



TERMINAALI 37V
JUHILAPAINOS

Sisällysluettelo

Puheenjohtajan palsta

Hyvän päälle on hyvä rakentaa 4

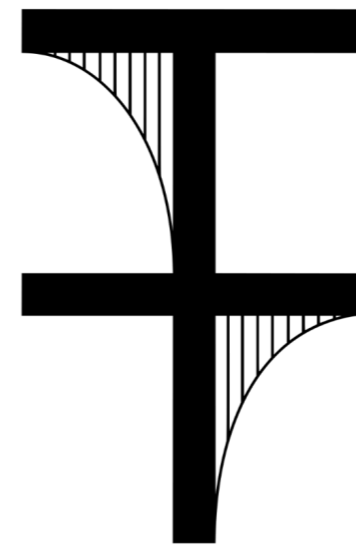
Wanhoiden Puheenjohtajien Terveisiä 6

Silk Song Mayhem, Aleksanteri Seppä 10

GNU, Jimi Gustafsson 12

Poikkeustilat ja niiden käsittely, Jiska Parrila 13

Sarjakuvanurkkaus, Leevi Astikainen 18
Fuksit vaan nuorenee



Julius Kuruheimo
päätoimittaja

Vasta toisen vuoden OTiTLaisena moni asia vieläkin on uutta ja ihmeellistä killassa. Rakastan uusien asioiden oppimista ja kokeilemista, ja olenkin Tiedotusministerinä päässyt tekemään ja toteuttamaan itseäni sopivan rennossa mutta vaativassa ympäristössä. Niille jotka eivät vielä tiedä, niin pittelette käsissänne jotain aivan uutta, jotakin, joka oli vielä keväällä vain konsepti, hauska idea joka nousi ilmoille, en edes muista kenen päästä: Tehdään vakava Terminaali! Lähdimme innokkaana toteuttamaan ideaa ja tällainen tästä nyt tuli. Lämmin kiitos kaikille juttuja tehneille ja kanssa-intoilijoille.

Yritimme tuoda esille paljon erilaisia näkökulmia killasta ja tietotekniikasta yleisesti. Taittoprosessissa otin oppia mm. TEK:n lehdestä. Lopputuloksena olevaa tuotetta pääsetekin nyt kuluttamaan Te, rakkaat lukijat. Toivottavasti nautitte tästä ai-nutlaatuisesta tuotoksesta vähintään yhtä paljon kuin vuosijuhlistamme muutenkin.

-Julius "molokku" Kuruheimo

Terminaali

Oulun Tietoteekkarit ry:n kiltalehti
Pää-äänenkannattaja jo vuodesta 1988
38. vuosikerta
Vuosijuhlapainos
20 sivua
Painopaikka Monisto

Oulun Tietoteekkarit ry/Terminaali
PL4500 90014 Oulun Yliopisto

Lehden teossa mukana

Toimittajat:

Aleksanteri Seppä
Verner Pudas
Jimi Gustafsson
Leevi Astikainen

Kansi

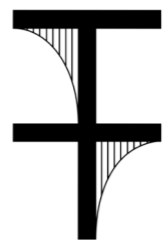
Erki Kuivonen
Taitto ja ulkoasu
Julius Kuruheimo
Jimi Gustafsson (GNU)
Jiska Parrila (Poikkeustilat ja niiden käsittely)

Kiitokset

Pekka Koponen
Visa Seppä
Jiska Parrila



Salla Rissanen
Puheenjohtaja



Hyvän päälle on hyvä rakentaa

Maailma ja meidän kiltamme on tänä päivänä hyvin erilainen kuin mitä ne olivat, kun OTiT:ia perustettiin. Mutta maailma muuttuu ja meidän pitää muuttua mukana, eikö? Osittain muutos on tullut myös pakon edessä – mm. koronavuodet tappoivat paljon perinteitä, mutta toivat myös uusia tuulia mukanaan. Osa perinteistä kulkee edelleen matkassa mukana, mutta osa on jäänyt hyllylle pölyttymään. Joka tapauksessa uusia perinteitä pyritään luomaan ja kehittämään meidän toimintaamme tekijöidensä näköiseksi.

Tämä vuosi on ollut täynnä muutoksia ja olen iloinen siitä, että meillä on porukkaa, jotka uskaltavat yrittää uusia asioita. Uudet asiat saattavat tuntua joskus oudoilta, eikä tiedetä, miten niihin pitäisi suhtautua. Ei tiedetä, onko idea hyvä vai huono, mutta tähän selviää vain kokeilemalla. Yksi näistä uusista konsepteista oli tihentää vuosijuhlien järjestämisä.

