

Dokumentaatio

T-106.1215 Ohjelmoinnin peruskurssi, osa 2:n projektityö

Henkilötiedot

Nimi: Juho Salmi

Koulutusohjelma: AUT

Vuosikurssi: 5

Päiväys: 7.5.2013

Yleiskuvaus

Työkalu opintojen suunnitteluun, joka kertoo käyttäjälle mm. esitetietojen täyttymisestä sekä opintopistemääristä. Opintosuunnitelman voi ladata ja tallentaa. Työ on toteutettu melko matalalla työläysasteella.

Käyttöohje

Aja gui.py ja avaa valmiina oleva opintosuunnitelmatiedosto esim. aut.csv. Ohjelma avaa opintosuunnitelman. Klikkaamalla kursseja, voit siirtää niitä aiemmaksi, myöhemmäksi, suoritetuksi tai poistaa ne suunnitelmasta. Kurssin nimi näkyy oranssina, jos sen esitetietokurssit eivät täyty. Periodien tekstit näkyvät violettina, jos niillä on liikaa opintopisteitä ja punaisena, jos niillä on liian vähän opintopisteitä. Lopuksi suunnitelman voi tallentaa.

Ohjelman rakenne

Periodiluokka: `period.py`

Sisältää tiedon, missä periodissa kurssi alkaa ja missä päättyy sekä ylikuormitetut metodit periodien vertailuun.

Kurssiluokka: `course.py`

Sisältää kurssitiedot: kurssin nimi, opintopisteet, periodi, kuvaus, esitiedot. Käyttää luokkaa `Period`.

Opintosuunnitelmaluokka: `studyPlan.py`

Kaikki tarvittava opintosuunnitelman hallintaan. Käyttää luokkia `Period` ja `Course`.

- Lista sanakirjoja aikataulutetuista kursseista avaimena kurssin nimi.
- Set-tietorakenne suoritetuista kurssinimistä
- Set-tietorakenne aikatauluttamattomista kurssinimistä.
- Sanakirja kaikista kursseista avaimena kurssin nimi.

`scheduleCourse(self, courseName, year, period):`

Aikatauluttaa kyseisen kurssin.

getCourse(self, courseName):

Palauttaa vuoden ja periodin, jolloin kurssi on tarkoitus suorittaa.

listCourseNamesWithDissatisfiedPrerequisites(self):

Listaa kurssit, joiden esitiedot eivät täyty.

listCreditsPerPeriod(self):

Tekee listan, josta voi poimia, kuinka paljon kussakin periodissa suoritetaan opintopisteitä.

IO: plannerIO.py

1. Lataa listan kaikista kursseista csv-tiedostosta
2. Lataa opintosuunnitelman csv-tiedostosta
3. Tallentaa opintosuunnitelman csv-tiedostona

Käyttää luokkia Period, Course ja StudyPlan.

Graafinen käyttöliittymä: gui.py

Käyttää kaikkia muita luokkia.

class MainWindow(QtGui.QMainWindow):

Pääikkunahallinta.

class CourseButton(QtGui.QPushButton):

Kurssinappi valikkoineen.

class StudyPlanner(QtGui.QWidget):

GridLayout kurssinappeja ja muuta käyttöliittymää. Sisältää jyvän algoritmin kurssien järjestämiseen mahdollisimman visuaalisesti tiiviisti.

Algoritmit

Gui:n StudyPlanner-luokassa on kurssinappien asetteluun algoritmi, joka osaa asetella erimittaiset kurssit mahdollisimman tiiviisti: yhden periodin kurssit lomittuvat mukavasti pidempien kurssien kanssa, eikä joka ikiselle kurssille tarvitse omaa riviään. Lyhykäisyydessään algoritmi etsii oikeiden periodien alta kolon, johon kurssi mahtuu.

StudyPlan-luokassa on kevyempiä algoritmeja oikeiden tarvittavien tietojen hakemiseen kurssitiedoista.

PlannerIO-luokassa on kevyet algoritmit csv-tiedostojen käsittelyyn.

Tietorakenteet

StudyPlan-luokassa olen käyttänyt Set-tietorakennetta aikatauluttomiin ja suoritettuihin kursseihin. Avaimena käytän kurssien kohdalla kurssinimeä, eikä aikatauluttamattomien ja suoritettujen kurssien kohdalla ole muuta relevanttia kuin nimi. Itse aikataulutettujen kurssien

tietorakenne on lista sanakirjoja. Kyseisen listan indeksit kuvaavat kutakin vuotta. Kunkin vuoden kohdalla olevan sanakirjan avain on kurssin nimi ja arvo kurssin periodi. Tämän havaitsin kätevimmäksi tavaksi esittää opintosuunnitelman. StudyPlan luokassa on myös sanakirja kaikista saatavilla olevista kursseista avaimena kurssin nimi ja arvona Course-olio. Varsinaiset kurssitiedot haetaan kyseisestä sanakirjasta.

