

Contents

1 Lomapäivä laskuri	1
1.1 Lomapäivien laskemisesta	2
1.2 Ohjelman rakenteesta	2
1.3 Lomapalkan laskemisesta	4
1.4 testien ero sanallisesti	7
1.5 Toteutumattomat osat	7
2 Jatkoa	8
3 Korjauksia	9
3.1 Stringeistä enumeihin	9
3.2 BigDecimal	10
3.3 14pv säännöstä	10
3.4 Työajan tasaaminen	11
3.5 Testien parantelua	11
3.6 jälkihuomioita	12
4 Lähteitä	13

1 Lomapäivä laskuri

Tehtävä vaikuttaa suhteellisen työläältä. Työehtosopimuksien ja lakitekstien tulkitseminen on yleensä kaukana suoraviivaisesta. Ei kai auta kuin kaivaa esiin haluttu TES. TES viittaa heti ensimmäisenä vuosilomalakiin, joten avataan se tähän vierelle. Tehtävänannon mukaan tarkoituksena on laskea vuosilomapäivien määrä ja vuosilomapalkka. Lähdetään purkamaan TES:siä ja vuosilomalakia, jotta saadaan halutut laskukaavat.

1.1 Lomapäivien laskemisesta

- (TES §20.2) Loman määräytymiskausi: 1.4-31.3 (mukaanlukien 1.4 ja 31.3)
- (TES §20.2) Lomapäiviä per lomanmääräytymiskuukausi
 - 2pv jos työsuhte on kestänyt 31.3 alle vuoden
 - 2.5pv jos työsuhte on kestänyt 31.3 vähintään vuoden
- Lomanmäärittelykuukausi = kalenterikuukausi, jolloin töitä on tehty 14pv (tai 35h jos töitä on sopimuksessa alle 14pv/kk. Tulkitaan että tämä pätee esimerkkitapaukseen A)

Mutta mikä lasketaan töiden tekemiseksi? Tätä käsittelee vuosilomalaki §7. Työssäolon veroinen aika on aikaa josta maksetaan palkkaa. Lisäksi on pitkä lista erikoistapauksia, joista testitilanteisiin pätee vanheinpainvapaa, joka on työssäolon veroista ja hoitovapaa, joka ei ole (ja tasaamiseen annetut vapaapäivät tietyin oletuksin?). Molemmat vapaat on TES 19§2 määrännyt palkattomiksi. Tuntityöntekijälle työssäolon veroisiksi tunneiksi lasketaan ne tunnit, jotka työntekijä sopimuksen mukaan ilman poissaoloa olisi ollut työssä. En löytänyt tästä sen enempää ja oletan että tämä on testitapauksessa A 0 tuntia.

1.2 Ohjelman rakenteesta

Lakipykälät on ehkä syytä vielä kerrata ja tarkistaa, mutta uskoisin että voin näillä tiedoilla jo suunnitella ohjelman rakennetta. Testidatan käsittelemistä varten voisi olla hyvä tehdä työhistoria-luokka, joka sisältää taulukon työpäiväolioita, joista jokainen vastaa yhtä riviä annettua dataa.

Työhistorian toteuttaminen oli melko suoraviivaista. Javan LocalDate olio on aika kökkö, mutta päihittää päivämäärän käsittelemisen merkkijonona enkä halua käyttää aikaani oman päivämääräluokan tekemiseen. Tein myös mukavuussyistä metodin työhistorian lukemiselle tiedostosta.

Suunnittelin ohjelman alustavaksi rakenteeksi

- lomapäivä laskuri, joka koostuu seuraavista osista:
 - työsuhde tiedot, joka koostuu osista
 - * työhistoria
 - * sopimuksen tyyppi
 - vuosilomaehdot extends vuosilomaLaki
 - * lomapäivä kertoimien asetus
 - * erityyppisten vapaapäivien käsittely jne

Tarkoitus olisi hoitaa kaikki keskeisimmät lakiin liittyvät asiat lomapäivälaskurissa.