Mutta onhan meillä vietetty ”välivuosisjuhlia” aiemminkin? No onhan toki, 32-vuosijuhlat bittien kunniaksi olivat viimeisimmät. Ja tästä idea syntyiikin, että miksi ei voitaisi juhlia meidän syntymäämme useammin kuin viiden vuoden välein. Seuraavat bittijuhlat olisivatkin vasta 64-vuotiaana eli sinne on vielä matkaa. Kuten vuosijuhlilla kuuluukin, haluan tässä palstassa hieman muistella viime vuosien tapahtumia.

Edellisen kerran pääsimme juhlimaan vuonna '23 OTiT:n 35-vuotis vuosijuhlia ja heti seuraavana keväänä '24 järjestimme Titeenien taistot Oulussa. Hieman lyhyellä aikavälillä, mutta ei mennyt päällekkäin! Molemmat olivat oikein onnistuneita tapahtumia. Vuonna '24 lanseerattiin myös uusi alumnitapahtuma JOOO (Jotkut OTiTlaiset Oppivat Omiltaan), missä alumnit kertovat paneelimuotoisesti opiskelulle jäsenille omista opinnoistaan, työnhausta ja työelämästä riippuen hieman siitä, mitä kysymyksiä järjestävä taho heille on esittänyt ja mitä opiskelijat haluavat tietää.

Myös Fragment LAN -tapahtuma elvytettiin TST-kiltojen kanssa yhteistyössä viime vuonna ja siitä tulikin heti hitti. Viikonlopun kestäväillä laneilla oli pelaajia vuorokauden ympäri hakemassa itselleen kunniaa erilaisten turnausten muodossa ja muutenkin pitämässä hauskaa. Myös vanhaa kehitettiin lisäämällä OTiT OTiT osakilpailuja vanhojen perinteisten kilpailujen rinnalle. Mm. mölkky- ja beerbongturnaukset ovat olleet näissä hittituotteita. Nykyisin osakilpailun voitosta saa myös muistokseen haalarimerkin nopeasti tyhjenevän pullon lisäksi.

Kaikenlaisten tapahtumien lisäksi on meillä hallinnollisestikin rakenne muuttunut. Vuonna '23 vaalikokous päätti lisätä hallitukseen uuden pestin Liikunta-, urheilu- ja nuorisoministeri, jonka vastuulla on mm. kaikki vapaa-ajan toiminta. Vuoden '24 vaalikokouksessa päätettiin kokeilla toimikuntien toimintaa tukemaan toimihenkilöiden ja hallituksen toimintaa. Lisäksi tänä syksynä on testaus-



OTiT 35v - Joona Syrjäkoski

mielessä lanseerattu OTiT:n oma kerhotoiminta ja ainakin tähän mennessä koikeilu on herättänyt kiinnostusta.

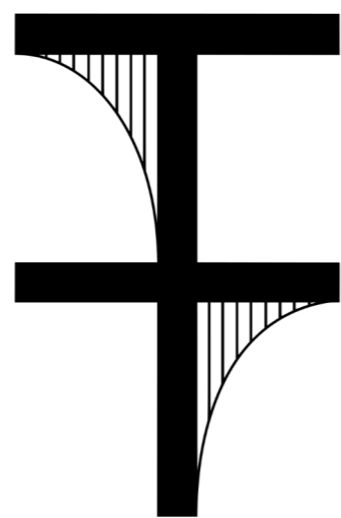
Monenlaisia uusia asioita on siis tullut toimintaamme mukaan, enkä tässä ole maininnut kuin vain murto-osan näistä muutoksista. Sanomattakin on selvää, ettei kiltatoiminta ihan vielä ole kuolemassa, kun uusia juttuja keksitään ja toteutetaan. Mutta vaikka uusia asioita lisätään, niin muistetaan myös kunnioittaa meidän perinteitämme ja arvostaa matkaa, jonka kiltamme on tähän mennessä kulkenut. Ei ole itsestään selvää, että toimintaa edelleen harjoitetaan ja uusia toimijoita saadaan mukaan vanhojen toimijoiden rinnalle. Iso kiitos tästä on meidän juurissamme, jotka perustajamme ovat 37 vuotta sitten meille luoneet.

Hyvän päälle on hyvä rakentaa.

On vuosi '25, kauan on aikaa siis. Siitä kun viimeksi juhlimme OTiT:n synttäreitä. No ei nyt ehkä ihan niin kauan aikaa sitten kuitenkaan. Suomalaisia hittibiisejä tuntevat saattoivat jo luulla, että tästä se lähtee, nimittäin vanha kunnon Eppu Normaalin Vuonna '85 hieman modifioituna. Mutta ehkä tällä palstalla on jotain muutakin ideaa kuin vain kopioida erinomaisia lyriikoita.