Tiedostot

PlannerIO käsittelee CSV-tiedostoja valmiin Pythonin CSV-kirjaston avulla. Ohjelma käyttää kahdenlaisia CSV-tiedostoja: kurssilistauksia ja opintosuunnitelmia. Kurssilistauksia ei voi ohjelmalla tallentaa, vaan ne pitää tehdä käsin tai jollakin muulla ohjelmalla. Opintosuunnitelmia voi ladata ja tallentaa.

Testaus ja ohjelman tunnetut puutteet ja viat

StudyPlan-luokan yksikkötestaamiseen käytettiin test.py:tä ja PlannerIO:n testaamiseen testIO.py:tä. Kun graafinen käyttöliittymä luotiin, meni homma enemmän käsin testaamiseksi, kun siitä tuli suoraviivaista ja kätevää. Ohjelmaa ei ole todellakaan testattu läpikotaisesti.

Virhe käsittelyä ei ehditty juurikaan toteuttaa, joten ohjelman saanee kaatumaan helposti etenkin IO:n kohdalla. Ohjelmalla ei pysty luomaan kokonaan uutta opintosuunnitelmaa, eikä sillä voi katsella, mitkä kurssit on suoritettu, mitä ei ole aikataulutettu ja mitä kursseja voisi suunnitelmaan lisätä. Ohjelmaan on toistaiseksi kovakoodattu tiedosto, josta kaikki kurssit löytyvät sekä op/periodin ylä- ja alarajat. Nämä olisivat seuraavat ominaisuudet, jotka ohjelmaan tulisi toteuttaa. Kun ohjelman toiminnallisuus on kunnossa, pitäisi kehittää virhe käsittelyt ja tehdä kattavat testit.

Hyvät ja huonot

Ohjelman rakenne ja logiikka ovat onnistuneet. PyQt tuotti harmaita hiuksia, kun en saanut oikealle hiiren näppäimelle haluamaani toiminnallisuutta. Tein lopulta toteutuksen eri tavalla, ja nyt on voittajafiilis PyQt:n päihittämisestä. Ohjelma myös on tehty siten, että sitä on helppo lähteä kehittämään eteenpäin.

Huonot jutut mainitsinkin edellisen osion puutteissa.

Poikkeamat suunnitelmasta, työjärjestys ja aikataulu

Lopullinen toteutus on aika kaukana alkuperäisestä. Tietorakenteet kehittyivät, ylimääräisiä luokkia ja toiminnallisuuksia pudotettiin pois ja graafinen käyttöliittymä toteutettiin kaikesta huolimatta. Automaattista suunnittelualgoritmia ei toteutettu, vaan keskityttiin käyttöliittymässä tapahtuvaan suunnitteluun. Testausta tehtiin alussa sääntillisesti yksikkötestauksin, mutta graafisen käyttöliittymän tullessa kuvioihin yksikkötestaus jäi. Aikataulu ei vähemmän yllättäen pitänyt, vaan koodaaminen jäi viimehetken sykähtelyihin ja DL:n tuolle puolen.

Arvio lopputuloksesta

Eihän tätä ajan kanssa tehty. Pääsin omiin oppimistavoitteisiini: opin vähän Pythonin

sielunelämää, käytännön ohjelmistokehitystä ja graafisen käyttöliittymän toteutusta. Sen lisäksi muulla kuin kurssin läpäisyllä ei oikeastaan ole minulle väliäkään.

Esimerkki käyttöliittymästä

The screenshot shows a window titled "Study Planner" with a "File" menu. The main area is divided into three columns representing "Vuosi 1 (59.0op)", "Vuosi 2 (59.0op)", and "Vuosi 3 (27.0op)". Each column contains a grid of course selection buttons. The buttons are organized into rows, with some rows having multiple buttons. The buttons are labeled with course codes and names, such as "Kie-98.7001", "AS-84.1132", "Mat-1.1131", "Mat-1.1132", "KE-35.9700", "AS-74.2112", "AS-84.2161", "T-106.1111", "S-88.1110", "S-104.1021", "S-104.1022", "AS-74.1101", "Mat-1.2620", "AS-116.2120", "AUT.kand", "Kie-98.7002", "Mat-1.1120", "TU-22.1101", "Kie-98.xxx", "AS-0.2230", "Mat-1.1110", "AS-0.1103", "Kon-41.2010", "AS-74.2111", "T-106.1210", "Tfy-3.1540", "AS-0.1501", "AS-84.1137", "S-104.1011", "AS-0.1401", "S-55.1100", "T-106.1223", "S-104.1012", and "AS-0.1502". The buttons are arranged in a grid that allows for selecting courses for each year.

Kirjallisuusviitteet ja linkit

Python v2.7.3 documentation (<http://docs.python.org/2.7/>)

PyQt Wiki (<http://wiki.python.org/moin/PyQt>)

Zetcode, PyQt tutorial (<http://zetcode.com/tutorials/pyqt4/>)

Liitteet

Ohjelman lähdekoodi.