

TCalendar (TeekkariKalenteri)

Yleissuunnitelma

Atte Isopuro, 79965R

TIK v.1 17-21.02.2011

Vaikean tason toteutus Teekkarikalenterista, eli GUI tekstikäyttöliittymän sijaan sekä yhteensopivuus iCal-tallennusformaatin kanssa.

Go To tekstikenttään voi kirjoittaa päivämääriä muodoissa ddmm, ddmmyy, ddmmyyyy tai dd.mm.yyyy (esim 201185 tai 15061956 tai 14.3.1973). Save ja Load tallentavat ja vastaavasti hakevat muistista iCal-muotoisia tekstitiedostoja. New- napilla saa apupaneelin johon voi erinäisiin tekstikenttiin kirjoittaa uuden tapahtuman alun, lopun, keston, toistettavuusehdot, prioriteetin ja kategorian. Import-vaihtoehto mahdollistaa .ics-tiedostojen lataamisen, ilman että nykyinen kalenteri häviää. Ylälaidassa olevalla valikkorivillä on myös navigointipainikkeet viikkonäkymää varten. Ohjelman oikeassa laidassa on tyhjä paneeli, johon ilmestyy täytettävät tekstikentät jos ollaan luomassa/muokkaamassa tapahtumaa. Painettaessa Month View nappia ilmestyy kyseiseen paneeliin päällekkäin sarja pieniä laatikoita joista voi lukea siinä kuussa korkean prioriteetin omaavien tapahtumien kategorian ja alku/loppu ajan ja valita josko tapahtumaa haluaa muokata tai poistaa. Näiden alle ilmestyy lyhyt lista jossa näkyy ajankulutus kategorioittain.

Käyttöohjeena on kooditiedostojenkin liitteenä oleva Readme-teksti:

The GUI can be run from the RunGUI class.

HowTo:

NOTE: All dates should be entered as ddmm, ddmmyy, ddmmyyyy or dd.mm.yyyy

WARNING: Importing .ics files built by other programs and then saving onto the same file will incur data loss: this program is far from fully compatible.

WARNING: The program will not ask for confirmation when Saving/Loading/Deleting: pressing the Delete button will Delete the specified event immediately and loading a new calendar will automatically overwrite your current calendar.

In the File menu:

- New allows you to define a new event.
- Import allows you to import .ics files.
- Load allows you to load a preexisting .ics file. This will also overwrite your current calendar: any unsaved progress will be lost.
- Save allows you to save your current calendar.

On the menu bar:

- Previous, Current and Next: these buttons allow you to navigate weeks
- Go To Date: enter a date and press enter: the week view will show the week containing the date.
- Month View: Shows a list of all high priority events in the current

month, and a summary of durations by category.

On the week view:

- Events are shown as colored rectangles. The topmost part contains the category of the event and its start and end times.
- If several events overlap, the sides of the area where they overlap will be colored red.
- By clicking on the weekview, you can bring up a list of all the events that are valid at the point of the click. You can then look at them more closely or change them by clicking Edit. Delete will remove the event: be warned, the program will not ask for confirmation.

On the left side tool panel:

- If you are creating a new event, pressing the Apply button will create a new event with the specified parameters. If you are editing an event, pressing the Edit button will change the event in question and pressing the Apply button will create a new event with the specified parameters. WARNING: Pressing the Delete button will delete the event, not the occurrence: if the event is repeating, all of its occurrences will also be removed.

Ohjelma on jaettu pääosin kalenteri ja GUI-osioihin. Kalenteriosio hoitaa tapahtumien käsittelyn: tarkistaa tapahtumakeston, josko on voimassa erityisenä päivänä jne. Jokaista tapahtumaa kuvaa yleisesti Event-rajapintaa toteuttava olio, tässä toteutuksessa MeetingEvent-luokka. Event-tyyppiset luokat koostuvat suurimmaksi osaksi yksinkertaisista get/set metodeista joita käytetään niiden rakennuksen ja muokkaamisen yhteydessä. Metodi getSerialization palauttaa StringBuilder-olion johon on rakennettu iCal-formaatin mukainen kuvaus tapahtumasta. isValid-metodi tarkistaa onko tapahtuma voimassa parametrina annetulla aikavälillä. Luokka viittaa kaikissa toistumiseen liittyvissä metodeissaan luokan Calendar julkisiin kenttiin (esim. Calendar.MONTH jos tapahtuma toistuu joka kuukausi). Luokka getDurationInMonth taas laskee kuinka monen millisekunnin verran tapahtuma on voimassa yhden kuukauden ajan (alkaen parametrina annetusta kuun ensimmäisestä päivästä).