37 vuotta takana ylä- ja alamäkiä, huikeita tapahtumia ja huippua porukkaa. Itse olen saanut olla osa tätä tietotekniikan eliittiä vasta 3 vuotta, mutta en vaihtaisi näitä kolmea vuotta mihinkään. Olen saanut kunnian olla osa meidän aktiiviporukkaamme koko opiskelu-urani ja päässyt sitäkin kautta osallistumaan monenmoisiin juttuihin, joita en osannut edes kuvitella ennen

Wanhoiden Puheenjohtajien Terveiset



”Tietotekniikkaopinnot olivat aikanaan todellinen jackpot”

OTiTin perustaminen 1988 oli upeaa aikaa. Saimme nuorelle ja viriilille tietoteekkarilaumalle oman ja kadehditun identiteetin. Tuolta ajalta tuli paljon ystäviä, mahtavia muistoja ja sosiaalisia taitoja, jotka ovat kantaneet tähän päivään.

Tein dippatyön Nokia Mobile Phonesille 1990-91. SIM-korttisofta meni lopulta 500 miljoonaan taskuun. Sitten maailmalle Singaporeen, Kiinaan, Tsekkisiin, Sveitsiin MBA:ta opiskelemaan kunnes Nokian alustastrategiajohtajapestin jälkeen omaan firmaan.

Spinversestä tuli 20 vuodessa Pohjoismaiden suurin EU-rahoituskonsultti, joka on kerännyt yli 4 miljardia euroa asiakkailleen, laajasti tietotekniikkaa kehittäviin hankkeisiin. Firman myynnin jälkeen 2022 olen keskittynyt rahoittamaan digitaalista oppimista, nyt 7 koulutus-alanyrityksessä sijoittajana.



Pekka Koponen
OTiT 1. Puheenjohtaja 1988-89
perustajajäsen

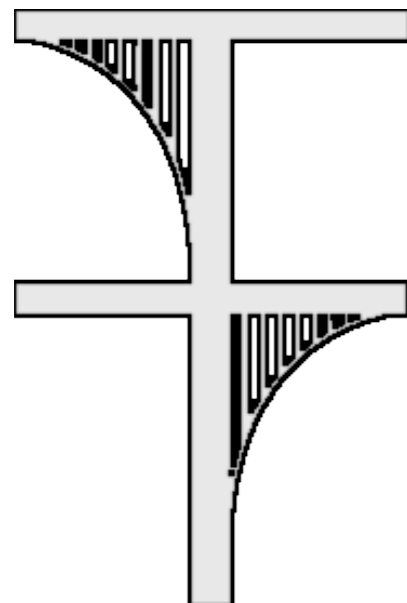
Lisäksi digitransformaatiota-kin kehittävä työelämäprofessori Vaasassa ja puheenjohtajana kaikkien korkeakoulujen omistamassa www.opin.fi-digioppimisalustassa.

Tietotekniikkaopinnot olivat aikanaan todellinen jackpot, joka mahdollisti uskomattoman uran ympäri maailmaa. NykYTEKKARI miettii varmaan miten AI vaikuttaa tulevaisuuden työhön ja työpaikkoihin. Ei huolta! AI-koodauksen osaajalle aukeaa yhä vaativampia tehtäviä, muistakaa opiskella myös bisnestä että ymmärrätte paremmin mitä asiakas tarvitsee.



Verner "napalmi" Pudas
OTiT Puheenjohtaja 2024

Propagandaministeri 2021, lisäksi lukuisia toimaripestejä



The illusion of Free will

Tietotekniikassa parasta on vapaus valita. Ala on erittäin laaja ja täynnä mahdollisuuksia. Kaiken voi tehdä uskomattoman monella tavalla ja lopputulos on silti aina paskaa loppukäyttäjän mielestä!

