

# UAB „DEBETAS“ ŪKININKŲ EKONOMINĖS VEIKLOS ANALIZĖS „ŽINYNAI“ PAPROGRAMĖS KŪRIMAS

Alberta Gudauskytė, vadovas Marijus Montvilas

*Kauno kolegija*

## ANOTACIJA

Straipsnyje analizuojamos trys skirtingos integruotos programavimo kūrimo aplinkos, palyginamos pagal keletą skirtingų kriterijų. Pristatomi, kokie turi būti funkciniai ir nefunkciniai reikalavimai, kad paprogramė veiktų sklandžiai ir netrikdytų vartotojo dėmesio dirbant su ja.

**Raktiniai žodžiai:** IMT (Ilgalaikis metarialis turtas), IDE (Integrated Development Environment reiškia integruota programavimo kūrimo aplinka)

## ĮVADAS

Įmonės, kurios yra suinteresuotos naudingumu bei pelningumu ūkininkavimu, sėkmingu veiklos vystymu bei plėtimu užsiima ekonomine veiklos analize. Ekonominė veiklos analizė įmonėms yra labai svarbus reiškinys, kadangi jos dėka ūkininkai gali įvertinti darbo produktyvumą, stebėti savo pelno arba nuostolių kitimo rodiklius, numatyti galimus trikdžius ir kryptis, kurias būtų galima tobulinti.

Atliekant ekonominę veiklos analizę svarbu rasti ir panaudoti tinkamus duomenis. Rezultatas nebus efektyvus, jeigu analizėje nebus nagrinėjamas įmonės pradinis turtas, viso ilgalaikio turto balansinės vertė, darbuotojų skaičius, kiek ilgalaikio materialiojo turto (toliau IMT) įsigijo ar buvo nurašyta bei kitų panašių rodiklių. Būtent dėl šitos priežasties įmonių veiklos analizės programose turi būti paprogramė „Žinynai“, kurios dėka darbuotojai galėtų suvesti informaciją apie įmonės rodiklius ir juos galėtų lyginti bei peržiūrėti kompiuterio ekrane taip taupydami savo laiką.

**Darbo tikslas** – Sukurti esamai UAB „Debetas“ Ūkininkų ir žemės ūkio įmonių ekonominės veiklos analizės programai veikiančią „Žinynai“ paprogramę, kuri leistų vizualiai pateikti finansines ataskaitas. Tai leis paprasčiau ir taupant laiką lyginti kelių metų rezultatus.

### Darbo uždaviniai:

1. Atlikti kelių integruotų programavimo aplinkų apžvalgą.
2. Sudaryti funkcinis ir nefunkcinis reikalavimus.
3. Sukurti paprogramės algoritmą.
4. Atlikti testavimą.

**Darbo rezultatas** – Sukurta veikianti ir lengvai vartotojui suprantama „Žinynai“ paprogramė.

## 1. INTEGRUOTA PROGRAMAVIMO KŪRIMO APLINKA

Integruota programavimo kūrimo aplinka (angl. *Integrated Development Environment* - IDE) – tai kūrimo aplinka, kuri palengvina ir pagreitina programuotojo darbą ir programų kūrimą. Integruota programavimo kūrimo aplinka (toliau IDE) pasižymi tokiomis savybėmis, kaip patogi grafinė vartotojo sąsaja, automatinis teksto pabaigimas ir generavimas, grafinis programos struktūros vaizdavimas (Muslu, K., Brun, Y., Holmes, R., Ernst, M.D., Notkin, D., 2012).

Šioje darbo dalyje analizuojamos trys skirtingos integruotos programavimo kūrimo aplinkos. „Clarion“ - integruota, komercinė kūrimo aplinka, leidžianti vartotojui įsigyti bei naudotis programa, tačiau nesuteikianti galimybės jos platinti. „Clarion“ integruota kūrimo aplinka (toliau IDE) palaiko „Clarion“ programavimo kalbą. Šis įrankis leidžia nuskaityti ir atidaryti duomenų failų tipus, tokius kaip .dat, .csv, .tps ir daugelį kitų. Be to ji yra palaikoma „Windows“ operacinės sistemos aplinkoje.