Lomapäivien lukumäärää varten tarvitaan loman määräytymiskuukausien lukumäärä. Aloitetaan tekemällä testit metodille, joka testaa onko annettu kuukausi lomanmääräytymiskuukausi. Jätetään useat erikoistapaukset toistaiseksi pois ja testataan vain onko työpäiviä riittävästi (14 sopimus) tai työtunteja riittävästi (35 sopimus). Tuntien tai päivien laskeminen onnistuu, vielä pitäisi erikseen käsitellä työhön verrannolliset tunnit tai päivät, mutta jatketaan toistaiseksi eteenpäin lomapäivien lukumäärän laskemiseen.

Aloitetaan taas testeistä ja käytetään tällä kertaa tiedostoa apuna. Laskeaan testeihin käsin montako lomapäivää työntekijän pitäisi saada. Olellisesti erilaisia testitapauksia saadaan eri lomavuosista, muuttamalla sopimuksen alkamispäivää ja sopimuksen tyyppiä 14/35. Voisi testata myös vuosilomakertoimien muuttamista, mutta jätetään sekin toistaiseksi pois.

Testit paljastivat kaksi ongelmaa, tiedoston alussa olleet pelkän päivämäärän sisältävät rivit tulkittiin työpäiviksi 14 sopimuksella, mikä liittyy myös toteuttamatta olevaan työpäivien tasauksen käsittelyyn. Toistaiseksi poistin tyhjät rivit.

Toinen testi epäonnistuu, koska toistaiseksi ohjelma käsittelee kaikki merkin-
nät työssäolon veroisena aikana, erityisesti yhdessä kuussa on 11 työpäivää ja
4 hoitovapaata, joita ei pitäisi laskea loman määräytymiseen mukaan. Kor-
jataan tämä lisäämällä se kuukausien testitapauksiin ja muokkaamalla kuukau-
den tarkistusta asianmukaisesti. Testiluokassa voisi olla mahdollisuus pieneen
refaktorointiin tiedostojen alustamisessa mutta jätetään se myöhemmäksi.

Haluanko käsitellä työssäolon veroisuuden ehdot vuosilomalaskurissa? Hel-
pointa olisi ulkoistaa tämä työpäivä-luokalle, mutta silloin joutuisin käsittelemään
lakiasioita useammassa tiedostossa. Ehkä niiden ulkoistaminen yhteen tiedos-
toon ei ole realistista ja metodien nimiä seuraamalla asian tunteva voisi kyllä
tarkistaa noudatetaanko lakia ja tessiä oikein. Työpäivä voisi periaatteessa
kysyä vuosilomaehdot-luokalta onko jokin merkintä sopiva. Tai tätä voisi kysyä
vuosiloma-luokassa. Tehdään toistaiseksi niin, että vuosiloma hoitaa asian vu-
osilomaehdot luokan avulla.

Toistaiseksi vuosilomaehdot tarkastaa onko työpäivä työn veroinen sen pe-
rusteella kuuluuko päivän tyyppi työnveroisien päivien joukkoon. Tämä riittää
toistaiseksi, mutta pitää tarkistaa vielä laista ja tessistä. Testit menevät sentään
jälleen läpi. Jatketaan seuraavaksi vuosiloman palkan laskemiseen ja palataan
hienosäätämään sen mukaan miten aikaa riittää. Lähdetään taas vuosiloma-
palkan testeistä.

1.3 Lomapalkan laskemisesta

- (TES 20§6) vuosilomapäivän palkka = $kk \text{ palkka} / 25 + \text{lisät}$ (ei päde testi-
tilanteissa, sillä kumpikaan ei saa kuukausipalkkaa, toinen vaihtoehto on
”alle 37.5 tuntia tekevät”, mikä ei myöskään käy)
- (TES 20§7) lisät = $(\text{töissäylivuoden} \cdot 12.5\% : 10\%) \text{ vuoden lisistä}$
- Vuosilomalaki 10§: viikko- tai kuukausipalkkaan perustuva vuosiloma-

palkka (käytetään tätä testitapaukseen B?)

- (TES 20§8) alle 37.5 tuntia tekevän lomapalkka = $10/12.5\%$ vuoden ansioista
- (TES 20§9) ansio muodostuu
 - työssäoloajan palkka lukuunottamatta hätä- ja ylityö (ei vaikuta testeihin)
 - poissaoloajan palkka (ei vaikuta testeihin)
 - * Vuosivapaat
 - * sairaustumiset
 - (TES 20§10-11) laskennallinen palkka
 - * sairauspoissaolot (max75pv)
 - * vanhempainvapaa (huomioidaan tämä koodissa!)
 - * lomavapaa
 - * tilapäinen hoitovapaa
 - * ...
 - Laskennallinen palkka = $\text{tuntipalkka} * \text{keskimääräinen viikkotyöaika}$ poissaolon alkamishetkestä lähtien. Keskimääräinen työaika on edeltävän 12 viikon keskimääräinen viikkotyöaika.