<i>Sisällysluettelo</i>	
Pääkirjoitus	s.2
Sisällysluettelo	s.vittu arvaa
Pj-palsta	s.4
Banaaneista	s.5-6
Maskianen tulee taas!	s.7
Lord Of The Rings ihanne kumppani	s.8-9
Irkistatistiikkaa	s.11-13
Hallitus_epähaastattelu	s.14-19
Miten tehdään terminaali juttu?	s. 20-22
Aika	s. 23-25
Terminaali Oulun Tietoteekkarit ry:n kilttalehti	Päätoimittaja Verner Pudas

Perinteikkäämpää Terminaalin sisältöä. Kuvat ovat Vernerin propagandaministerikaudelta, Terminaalista 1/21



Syvempi tietotekniikan osaaminen auttaa ymmärtämään maailmaa

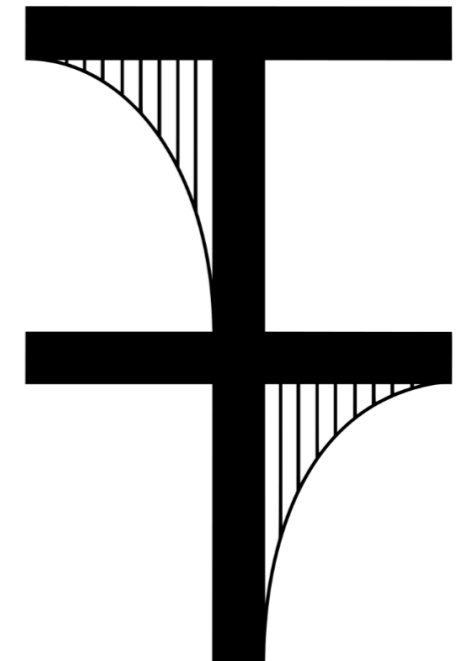
Tietotekniikassa alkujaan oli parasta sulautetut järjestelmät. Kyseinen suuntautuminen on edelleen huippua, mutta nykyään parasta on, kun tietotekniikkaa on kaikkialla ja se koskettaa nykyään lähes kaikkea mitä voi tehdä, niin syvempi alan ymmärtäminen auttaa ymmärtämään maailmaa kokonaisuutena paljon paremmin.

Yliopisto jakautuu todellisuudessa kahteen osaan: akateeminen koulutus ja sen ympärille rakentunut yhteisö. Molemmat kannattaa ottaa haltuun, välillä voi keskittyä yhteen ja välillä toiseen, mutta ikinä ei kannata täysin unohtaa näistä kumpaakaan. Se yhteisö voi olla sinulle esimerkiksi kiltta, harrastejärjestö ja/tai ylioppilaskunta. Kaikkea kannattaa kokeilla ja antaa mahdollisuus. Mene itse kokemaan ja katsomaan!



Visa "MegaTron" Seppänen

Puheenjohtaja (2015), Kulttuuriministeri (2017), Työvoimaministeri (2014), lisäksi useita toimaripestejä



Silk Song Mayhem

Seitsemän vuoden odotuksen jälkeen, 4.9.2025, Team Cherry julkaisi Hollow Knight: Silksong julkaistiin. Artikkelia kirjoittaessa olen vähän kerennyt peliä pelata, mutta tämä ei ole arvostelu pelistä, vaan lyhyt reportaasi draamasta sen ympärillä.

Hollow Knight:en Kickstarter kampanjassa yhtenä täyttyneestä rahoitustavoitteena oli peliin lisättävä toinen hahmo, ja tämän hahmon oli tarkoitus olla Hornet. Kuitenkin Team Cherry:n kehittäjät päättivät alkaa kokkaamaan, jolloin Hornet sai oman pelinsä, josta minä tässä kirjoitan.

Silksong:in kehitys on ollut tiedossa jo vuodesta 2019, mutta julkaisupäivä tuli meidän tietoon kaksi viikkoa ennen pelin julkaisua, 21.8.2025. Tämä aiheutti mielenkiintoisen ketjureaktion, kun Little Witch in the Woods lykättiin Silksongin julkaisun tieltä. Sen jälkeen useat indie pelit ovat siirtäneet julkaisuajankohtaansa pois Silksongin alta, jotta heidän pelinsä eivät hukkuisi julkaisun alle.

Julkaisupäivänä pelijakelualustat hajosivat. Esimerkiksi Humble Bundlestä ja Steamista pelin lisenssiavaimet loppuivat kesken, joka on hyvin huvittava yhteensattuma.



Peli on saanut kritiikkiä kiinalaisilta pelaajilta, sillä yksinkertaistetun kiinan lokalisaatio petti. Dual Shockersin mukaan pelaajat kritisoivat pelin lukevan kuin fantasia kevyt novelli, joka ei sovi pelin kontekstiin.

