessa. Lahtosen ilmoitti, että hänen täytyy poistua noin klo 11:30.

Santanen huomautti, että myös tavallisten, lähdekoodin seassa olevien, kommenttien kielenä tulee olla englanti. Lisäksi jokaisen tiedoston alkuun täytyy lisätä kommentteihin tiedoston tarkoitus, tekijät, versionumero ja lisenssi.

Ryhmä kertoi, että sovelluksen vikasietoisuus on parantunut. Sovelluksen nykyinen versio kaatuu kuitenkin, jos odt-dokumentin otsikon fonttikoko on ilmoitettu prosentteina.

Mikko Tyrväinen saapui. Alettiin käydä lähdekoodia läpi tiedosto kerrallaan.

2. controller.py

Olisi kätevää, jos jokaisesta tarkastimesta olisi olemassa tynkä, joka esimerkiksi palauttaa pelkän virheilmoituksen. Tämä helpottaisi jatkokehitystä.

Hyvä tapa olisi kertoa except-ilmoituksissa, mikä virhe otetaan kiinni.

Merkistöjen decode ja encode on omituista. Jos haluaa käsitellä pelkkää UTF-8:aa, sekin onnistuu kertomalla selaimelle mitä merkistöä käytetään.

Pythonissa varattu sana 'list' on huono muuttujan nimi. Listan viimeisen alkion saa antamalla negatiivisen indeksin -1.

Tyrväinen kertoi, että hän on käyttänyt lähdekoodin tutkimisen apuna PyInt-analysaattorityökalua.

3. conversions.py

Muuttujan nimenä ei tulisi käyttää sanaa dict, joka on varattu sana. 'def'-rivillä ei tulisi olla välilyöntiä funktion nimen ja parametrit sisältävien sulkujen välillä.

4. docx_inspector.py

Käyttämättömät importit tulee siivota lopulta pois.

string.find(muuttuja, muuttuja) on deprecated. Mieluummin tulisi käyttää funktiota muuttuja.find(etsittävä).

Varattuja sanoja ei tulisi käyttää.

doc-muuttuja on ollut document.xml-tiedoston DOM-puun sisältävä globaali muuttuja testaamista varten. Sitä ei tulisi enää käyttää funktioissa.

Rivien pituudet kannattaa ottaa huomioon, sillä lähdekoodi joudutaan vielä tulostamaan.

printDict-funktiota ei enää käytetä, joten sen voi pyyhkiä.

Jollain rivillä on sisennys väärin, joten sisennykset kannattaa vielä tarkastaa.

5. email_sender.py

Importissa on kirjasto, jota ei löydy uudemmista Pythonin versioista. Tämä johtuu siitä, että palvelin käyttää Pythonin versiota 2.4.

Merkistönä tulee tässäkin tiedostossa käyttää UTF-8:aa.

6. gui.py

Kaikki HTML pitäisi korvata DOMilla. Voi käyttää valmiita tiedostorunkoja. Merkistöt jälleen UTF-8:lla.

7. odt_inspector.py

Pythonin varattuja nimiä kuten list ja zip ei tule käyttää muuttujien nimissä.

8. ooo_meta_inspector.py

Print-lauseet on siivottava pois. Parametrit ovat jääneet väärin joissain aliohjelmissa, koska aliohjelmia ei ole enää käytetty. Turhat import-lauseet tulee pyyhkiä pois.

ooo_meta_inspector ja mso_meta_inspector eivät sisällä montakaan käytettyä funktiota, joten ne voisi yhdistää johonkin yhteiseen moduuliin. Koska funktiot kuitenkin toimivat nyt, tätä voi vielä harkita.

Tiedostojen paketti/kansiojakoa kannattaa harkita.

9. requirements.py

Muuttujan nimenä käytetty property on Pythonin varattu sana.

getErrorMessage-funktiossa on käyttämätön parametri.

Kommentit puuttuvat luokkadokumentaatiota varten.