TCalendar luokka itsessään tarkistaa mitkä kaikki tapahtumat ovat voimassa erityisinä päivinä ja hoitaa IO-operaatiot.

Graafinen käyttöliittymä TCalGUI käyttää apunaan muutamaa pienempää apuluokkaa: JToolPanel, JTimeBlock, JEventPreview ja JErrorFrame. TCalGUI itse ylläpitää viikkonäkymää, eli grid-näkymää johon on merkitty viikon tapahtumat erivärisinä- ja korkuisina suorakulmioina. Luokka toteuttaa ActionListener-rajapinnan actionPerformed-metodin ja kuuntelee kaikkia File-menyyn nappeja, Previous, Next ja Current-nappeja, Go To tekstikenttää ja Month View nappia. Luokka piirtää tapahtumat esille pyytämällä TCalendar-luokalta listan jokaiselle viikonpäivälle niistä tapahtumista jotka ovat silloin voimassa.

Käyttöliittymän apukomponentit kuvaavat pienempiä osia joilla saattaa olla omat napit:

JToolPanel lukee tekstikenttiinsä kirjoitetut tekstit ja luo niiden avulla uuden tapahtuman tai muokkaa vanhaa (riipuen josko konstruktorille syötettiin null tai

Event-olio).

JTimeBlock on viikkonäkymässä oleva aikablokki, tässä ohjelmassa 15 min pitkä. Painamalla blokkia hiirellä se tarkistaa omasta kentästään, mitkä tapahtumat ovat sillä hetkellä voimassa juuri kyseisessä blokissa. Se myös hoitaa oman värjäyksensä: taustaväri määräytyy tapahtuman prioriteetin mukaan ja jos monta tapahtumaa on päällekkäin blokin sivut värjätään punaisiksi.

JEventPreview on pieni esinäkömä tapahtumille jota käytetään listatessa tapahtumia (esim. kuukausittaisnäköymässä). Jos komponentissa olevaa Edit-nappia painetaan, se syöttää itselleen kuuluvan Event-olion JToolPanelin konstruktorille.

JErrorFrame on simppeli virheikkuna jonka voi pistää näyttämään viestejä käyttäjälle (esim. poikkeusten selityksiä). Ok-nappi vain poistaa ikkunan, ei muuta.

Näiden luokkien lisäksi on sekä calendar että calendarGUI packageissa apuluokat, DateCalc ja TCalParser. DateCalcissa on muutama metodi GregorianCalendar-luokilla laskemista varten (esim. startOf luokka palauttaa olion joka on "nollattu" parametrina annetun kentän alkuun). TCalParser taas tulkitsee käyttäjän tekstisyötteitä.

Ohjelman yksinkertaisimpiin algoritmeihin kuulunee JTimeBlock-luokan addEvent-metodi, joka lisää Event-olion omaan listaansa, värjää itsensä ja kutsuu rekursiivisesti seuraavan blokin addEventiä, kuin linkitetyssä listassa. Toteutus salli suhteellisen helpon tavan varmistua siitä että jos tapahtuma jatkuu keskiyön yli se myös näkyy käyttöliittymässä, ilman että jokaiselle päivälle tarvitsisi erikseen maalata tapahtuman osia. (Tämä toteutus tosin ei toimi täydellisesti, mistä lisää bugiosiossa).

Ohjelman keskeisin algoritmi on kiistämättä isValid-metodin tarkistusprosessi. Tarkistettuaan erinäiset perustavat mahdollisuudet (esim. onko tapahtuma toistuva, muussa tapauksessa vastaus on triviaali), metodi laskee ensin erotuksen kyseisen MeetingEvent-olion alkupäivämäärn ja parametrina annetun päivämäärän välillä katsomalla ainoastaan olion omaa toistokenttää (eli jos MeetingEvent event toistuu kuukausittain (Calendar.MONTH), metodi tarkistaa kuinka monta kuukautta on event.start:ista parametrina annettuun start:iin). Jos tämä luku modulo tapahtuman oma intervalli ei ole nolla, tapahtuma ei ole voimassa kyseisenä päivänä (jos event toistuu joka toinen kuukausi, esim. tammi, maaliskuu, touko, start-kentän kuukausi-arvo on 0 = January. Jos parametrina annetaan jokin päivämäärä huhtikuussa on sen kuukausi-arvo 3 = April. Näiden erotus on 3, mutta event toistuu joka toinen kuukausi, jolloin sen intervalliarvo on kaksi. $3 \% 2 \neq 0$, joten metodi palauttaa false).