Pelikehittäjiltä on myös tullut kritiikkiä Silksongin hinnasta. Peli on hyvin edullinen, vain vajaa 20€. IGN raportoi esimerkiksi Unbeatable pelin tuottaja (director?) RJ Lake avautui BlueSky:ssä, että pelin pitäisi maksaa tuplat, 40€, argumentoiden, että 20€ arvoriima iskeytyisi liian korkealle, eikä me hyväksyttäisi 20€ pienemmän skaalan tai laadun pelejä. Tähän sanaan yh-

Kuva: Joonas Syrjäkoski @joona_sk



Aleksanteri "dolmio" Seppä
Opetusministeri 2025

tyi muitakin indie-kehittäjiä niin BlueSkyssä kuin Twitterissä. Minusta argumentti ei toimi, sillä kuten Lake myös mainitsi, he voivat hinnoitella pelinsä miten haluavat. Silksongia kehitti kolmen henkilön tiimi.

Amista käydessäni opin, ettei omaa työtä saisi myydä liian halvalla, sillä se puski muidenkin asioiden hintatasoa alaspäin. Esimerkiksi tietokoneen huoltotyötä tai vaikka taidetilauksia taiteilijalta. Tämä voi olla jopa yhteisön epävirallista kartellitoimintaa, mutta peruseriaatteella järki on siinä, että jokainen ammattilainen saisi työstään arvoista palkkaa. Tämä indiekehittäjien argumentointi pohjaa tähän periaatteeseen, jota voidaan ajatella sosiaalisena kartellina, tai Walmartin tapaisen monopoliasemaa hakevaan tukahdushinnoitteluun (joka ei selkeästikään toimi tässä kontekstissa).

Minusta on kuitenkin hyvä pohtia, minkä hintaisina pelejä arvottaa. Suuremmat pelistudiot haluavat nostaa pelien hintoja pilviin. 60€ oli suhteellisen pitkään vakio, mutta nyt hintaa ollaan nostettu 80€. GTA VI:n huhutaan kantavan 100€ hintalappua. Riittääkö pelaajilla rahaa ostaa näitä pelejä, joilla on suuret budjetit ja hintalappu? Joillain kamelinselkä on jo mennyt rikki.

-dolmio



Kun puhutaan Linuxista, monen mieleen tulee suomalainen Linus Torvalds, joka kehitti kuuluisan Linux-ytimen 1991. Torvaldsin nimi on tuttu teknologiapiireissä ja usein liitetään koko Linux-käyttöjärjestelmään. Mutta kaikki eivät tiedä, että Linux ei olisi käyttökelpoinen ilman Richard Stallmania ja hänen aloittamaansa GNU-projektia.

Linux on käyttöjärjestelmän ydin, eli ohjelma, joka hallinnoi laitteiston resursseja ja toimii rajapintana sovellusten ja laitteiston välillä. Se ei kuitenkaan yksinään tarjoa käyttäjälle toimivaa järjestelmää. Käyttöjärjestelmä tarvitsee ympärilleen kokonaisuuden: komentorivityökaluja, kääntäjiä, kirjastojen hallintaa, tiedostojärjestelmän tukea ja paljon muuta. Juuri nämä osat tulevat GNU-projektista.



GNU-projekti käynnistettiin vuonna 1983 tavoitteena rakentaa vapaa Unix-yhteensopiva käyttöjärjestelmä. Kun Linux-ydin julkaistiin vuonna 1991, se yhdistettiin GNU:n ohjelmistokomponentteihin, ja näin syntyi GNU/Linux. Kokonainen käyttöjärjestelmä, joka on nykyään käytössä lukemattomissa jakeluissa.

Vaikka teknisesti kyse on GNU/Linux-järjestelmästä, nimi "Linux" on vakiintunut käyttöön. Suurin osa käyttäjistä ei tiedä, että heidän käyttämänsä jakelu, olipa kyseessä Debian, Fedora tai Arch -pohjainen jakelu, sisältää Linux-ytimen lisäksi laajan valikoiman GNU-ohjelmistoa. GNU muodostaa suurimman osan siitä, mitä käyttäjä näkee ja käyttää.

GNU:n vaikutus ulottuu myös lisensointiin. Sen kehittämä GPL-lisenssi (General Public License) on yksi avoimen lähdekoodin kulmakivistä, ja se on mahdollistanut ohjelmistojen vapaan jakamisen ja muokkaamisen. Ilman GNU:ta ei olisi syntynyt nykyistä avoimen lähdekoodin ekosysteemiä, jossa Linux on keskeinen osa.

Linux on siis ydin, mutta käyttöjärjestelmä on kokonaisuus. GNU tekee siitä käyttökelpoisen.