Lähdetään rakentamaan lomapalkan laskemista alhaalta ylöspäin. Testitapauksia koskee vain vanhempainvapaa. Vapaat pitäisi ryhmittää yhdeksi palikaksi, jotta kaikilla on sama alkuhetki, eli ensin voitaisiin hajoittaa työvuoden historia eri palikoiksi työpäivän tyyppin mukaan. Kullekin palikalle lasketaan sitten palikan vaikutus ansioihin. Minkä luokan pitäisi olla vastuussa vanhempainvapaa-ajan laskennallisen palkan laskemisesta?

Aloitetaan kirjoittamalla testit ja toteutus työpäivän metodille, joka tarkastaa onko työpäivä annettulla välillä. Käytetään tätä hakemaan työhistoriasta kaikki merkinnät annetulta väliltä. Tätäkin olisi syytä testata, mutta jätetään se myöhemmäksi. Seuraavaksi haluaisin työhistoriaan metodin, joka voi pilkkoa työhistorian yhtenäisiin osiin tyyppin mukaan. Rakennetaan tämäkin testit edellä. Toistaiseksi olen olettanut aina, että työhistoria on järjestetty päivämäärän mukaan. Tätä pitäisi jossain tarkistaa tai järjestää, mutta jätetään se myöhemmäksi.

Käydään seuraavaksi läpi palaset ja lasketaan niistä muodostuva laskennallinen palkka ja työssäoloajan palkka. Tätävarten tarvitaan tiedot tuntipalkasta. Jätin työpäivä-olioon paikan tuntipalkalle, ja toistaiseksi aion vain kovakoodata tuntipalkan logiikan työpaivan muodostajaan. Tämä vaati myös pientä päivitystä työpäivän testeihin.

Laskennallinen palkka on hieman monimutkaisempi, niin tehdään sitä varten oma metodi ja sille pari testiä. Keskiarvon laskeminen viimeisen 12 viikon aikana vaikuttaa vähän työläältä käsin, niin tarkastetaan tulos myöhemmin. Vastaus on oikean suuruinen.

Laskennallisen palkan pohjalta saadaan helposti lomapalkka, tehdään sille testit valmiiksi, mutta jätetään oikean vastauksen laskeminen vielä myöhemmäksi. Lopputulos on oikean suuntainen. Tehtävään annettu kaksi päivää lähestyy myös uhkaavaa vauhtia, niin tehdään tässä vaiheessa hieman siivoamista, tarkistetaan lakipykälät ja lisätään ne kommentteihin sekä luetaan tehtävänanto vielä uudelleen.

Tarkastin koodin tehtävänannon ohjeiden mukaan sekä lisäsin ja tarkastin lakiviitteitä. Joitakin kohtia en ehtinyt toteuttaa. Lasketaan vielä tarkastusarvot käsin ja verrataan ohjelman antamiin tuloksiin. Laskin vastauksen yhdelle laskennallisen palkan testille, mutta aika alkaa loppumaan niin siirry-

tään eteenpäin. Kirjoitetaan ohjelmalle nopea dokumentaatio ja käyttöohje. Paljoa toiminnallisuutta ohjelmalla ei tässä vaiheessa ole.

1.4 testien ero sanallisesti

Testitapaus A on oletettu kuuluvan joukkoon Tes 20§2 ”työntekijän työskentelee työsopimuksen mukaan alle 14 päivää kuukaudessa”. Karkeasti ottaen tästä seuraa, että lomapäiviä karttuu kaikilta kuukauksilta, joissa töitä on tehty vähintään 35h. Testitapaus B ei kuulu tähän joukkoon, jolloin lomapäiviä kertyy kuukauksista, joissa töitä on tehty vähintään 14 päivää.