Jos yhdestäkään modulolaskennasta saadaan 0, tarkoittaa se että tapahtuma tulee olemaan voimassa samalla viikolla/kuussa/vuonna kuin parametrina annettu päivämäärä. Lopullinen tarkistus tehdään ynnäämällä tapahtuman omiin start ja end kenttiin aikaisemmin saatu kenttäerotus ja laskemalla (GregorianCalendarin omilla after ja before metodeilla) josko uusi start tai end sattuvat parametreina annettujen päivämäärien väliin tai molemmille puolille.

Pseudokoodina:

```

isValid(start, end) {
    difference = getDifference(this.start, start);
    if (difference % this.interval != 0) then return false;

    else:
        Period fork = new Period(start, end);
        futurestart = start.add(difference)
        futureend = end.add(difference)
        if (fork.contains(futurestart) ||
            fork.contains(futureend) ||
            futurestart.before(fork) && futureend.after(fork))
        then
            return true;

```

Projektikuvauksessa oli vaadittu ettei tulevan voimassaolopäivän laskeminen saanut kestää loputtoman pitkään. Ylläoleva toteutus takaa vakioaikaisen laskutoimituksen aikavälistä riippumatta.

TCalendar tallentaa kaikki toistumattomat tapahtumat TreeMap:iin josta saa nopeasti noudettua kaikki tapahtumat tietyltä aikaväliltä (esim. viikko). Toistuvat tapahtumat ovat ArrayList-rakenteessa koska ne joudutaan käymään yksitellen läpi jotta voidaan tarkistaa ovatko ne voimassa tietyllä aikavälillä. ArrayListassa on lisäksi helpot lisäys/poisto operaatiot.

Tiedostomuoto on, kuten yllä mainittu, iCal-yhteensopiva .ics-tekstiformaatti johon merkitään kentittäin tapahtumaan liittyvät tiedot tekstimuodossa. Tyhjä esimerkki, jossa täytettävät kentät on merkitty hakasuluin [] ja pakollinen kenttä * merkillä:

```

*BEGIN:VCALENDAR
*PRODID:TCalendar.tkk/aisopuro
*VERSION:2.0

*BEGIN:VEVENT
*DTSTAMP:[yyyymmdd]T[hmmss]
UID:[esim. TCalendar@aisopuro.tkk]
*DTSTART:[yyyymmdd]T[hmmss]
*DTEND:[yyyymmdd]T[hmmss] TAI DURATION:[esim. 1H20M (tunti 20min)]
CATEGORIES:[esim. Tentti]
PRIORITY:[1-9]
RRULE:FREQ=[Daily/weekly jne.],INTERVAL=[1,2,3,4...]
*END:VEVENT

*END:VCALENDAR

```

Yhteen .ics tiedostoon voi pistää monta tapahtumaa kunhan ne kaikki alkavat rivillä BEGIN:VEVENT ja loppuvat END:VEVENT. Esimerkkitiedostoja löytyy TCalendar/iCalTestFiles kansioista.

Testaus toteutui suurimmaksi osaksi käyttöliittymän kautta, käyttäen eri paikkoihin sijoitettuja `System.out.println` komentoja. Ennen käyttöliittymän valmistumista kopioin testattavat metodit erillisiin testiluokkiin jossa ajoin niitä erinäisillä syötteillä. Ohjelma rakentui metodi kerrallaan, ja sitä mukaa kuin metodin valmistuivat niitä tuli myös testattua.

Tunnetut viat:

-Jos tapahtuma alkaa ennen keskiyötä maanantaina ja loppuu sen jälkeen, ohjelma ei maalaa sitä viikkonäkymään. Ongelma lienee joko `TCalGUI:n` `getStartingBlock` metodissa tai `MeetingEvent:in` `isValid`-metodissa: näitä tutkimalla ongelmaan löytyisi luultavasti ratkaisu ennen pitkää.