"Se, mihin viittaat Linuxina, on itse asiassa GNU/Linux, tai kuten olen viime aikoina alkanut kutsua sitä, GNU plus Linux. Linux ei ole käyttöjärjestelmä itsessään, vaan pikemminkin yksi vapaa osa täysin toimivaa GNU-järjestelmää, josta on hyötyä GNU-korelibrien, komentorivityökalujen ja tärkeiden järjestelmäkomponenttien avulla, jotka muodostavat POSIXin määrittelemän täydellisen käyttöjärjestelmän. Monet tietokoneen käyttäjät käyttävät päivittäin muunneltua versiota GNU-järjestelmästä huomaamattaan. Erään erikoisen käänteiden vuoksi nykyään laajalti käytettyä GNU-versiota kutsutaan usein Linuxiksi, eivätkä monet sen käyttäjät tiedä, että se on pohjimmiltaan GNU-projektin kehittämä GNU-järjestelmä. Linux on todella olemassa, ja nämä ihmiset käyttävät sitä, mutta se on vain osa heidän käyttämäänsä järjestelmää. Linux on järjestelmän ydin: ohjelma, joka jakaa koneen resurssit muille ohjelmille, joita käytät. Ydin on olennainen osa käyttöjärjestelmää, mutta hyödytön yksinään; se voi toimia vain täydellisen käyttöjärjestelmän yhteydessä. Linuxia käytetään yleensä yhdessä GNU-käyttöjärjestelmän kanssa: koko järjestelmä on periaatteessa GNU, johon on lisätty Linux, eli GNU/Linux. Kaikki niin sanotut Linux-jakelut ovat oikeastaan GNU/Linux-jakeluita!"

Poikkeustilat ja niiden käsittely

Osa 1

Arvoisa kiltalehden lukija. Olette päätnyt artikkelisarjan ääreen, jossa pureudutaan poikkeustilojen syvimpään olemukseen ohjelmointikieleen tai -tyyliin katsomatta. Sarja saattaa sisältää ja saada aikaan hämmennystä, kummastelua, outoja käsitteitä ja asiasisältöä. Erityisesti fukseissa. Teitä on varoitettu, siispä nauttikaa.

Tässä aloitusosassa esitellään aiheeseen liittyvää termistöä ja aloitetaan aiheen käsittely poikkeustilojen välitysmekanismeilla.

Viat, virheet ja poikkeustilat

Poikkeustilojen käsittely on laaja aihe. Sen opettaminen on puutteellista ja usein laiminlyödään täysin. Asian ymmärtämistä sotkee epäyhtenäinen termistö, jossa termejä virhe ja poikkeus käytetään vaihdannaisesti. Poikkeustilojen oikeaoppinen käsittely on elintärkeää toimintavarmojen ja ymmärrettävien ohjelmistojen kehittämisessä. Onkin siis aiheellista ymmärtää käytettävissä olevat poikkeustilojen hallintaan tarkoitetut mekanismit, tietomallit ja käytännöt.

Poikkeus ja virhe ovat kaksi eri asiaa. Edeltävä on viestintämekanismi, joka kertoo toiminnon loppuunsaattamattomuudesta. Jälkimmäinen on väärän toteutuksen ilmentymä, jota ei pitäisi olla olemassakaan. Virheet tulisi korjata, jos suinkin mahdollista, ja poikkeustilat tulisi käsitellä tapauskohtaisesti.

Virheitä voi ilmentyä lähes missä tahansa. Ohjelmakoodi, toiminnallisuusmäärittely ja käyttäjäsyöte voivat kaikki olla virheellisiä. Oikea virheenkäsittelytapa määräytyy virheen tekijän, sijainnin ja laaja-alaisuuden mukaan. Virheiden käsittelyyn käytettävät säännöt eivät sovellu poikkeustentilojen käsittelyyn.

Vika on virheen ilmentymä. Järjestelmän ulkopuolella oleva vika ei saisi aiheuttaa virhettä järjestelmässä vaan poikkeustilan. Viat voidaan jakaa piileviin ja näkyviin. Piilevät vikat eivät ilmene järjestelmän toiminnassa, mutta saattavat jossain tilanteessa muuttua näkyviksi.

Näkyvät viat näkyvät järjestelmän toiminnassa ja aiheuttavat virheellistä toimintaa ja käyttökelvottomuutta. Vika voi olla hetkellinen tai pysyvä.

Poikkeustilat ovat funktion määrittelemiä sivuvaikutuksia, joita sen suorittaminen voi aiheuttaa. Näiden poikkeustilojen täytyy olla helposti tiedettävissä ja tunnistettavissa, jotta ne voidaan käsitellä asianmukaisesti. Poikkeustilat ovat osa funktion rajapintamääritelmää. Niiden muuttuminen voi aiheuttaa yhteensopimattomuuksia funktion ja sitä kutsuvan koodin välillä.