Lomapalkan osalta tes ja laki oli vielä epäselvempi ja päädyin tulkitsemaan niitä siten, että molemmat kuuluvat tessin joukkoon §20.8 ”Alle 37,5 tuntia tekevät” ja niihin sovelletaan pykälää 8-11. B kohdalle perusteltu vaihtoehto olisi ollut myös TES kohta 20§6, mutta kiinteän kuukausipalkan puuttuminen ja tuntipalkan vaihtuminen kesken lomanmääräytymis vuoden tuntui sulkevan tämän vaihtoehdon pois. Ei sillä että tapaus B kuuluisi myöskään sen kohdan alle johon sen laitoin, mutta en löytänyt parempaakaan ratkaisua annetussa ajassa.

Oleellisin ero lomapalkan laskemisen suhteen on se, kuinka laskennallinen palkka lasketaan siltä ajalta, jolloin työntekijä oli vanhempainvapaalla. Kiinteillä viikkotunneilla laskennallisessa palkassa käytetään annetuista viikkotunneista saatavaa päivän tuntimäärää. Tapauksessa A laskennallisen palkan oletettu päivän tuntimäärä taas on edellisen 12 viikon työtuntien keskiarvo.

1.5 Toteutumattomat osat

Isoin toteutumatta jäänyt osa on laskettujen tuloksien pohjalta muodostettava vuosilomaPalkkalaskelma. Lisäksi ”työajan tasaamiseksi annettuja vapaapäiviä” ei käsitellä koodissa mitenkään, vaikka näistä oli jotain mainintaa

vuosilomalaissa. Perusteltua olisi myös lisätä vuosilomapalkan laskeminen suoraan kuukausipalkasta TES 20§6 mukaisesti. Tehtävänannossa määrätty palkka on toistaiseksi koodattu TyoPaiva-olion sisälle ja tähän olisi syytä keksiä jotain parempaa. TyoPaiva-luokan ja muutaman muun nimiä voisi myös vaihtaa kuvaavemmiksi ja koodia refaktoroida. Työhistorian käsittely Vuosilomalaskurissa on myös huono valinta ja se pitäisi siirtää TyosuhdeTiedot rajapinnan taakse. Testejä voisi lisätä useaan paikkaan.

2 Jatkoa

su 31.7 10:00-11:00 Lisätään ohjelmaan toteutumatta jääneitä osia. Aloitetaan laskemalla oikeat vastaukset ja katsomalla meneekö testit läpi. Työhistorian paloittelusta paljastui bugi, jossa ensimmäisen palasen ensimmäinen työpäivä lisättiin kahdesti. Lisäsin tähän tarkemmat testit ja korjasin ongelman. Testeistä paljastui toinen bugi, jossa olin muuttanut lomapalkkakertoimen sijaan lomapäiväkerrointa. Näiden korjauksien jälkeen tulos vastaa käsin Excelin avulla laskettua lomapalkkaa.

11:15-12:00 Toteutetaan seuraavaksi vuosilomapalkkalaskelma. Toistaiseksi olen laskenut molempien tapauksien lomapalkan prosenttiperusteisena. Lisäsin vuosilomalaskuriin muutamia rivejä päivittävät palkkalaskelman kenttiä laskutuloksilla. Lisäsin myös palkkalaskelmaan toString metodin, jolla se voi tulostaa palkkalaskelmaan tarvittavat tiedot. Testataan tätä vuosilomalaskurin main metodissa ja tehdään lopputuloksesta sopivasti nimetty tekstitiedosto. Vaikuttaisi toimivan, mutta muutetaan vielä liukulukujen tulostusta hieman sievemmäksi.

12:30-13:15 Piirretään mind map, testataan vuosilomapalkkalaskelma ja kerataan tehtävänanto. Työajan tasaamiseen annettujen vapaapäivien käsittely ei muuttaisi mitään ja minulle jäi myös hieman epäselväksi mitkä päivät näihin

luetaan, joten jätetään se edelleen toteuttamatta.

13:15-14:00 Siirretään työhistorian käsittely pois vuosilomalaskurista rajapinnan taakse ja korjataan tästä rikkoutuvat testit.

14:00-15:00 Lisätään vielä kuukausi/viikkopalkkaan perustuva lomapalkan laskeminen sekä sen testit ja tulostus.

15:00-15:30 Ratkaisu vaikuttaisi nyt olevan tehtävänannon mukainen. Käydään koodi vielä läpi ja siistitään tarpeen mukaan (esim metodit johonkin järkevään järjestykseen). Päivitetään myös käyttöohje.

ma 1.8 8:30-9:00 Käydään vielä lopputulos pikaisesti läpi ennen palauttamista.