10. word_processing.py

Funktion nimi makeDict voisi olla kuvaavampi, esimerkiksi makeDocumentDict.

checkers-dictin voisi alustaa tyhjänä, ja aina kun kirjoittaa funktion, lisää funktion dictiin. Siinä tapauksessa kaikkia funktioita ei näe kerralla, mutta checkers-listan voisi alustaa heti alussa. chekcers-dictin tämänhetkisessä alustuksessa on selkeämpää esitellä vain yksi avain-alkio -pari riviä kohden.

file on Pythonin varattu sana, eikä sitä kannata käyttää muuttujan nimenä.

11. common_methods.py

checkEmailAddress tarkastaa vain, löytyykö @-merkki merkkijonosta. Tehokkaamman tarkastuksen voisi tehdä esimerkiksi regular expressioneilla. Valmiita toteutuksia tähän löytynee internetistä.

Sähköpostiosoitetta ei vielä verrata käyttöliittymän lomakkeelle syötettyyn sähköpostiosoitteeseen. Jatkossa dokumentin kansisivun sähköpostiosoitetta verrataan myös siihen.

12. JavaScript-tiedostot

JavaScript-tiedostoja voi tarkastaa JSLint.com -sivuston avulla. Käyttöliittymän JavaScript-tiedostot käyttävät kuitenkin jQuery-kirjastoa, jonka syntaksista JSLint antaa aiheettomia virheilmoituksia.

JavaScriptissä syntaksi '==' on eri asia kuin '==='.

Sähköpostiviestin koostaminen kannattaa tehdä palvelimella, ei JavaScriptissä. Lynx/linx auttaa sähköpostin koostamisessa.

13. Yhteenveto

Lähinnä lähdekoodi kaipaa komentteja ja siistimistä.

Jotkut aiemmat sovellusprojektit ovat tehneet luokkadokumentaatiot myös pdf-muotoon. Silloin luokkadokumentaatiot on saatu helposti paperimuotoon ja myös projektikansioon. PyDoc osaa tehdä luokkadokumentaatiot vain html-muotoon. EpyDoc on sitä uudempi työkalu luokkadokumentaation tekemiseen, jolla myös pdf-muotoiset luokkadokumentaatiot onnistuvat.

Lähdekoodien kommentointitavoissa otetaan mallia Pythonin omista kommentointitavoista.

Lahtonen poistui paikalta.

14. Ryhmän kysymykset

Lahtosen poistuttua ryhmä kysyi vielä Tyrväisen mielipidettä joihinkin lähdekoodissa askarruttaviin asioihin.

14.1. odt_inspector.py

Kauppinen kysyi getStyle-funktion tavasta tulkita dokumentista parsitut tyyliasetukset translateDict-dictin avulla. Tyrväisen mielestä tapa on ihan hyvä, mutta jos tällaista tehdään enemmänkin, muunnokset sisältävät dictit voisi sijoittaa johonkin yhteen paikkaan. Translate on ehkä huono termi kuvaamaan translateDictin käyttötarkoitusta.

Kauppinen ja Tyrväinen keskustelivat desimaalilukujen pyöristämiseen liittyvistä ongelmista.

14.2. conversions.py

convertCmOrInDictToString-funktiossa kannattaa varautua else-lauseella siihen, jos mittayksikkö ei olekaan sentti eikä tuuma. Prosentteina ilmoitetut mittayksiköt pitää käsitellä odt_inspector-tasolla, sillä ne vaativat tyylihierarkian läpikäyntiä.

14.3. docx_inspector.py

checkSections-funktio on pitkä, mutta toimii. Se voi olla työläs pilkkoa osiin, joten sitä ei ehkä kannata tehdä projektin puitteissa.

15. Lopetus

Lähdekoodien tulostaminen YouSourcesta ei onnistu helposti. Myös Eclipsestä saa tulostettua kera rivinumeroiden. Ainakaan Eclipsen aiemmat versiot eivät kuitenkaan ole osanneet rivittää liian pitkiä rivejä oikein. Yhdelle paperiarkille kannattaa tulostaa neljä sivua lähdekoodia.

Sovellusraportissa tulee kirjata erikseen sovelluksen puutteelliset toteutusratkaisut ja jatkokehitysideat.

Lähdekoodien tiedostot kannattaa ryhmitellä omiin kansioihin, esimerkiksi inspectors-kansio, yleiset funktiot-kansio ja niin edelleen.