-Jos toistuva tapahtuma toistuu sellaiseen kohtaan viikkoa, että sen päättymisaika (keskiyö) leikkaa tapahtuman kahtia, ohjelma maalaa tapahtuman kokonaisuudessaan eikä leikkaa puuttuvaa aikaa pois. Tämän korjaamiseksi pitäisi joko `getDurationInBlocks()` metodia parantaa tai sitten koko värjäysprosessi pistää uusiksi.

-Jostakin kumman syystä oikeimmanpuolinen paneeli on aivan liian pieni leveydeltään kun käyttöliittymä tulee esiin. Tätä ongelmaa ei ollut ennen kuin ohjelma käynnistettiin koulun linux-koneella päivää ennen palautusta. Käyttöliittymän konstruktorissa on erikseen määritelty jakopaneelille mittasuhteet (70% viikkopaneelille, loput oikealle). Hiirellä vetämällä paneelin saa siedettävän mittaiseksi, minkä jälkeen se sellaisena myös pysyy (kunnes ohjelma avataan uudestaan).

Plussaa/miinusta:

+Vaihtelevat kestot. Vaikkei sitä projektiin vaadittukaan, se nyt vain jotenkin jäi ärsyttämään etteivät tapahtumat voisi olla eripituisia.

+Varoitus päällekkäisyydestä. Jos kaksi tai useampi tapahtuma osuvat päällekkäin, ohjelma varoittaa asiasta maalamalla niiden sivut punaisiksi.

+Suurimmaksi osaksi helppokäyttöinen GUI. Tekstikenttien täyttäminen on miellyttävämpää, kun voi kirjoittaa päivämääriä vähän lyhyemmällä muodolla.

-Ei täysin helppokäyttöinen GUI: ajanpuutteen vuoksi en ehtinyt toteuttaa varoitusikkunaa joka olisi kysynyt käyttäjältä josko tämä todellakin haluaa poistaa tapahtuman tai ladata uuden kalenterin nykyisen tilalle. Tooltipit eivät olisi olleet haitaksi.

-Maanantai 00:00: Vaikkakin projektiin ei vaadittu eri kestoisia tapahtumia, päätin silti toteuttaa ne, joten jää aika pahasti jäytämään tämä bugi.

-Epäelegantti koodi: huono suunnittelu johti ajoittain aikamoiseen sekamelskaan. Vaikka koodia on nyt siistitty ja kommentoitu on siellä varmasti jäljellä turhaa kloonaamista ja saman metodin turhaa peräkkäiskutsua. Pyöräkin on varmasti keksitty ainakin kahteen otteeseen uusiksi.

Poikkeamat suunnitelmasta eivät olleet järisyttäviä, kun ei ollut suunnitelmakaan mitenkään kiveen kaiverrettu. Kuukausinäköymästä piti alunperin tulla vähän hienomman näköinen, mutta viisas assari neuvoi helpomman

vaihtoehdon. Alunperin oli myös suunnitteilla erillinen ToDoEvent-luokka, mutta siinäkin assari rauhoitti kunnianhimoa.

TCalendar-luokan luominen meni odotettua nopeammin, mikä oli sinänsä hyvä asia koska käyttöliittymän kokoaminen kesti arvattua kauemmin. Nappien toiminta, viikonäkymän maalaminen ja tiedostojen tallennus/avaamis periaatteet oli kaikki uutta, mutta osa osalta se tuli lopulta koottua.

Kaiken kaikkiaan ohjelman puutteet johtuvat enimmäkseen liiallisesta optimismista: käyttöliittymän kokoaminen osoittautui vaikeammaksi kuin aluksi luulin ja erinäiset itse omaksumani lisävaatimukset eivät tehneet asioista helpompia.

Luokkakokoonpano vaikuttaisi omasta mielestäni suhteellisen hyvältä: käyttöliittymän voi vapaasti eristää TCalendar luokasta ilman että kalenteri itsessään menee rikki. Jotkin metodit olisi varmasti kannattaa siirtää erillisiin luokkiin (esim. DateCalc:iin tai TCalParser:iin). TCalGUI-luokka on myös suhteellisen pitkä: suuremman osan sen komponenteista olisi voinut pistää omiin luokkiinsa. Mutta kaiken kaikkiaan onnistunut projekti, jos näin tohtii itse sanoa.

Viitteet:

1. Java API 6.0
2. Internet Calendaring and Scheduling Core Object Specification:
[\[RFC5545\]](#)
3. Oracle:n Java-foorumi
4. The Java Tutorials

Liitteet löytyvät kansioista TCalendar/Documentation.