Jokainen ohjelmistomoduuli määrittelee, mistä poikkeustiloista se voi viestiä. Moduulin domaini ja tekniset vaatimukset sanelevat viestittävät tilat. Lähtökohtaisesti viestejä on kolmenlaisia: moduulin riippuvuuksien poikkeustiloja ilmaisevat; moduulin oman laskennan määrittelemät; ja moduulia käyttävän koodin virheistä kertovia. Näistä kaksi ensimmäistä voidaan käsitellä melko vaivattomasti osana moduulia käyttävän ohjelman tavallista toimintaa. Kolmas on eriävä, koska se syntyy järjestelmän sisäisestä, mutta moduulin ulkopuolisesta virheestä ja sen tulisi tuottaa virhetilaviesti poikkeustilaviestin sijaan.

Poikkeustilaviestit voivat liikkua funktio-, moduuli- ja ohjelmarajojen yli. Näin tapahtuessa viesti päättyy toiseen asiayhteyteen, jossa sen merkitystä tulkitaan vastaanottavan asiayhteyden domainisääntöjen mukaisesti. Vastaanottavalla asiayhteydellä on täysi valta päättää viestin käsittelystä. Tilanne joka luokitellaan poikkeavaksi yhdessä domainissa, ei välttämättä ole sitä toisessa domainissa.

Poikkeustilaviestin välitysmekanismit

Sopivan poikkeustilan välitysmekanismien valinta ei ole yksiselitteistä. Tähän vaikuttavat mm. käytettävä ohjelmointikieli, ohjelmalle asetetut suoritus- ja toimintavaateet, ohjelman käyttöikä ja ajoympäristö, ja viestintä ulkopuolisten järjestelmien kanssa. Useimmiten yksittäinen mekanismi ei pysty täyttämään kaikkia vaatimuksia, vaan on käytettävä useampaa eri mekanismia. Mahdollisimman pienen ja sopivan mekanismijoukon valinta ja sen järjestelmällinen hyödyntäminen helpottavat ohjelman ymmärtämistä ja kehittämistä.

Ehkäpä yleisin ja monimutkaisin käytössä oleva mekanismi on kutsupinon purkava ei-paikallinen paluu, joka voi suorittaa lykättyä laskentaa. Useimmiten tämä mekanismi tunnetaan nimellä heitto, nosto tai lähete. Muita mekanismeja ovat viiteparametrit; funktion paluuarvon käyttö; monikon tai merkityn summatyypin käyttö funktion paluuarvona;

yleismuuttujat ja säiepaikalliset muuttujat; paikallinen- ja pitkä hyppy; sekä funktionaalista ohjelmoinnista tulevat monadit jompikumpi, kaippa ja koeta.

Jotkin tässä tekstissä mainittavat mekanismien rajoitteet, ominaisuudet ja käyttö riippuvat käytettävän ohjelmointikielen ominaisuuksista. Esimerkiksi tyyppitykseen liittyvät rajoitteet vaihtelevat nimeävien, rakenteellisten ja piilevien tyyppijärjestelmien välillä.

Edellä mainittujen mekanismien lisäksi poikkeustilojen käsittelyyn voidaan käyttää algebrallisia vaikutusjärjestelmiä. Vaikutusjärjestelmä toimii tyyppijärjestelmän rinnalla täydentäen sitä. Vaikutusjärjestelmä sallii funktion vaatimien sivuvaikutusten merkitsemisen osaksi funktion määrittelyä ja mahdollistaa monikäyttöisten suoritusvoiden rakentamisen. Se poistaa täysin tarpeen tyypeille, jotka tuovat sivuvaikutuksia tyyppijärjestelmän piiriin. Vaikutusjärjestelmät ovat kuitenkin vähäntunnettuja ja harva ohjelmointikieli tukee niitä riittävän hyvin, joten tässä tekstissä niihin ei paneuduta. Mainittakoon kuitenkin, että heitto-mekanismi on vaikutusjärjestelmien käyttämien vaikutuskäsittelijöiden erikoistapaus.

Viiteparametri

Viiteparametrejä käytetään melko vähän poikkeustilaviestien välitysmekanismina. Ne on helppo unohtaa tarkistaa, ne vaativat ylimääräisen argumentin ja ne eivät estä käyttämästä funktion paluuarvoa poikkeustilanteessa.

Funktion onnistuneesta suorittamisesta huolimatta viiteparametrin sisältämää arvoa täytyy tarkastella. Tarkastelu vaatii haarautuvaa logiikkaa, jotta poikkeustilanteet voidaan käsitellä. Onnistuneen laskennan ja poikkeuskäsittelyn toteuttava koodi täytyy kirjoittaa limittäin, jolloin ne on hankalampi erottaa toisistaan.