3 Korjauksia

3.1 Stringeistä enumeihin

Korvataan string tyypit enum-tyypeillä. Kokeilin tätä jo aikaisemmin, mutta se ei toiminut niin kuin halusin, joten aloitetaan tutustumalla javan enumiin vähän paremmin. Laitetaan merkintöjen tyypit omaan enum tiedostoon, joka voisi joskus vastata myös eri merkintöihin liittyvän logiikan käsittelystä. Lisätään toistaiseksi ainoastaan niiden merkintätyyppien käsittely mitä esimerkkidatasta löytyy. Refaktoroidaan loppu koodista käyttämään näitä enum-tyyppejä aloittaen työpäivämerkinnästä. Muokataan myös työpäivän parsiminen merkkijonosta hieman nätimmäksi switch-case:n avulla. Testit menevät refaktoroinnin jälkeen edelleen läpi. Olen hieman yllättynyt, toisaalta testit eivät ottaneet kantaa siihen missä muodossa mikäkin kenttä on tallennettu. Lisätään kuitenkin testit vielä erikseen merkinnän tyypille.

Työpäivämerkinnän muuttaminen rikkoi työhistorian, joten jatketaan refaktorointia korjaamalla se ja sen testit muutaman rivin muutoksella. Loputkin

korjauksista oli vain muutamia rivejä ja testit pysyivät ehjinä.

3.2 BigDecimal

Tutustutaan seuraavaksi bigdecimaliin. Pikaisella lukemisella vaikuttaa siltä, ettei tämäkään pysty käsittelemään tarkasti esimerkiksi lukua $1/3$? Esimerkiksi 12 vk keskimääräisen päiväpalkan laskemisessa saadaan helposti päättymätön desimaaliluku, jota kumpikaan bigdecimal tai double ei pysty esittämään tarkasti. Laskut voitaisiin tässä tapauksessa tehdä tarkasti rationaaliluvuilla.

BigDecimal tuottaa ongelmia laskennallisessa palkassa, jossa saadaan päättymättömiä desimaalilukuja. Jaettaessa pitää antaa haluttu tarkkuus jakolaskun tulokselle ja etenkin jos lukua myöhemmin kerrotaan jollain suurella luvulla, tarkkuus heikkenee. Lisätään tarkuutta reilusti ja pyöristetään tarkistaessa lopputulokset senttien tarkkuuteen. Testit menevät jälleen läpi, mutta en usko että mahdollisista pienistä lasku- ja pyöristysvirheistä olisi kokonaan päästy eroon. Vähän turhan työlään uudelleenkirjoittelun jälkeen sitä arvostaa entistä enemmän C++:n kaltaista operaattorien kuormittamista.

3.3 14pv säännöstä

Kaupalanalan tessissä ei mainita vuosilomalain 11§ keskipäiväpalkkaan perustuvaa vuosipalkkaa, vaan vaihtoehdot ovat kuukausipalkka/25 ja ”alle 37.5 tuntia”, joka sisältää tuntipalkkaiset. 14pvKuukaudessa on kaikkialla määritelty sen mukaan onko ”sopimuksen mukaisesti työssä vähintään 14 pv kuukaudessa”. Hieman epävirallisemmilta sivuilta löytyy tarkennus ”ratkaisevaa on se, millainen työntekijän työaika on käytännössä vakiintuneesti ollut,” mikä ei sekään vielä kerro oikein mitään. Lisätään ohjelmaan tunnistus sen mukaan onko työpäivien lukumäärä kuukaudessa ollut keskimäärin yli 14.

3.4 Työajan tasaaminen

Lisätään käsittely työajan tasaamiseksi annetuille vapaapäiville. Tämän ehdot olivat jokseenkin monimutkaisemmat kuin aiemmilla vapailla. Koska päätelysäännöt perustuvat vuosilomalakiin, tehdään myös toteutus tähän luokkaan, jota kätevästi kutsutaan jo validoimaan onko kukin merkintä työssäolon veroinen.

