

Kalenteri Kehittämisopas

Maiju Karjalainen

13.3.2016

Sisällys

Johdanto.....	3
Keskeiset ohjelmassa käytettävät käsitteet	4
Ohjelman käyttötarkoitus ja rajoitteet.....	5
Ohjelman tietomalli.....	6
Ohjelman toteutus	7
Jatkokehitys ja tunnetut ongelmat.....	8

Johdanto

Tämä kehittämisopas on tarkoitettu Kalenteriohjelman kehittäjille ja ylläpitäjille.

Kalenteriohjelma pitää kirjaa käyttäjän syöttämistä tapahtumista ja näyttää ne päivämäärän mukaan.

Ohjelma on testattu toimivaksi Windows 7 käyttöjärjestelmällä ja Python-kielen versio 3.0 tulkilla.

Tässä oppaassa on kuvaus ohjelmasta ja tiedot ylläpitoa varten.

Keskeiset ohjelmassa käytettävät käsitteet

Ohjelma käsittelee taulukossa 1 esitettyjä tietoja.

TAULUKKO 1 MUISTUTUSTEN TIEDOT

Päivämäärä	Kalenteria selattaessa käytettävä päivämäärä
Muistutuksen päivämäärä	Muistutus –tietueessa käytettävä päivämäärä
Aika	Muistutuksen ajankohta
Aihe	Muistutuksen aihe
Hälytys	Halutaanko muistutukselle hälytys vai ei

Ohjelman käyttötarkoitus ja rajoitteet

Ohjelma on tarkoitettu yhden tai useamman henkilön käytettäväksi muistutusten tallentamiseen ja näyttämiseen kalenterimuodossa. Ohjelmalla voi lisätä ja poistaa muistutuksia. Ohjelmassa voi selata päivämäärää painikkeilla tai kirjoittamalla päivämäärä. Päivämäärän kohdalla näytetään sille tallennetun muistutuksen aihe ja ajankohta. Poistaminen tapahtuu antamalla muistutuksen päivämäärä ja aihe.

Ohjelmassa ei pysty tarkastelemaan kaikkia muistutuksia kerralla, ainoastaan päivä kerrallaan. Muistutuksia ei myöskään voi muokata muuten kuin poistamalla ja tekemällä uuden muistutuksen.

Ohjelman tietomalli

Ohjelmassa muistutuksen tiedot mallinnetaan olioluokan Muistutukset avulla.

Muistutus

teksti: string

paiva: string

aika: string

halytys: int

__init__(teksti, paiva, aika, halytys)

set_teksti(teksti)

set_aika(aika)

set_paiva(paiva)

set_halytys()

get_teksti()

get_paiva()

get_halytys()

__str__()

Muistutus-olioiden tiedot tallennetaan tekstitiedostoon, eri tiedot erotettuna £ -merkillä.

Tekstitiedoston käsittelyä on toteutettu taulukossa 2 esitetyt metodit.

TAULUKKO 2: TEKSTITIEDOSTON KÄSITELYN METODIT

tallenna()	Käytetään uuden muistutuksen lisäämistä varten avatussa ikkunassa. Tarkistaa syötteiden muodon. Tallentaa muistutus-olion tiedot tekstitiedostoon.
ok()	Käytetään muistutusten poistamista varten avatussa ikkunassa. Tarkistaa syötteet. Hakee päivämäärän ja aiheen avulla muistutuksen tekstitiedostosta ja poistaa sen.
paivita()	Käytetään pääikkunassa. Aina kun päivämäärä vaihtuu, funktio käy läpi tekstitiedoston ja hakee sieltä päivään kuuluvat muistukset ja kirjoittaa ne tekstikenttään.

Ohjelman toteutus

Ohjelma koostuu kolmesta ikkunasta, joista kukin on omassa tiedostossaan, sekä muistutus –luokasta. Taulukossa 3 on esitelty ohjelman neljä moduulia.

TAULUKKO 3 OHJELMAN MODUULIEN KUVAUS

Moduuli	Kuvaus
Käyttöliittymä.py	Sisältää graafisen käyttöliittymän määrittelyn olioluokan GUI avulla. Moduulin ajaminen käynnistää ohjelman.
Muistutukset.py	Sisältää olioluokan Muistutus
Uusi_ikkuna.py	Sisältää uutta muistutusta varten avatun ikkunan graafisen käyttöliittymän määrittelyn olioluokan Uusi_ikkuna avulla.
Poista.py	Sisältää muistutusten poistamista varten avatun ikkunan graafisen käyttöliittymän määrittelyn olioluokan Poista avulla.

Ohjelmiston pääohjelma on moduulissa Käyttöliittymä. Ohjelma käynnistää graafisen käyttöliittymän luomalla olion luokasta GUI.

Luokka GUI sisältää taulukossa 4 kuvatut metodit.

TAULUKKO 4 LUOKAN GUI METODIT

Metodi	Kuvaus
__init__()	Luo ohjelmaikkunan
paivita()	Lukee muistutusten tekstitiedostoa, hakee sieltä päivämäärän mukaan muistutuksen ajan ja aiheen ja kirjoittaa sen pääikkunan tekstikenttään.
enter()	Kun päivämäärää on muutettu kirjoittamalla se ja painettu enteriä, kutsutaan paivita() –funktiota.
vasen()	Kun painetaan '<' –nappia, päivämäärästä vähennetään yksi päivä ja tekstikenttä päivitetään.
oikea()	Kuten edellä, mutta päivä lisätään.
uusi()	Avataan uusi ikkuna muistutusten lisäämistä varten.
poista()	Avataan uusi ikkuna muistutusten poistamista varten.

Ohjelmisto on toteutettu Eclipse –ohjelmistokehitysvälineellä käyttäen PyDev:iä

Jatkokehitys ja tunnetut ongelmat

Kalenteriohjelman alkuperäiseen suunnitelmaan kuului hälytys joka soisi muistutuksen aikaan. Asiakkaan kanssa päätettiin jättää tämä ominaisuus tässä vaiheessa vielä pois hieman tarpeettomana ja työläänä toteuttaa. Ohjelmassa on kuitenkin kohta hälytystä varten, uutta muistutusta lisätessä kysytään halutaanko hälytystä. Tämä kohta on jätetty ohjelmaan tarkoituksella helpottamaan tulevaa kehitystyötä. Ohjelmasta voisi saada käytännöllisemmän seuraavilla lisäyksillä:

- Muistutukset voisi näyttää aikajärjestyksessä. Lisäisi käyttömukavuutta
- Lista jossa näkyisi kerralla kaikki muistutukset

Ohjelmassa on ainakin yksi havaittu virhetilanne. Jos samana päivänä on useita samannimisiä tapahtumia, tapahtumaa poistettaessa ne kaikki poistetaan. Muistutuksen poistaminen ei ole tämän tarkempi, jotta käyttäjän ei tarvitsisi kirjoittaa muistutuksen kaikkia tietoja sitä poistaessa. Muistutuksen poistamista pitäisi lähestyä hieman eri suunnalta, jotta siitä saataisiin käytännöllisempi ja tarkempi.