Viiteparametrien käyttö poikkeustilatiedon välittäjänä pakottaa funktion paluuarvon muotoon, jossa se on paluutyypin mukainen, muttei käyttökelpoinen. Viiteparametrit soveltuvat paremmin käytettäväksi funktioissa, jotka eivät tuota paluuarvoa.

Etuna viiteparametreillä on, että ne voidaan tyyppittää vapaasti riippumatta muista funktion parametreista ja paluuarvosta. Tämä mahdollistaa yhtenäisen poikkeustilaviestityypityksen moduulin laajuisesti. Lähtökohtaisesti viiteparametrit eivät ole alttiita kilpailutilanteille. Niiden

omistajuus vaikuttaa tähän. Viiteparametrit ovat helposti paikannettavissa lähdekoodista, mikä helpottaa oikean poikkeustilakäsittelyn toteuttamista.

Paluarvo

Funktion paluarvon käyttö poikkeustilaviestin välittämiseen on harvoin toimiva ratkaisu. Sen tarkastelu on helppo unohtaa, se vaatii tarkastelua onnistumisesta huolimatta, poikkeustilakäsittely ja onnistunut laskenta täytyy toteuttaa limittäin, eikä arvon käyttöä poikkeustilanteessa estetä. Etuna paluarvoilla on, että ne eivät vaadi ylimääräistä parametriä vaan hyväksikäyttävät onnistuneen laskennan arvoa.

Paluarvon käyttö poikkeustilaviestinä vaatii, että se voidaan erottaa onnistuneen laskennan tuottamista arvoista. Tilanne jossa tämä ei ole mahdollista tunnetaan semipredikaatti ongelmana. Funktion toiminnallisuus rajoittaa mahdollisten poikkeustilaviestien arvoja. Niiden täytyy olla tyypiltään sopivia funktion tavanomaisten paluarvojen kanssa. Nämä seikat voivat aiheuttaa epäyhtenäisyyksiä moduulin käyttämien poikkeustietomallien välille, mikäli moduulin tarjoamien funktioiden paluarvot kattavat monia eri arvovälejä ja tyyppisiä. Näitä ongelmia on mahdollista kiertää käyttämällä paluarvoa ainoastaan poikkeustilaviestintään ja palauttamalla oikea paluarvo viiteparametrin avulla. Tämä tosin vaatii ylimääräisen parametrin, eikä korjaa muita ongelmia.

Paluarvot soveltuvat paremmin poikkeustilaviestintään paluarvoa tuottamattomille funktioille. Näistäkin funktioista voi olla vaikeaa huomata, milloin paluarvoa käytetään poikkeustilaviestintään.

Paluarvot eivät ole alttiita kilpailutilanteille. Viitetyyppisillä paluarvoilla omistajuus vaikuttaa kilpailutilannealttiuteen.

Paluarvojen puutteita poikkeustilan viestimisessä voidaan paikata käyttämällä hylkäyksen esto määrettä ja järjesteltyä, lineaarista, affiinia tai relevanttia substruktuurista tyyppijärjestelmää.

Monikko

Monikon käyttö funktion paluarvona ja poikkeustilaviestin välittäjänä on samankaltainen viiteparametrien ja paluarvojen kanssa: monikon sisältämä poikkeustilaviesti vaatii aina tarkastelua; onnistunutta laskentaa ja poikkeustilaa käsittelevä koodi ovat limittäin; poikkeustilanteessa täytyy pystyä tuottamaan paluutyypille kevyt, mutta käyttöön kelpaamaton

Pelkkään paluarvoon nähden monikko tarjoaa vapaamman tyyppityksen poikkeustilaviesteille. Monikkoa käytettäessä tulee pyrkiä paluarvon ja poikkeustilatiedon yhtenäiseen järjestykseen. Tämä helpottaa poikkeustilanteiden käsittelyä. Monikko tekee poikkeustiloista näkyvämpiä ohjelmakoodissa, ollessaan osa funktion paluutyyppejä. Monikkopaluarvo ei ole altis kilpailutilanteille, poislukien viitetyyppien omistajuusongelmat.

Makaabereja mittoja välttääkseni, ensimmäisen osan on päätyttävä tähän. Ollos kuitenkin huoletta arvon lukija, seuraava osa julkaistaan kyllä. Luvassa on lisää poikkeustilojen välitysmekanismeja. Uppoudutaan niihin sitten, kun on sen aika. Terveisin Grilli.





OTiI hallitus 2025 virkistysmökillä