Toistaiseksi kutsun validointimetodia päivä kerrallaan, mutta koska ehto riippuu siitä onko tasauspäiviä annettu kuuden putkena, on ehkä parempi antaa vuosilomalain tutkia kerralla koko kuukauden merkinnät. Tästä tulee taas suhteellisen mutkikas, joten aloitetaan testeillä. Toteutus olikin odotettua simppelempi, mutta tein joitain oletuksia kuten että jokaiselta arkipäivältä on merkintä ja jos siinä ei ole muuta kuin päivämäärä, niin se on tasauspäivä. Oletin myös, että toinen testitapauksista on nollatuntisopimus, jolloin työssäolon veroisia tunteja ei tule ollenkaan työssäolon veroisilta poissaoloilta. Muussa tapauksessa pitäisi lisätä jonkinlainen keskityöajan käsittely ja siihen perustuva työnveroitusten tuntien lisääminen työnveroisilta poissaoloilta.

3.5 Testien parantelua

Siirrytään seuraavaksi kattavempien testien suunnitteluun ja muokataan ohjelmaa sitten niiden pohjalta. Lisätään korjausehdotuksissa mainitut testit yksitellen ja muokataan kunkin kohdalla koodia siten että testit menevät läpi. Tulkitsen edelleen molempien testitapauksien kuuluvan tessin perusteella prosenttiperusteisen lomapalkan piiriin.

Aloitin tekemään myös pientä siivoamista ja refaktorointia. Halusin käsitellä työsuhdetietoja enemmän vakiona kuin tilakoneena. Testien korjaaminen oli hiekkamutkikasta juurikin koska työsuhdetietojen tilaa muutettiin milloin missäkin. Huomasin myös, että testejä olisi syytä kommentoida vielä lisää. Ennen muita suuria uudistuksia on jo korkea aika ottaa käyttöön versiohallinnan

työkalut.

Muokataan koodi vastaamaan mindmappia. Lisätään lomapäivien laskuun käsittely työssäolon veroisille tunneille työssäolon veroisien vapaapäivien osalta. Oletetaan, että työntekijä työskentelee sopimuksen mukaan vapaapäivinä 0 tuntia, jos sopimuksessa ei ole sovittu tuntimäärää. Lomapäivien laskeminen on nyt kunnossa.

Lisätään myös toteutus tuntipalkkaperusteiselle lomapalkalle. Tämä oli oikeastaan helpompaa kuin odotin. Luodaan testitapaus C, jossa lasketaan testitapausten B palkka lisäoletuksella, että käytetään vuosilomalain tuntipalkkaperusteista lomapalkkaa. Testien teko oli suoraviivaista ja useimmat tarkastusarvoista ja laskukaavoista oli jo laskettu aiemmin. Testataan vastaavalla tavalla testitapaus B vuosilomalain viikkopalkkaan perustuvalla laskutavalla. Nimetään tämä testitapaukseksi D. Todellisuudessa voisi testata vielä monia muita kombinaatioita, mutta lopetetaan tähän.

Käsitellään sitten vielä lomaraha ja sen testit. Lomarahan maksuedellytyksen määrittämiseen tehtävänanto ei riitä, joten tehdään sille pelkkä tynkä toteutus. Lomarahan suuruus on suoraviivaisesti puolet lomapalkasta.

3.6 jälkihuomioita

Vuosilomapalkkalaskelmassa on nyt valtava määrä pelkkiä get/set metodeja. En tiedä olisiko järkevämpää muuttaa kentät julkisiksi tai korvata kaikki get-set metodit parilla switch-case lauseella ja käsitellä kenttien nimiä enum tyyppisinä. Samanlaista get-set koodia on monessa muussakin luokassa. Jollakin ohjelmointikurssilla neuvottiin laittamaan kentät lähtökohtaisesti privaateiksi ja käyttämään get/set metodeja enkä ole vain kokenut tarvetta kyseenalaistaa tätä käytäntöä.

Jotkin asiat olisi syytä erottaa omaan luokkaansa, etenkin koska ohjelman

laajentaminen toisi niihin entistä enemmän sisältöä. Esimerkiksi laskennalliseen palkkaan oikeuttavien vapaapäivien laskeminen. Etsitään ja poistetaan lopuksi kutsuhierarkian avulla muutamia turhia metodeja.

4 Lähteitä

https://www.pam.fi/media/tes_kauppa_web_pam.pdf

<https://www.finlex.fi/fi/laki/ajantasa/2005/20050162#L3>