„Clarion“ IDE palaiko sintaksės išskyrimą skirtingomis spalvomis ir pločiu, todėl galima lengvai atskirti kintamuosius, klases, funkcijas ir kitus elementus. „Clarion“ programavimo kalba nereikalauja rašyti didžiųjų ir mažųjų raidžių, svarbu parašytas teisingą kintamojo pavadinimą. Loginių operacijų pabaiga yra žymima taško ženklu. „Clarion“ IDE suteikia vartotojui galimybę rasti trūkstamus taškus ar kablelius. Pavyzdžiui, pažymėjus kablelio ar skliausto pradžia, iš karto matosi jo pabaiga, o jeigu pabaigos ženklo nėra, programuotojas žino, kur jį reikia padėti. „Clarion“ IDE neturi tokios funkcijos, kuri iškart galėtų aptikti klaidas. Norint sužinoti kokios klaidos yra kode, būtina sukompiliuoti programą, tik tada atsiveria langas su klaidomis. Paspaudus ant klaidos automatiškai vartotojui yra parodoma programos kodo vieta, kurioje aptikta klaida.

„Clarion“ IDE stiprioji pusė yra jos naudojami šablonai, kadangi atliekamas automatinis kodo generavimas. Atsižvelgiant į programos „Dictionary“ (liet. Žodynas) formą, kurioje yra aprašyti kintamieji, duomenų bazių struktūra bei nurodomi raktai, yra automatiškai generuojamas kodas projekto dalims, pavyzdžiui ataskaitoms, formoms, lentelėms. Ši savybė palengvina programuotojų darbą, kadangi kodas rašomas tik vietomis, kai reikia nurodyti laukus arba priskirti veiksmą. Automatiškai sugeneruotas kodas yra parašytas pilkame fone ir vartotojas jo koreguoti negali. „Clarion“ IDE yra stipriai automatizuota programa. Ji leidžia dirbti su universaliais duomenų bazės draiveriais, t.y. galima prisijungti bet kurią duomenų bazę: ODBC, MS SQL, Sybase bei Oracle. Programos kodas nesikeičia kai yra prijungta norima duomenų bazė, kadangi veikia automatinis kodo generavimas.

Kita analizuojama integruota programavimo kūrimo aplinka yra NetBeans. „NetBeans“ IDE - integruota programų kūrimo aplinka, kuri yra parašyta Java programavimo kalba. Pradžioje programa buvo sukurta programuoti tik Java programavimo kalba, tačiau dabar ji palaiko ir kitas programavimo kalbas, tokias kaip PHP, C, C++ , JavaScript ir daugelį kitų. „NetBeans“ IDE yra viena populiariausių integruotų programavimų aplinkų, kadangi ji yra atviro kodo programa, t. y. programa yra nemokama, laisvai prieinama ir ją galima laisvai platinti. „NetBeans“ palaikoma Windows, Linux, MAC OS operacinių sistemų aplinkoje.

„NetBeans“ IDE leidžia programuotojams dirbti produktyviau, kadangi programa turi tokias savybes, kaip patogi grafinė naudotojo sąsaja, sintaksės išskyrimas skirtingomis spalvomis. Aiškiai yra matoma, kur baigiasi kabutės ar skliaustai, taip pat klasių pavadinimai, kintamųjų tipai bei kiti elementai, kadangi jie išsiskiria savo spalva bei paryškiniu. Jeigu rašant programos kodą yra nelogiškai parašytas programos tekstas, jis iškarto yra pabraukiamas raudona linija, kurios dėka programuotojas lengvai randa klaidą ir ją ištaiso. „NetBeans“ integruota programavimo aplinka taip pat palaiko ir automatinį teksto pabaigimą ir generavimą. Programos kodo eilutės pabaiga yra nurodoma padėjus kabliataškio ženklą. Dar viena svarbi savybė yra tai, kad „NetBeans“ IDE programoje pakeitus kintamojo vardą jis bus atnaujintas visame projekte.

„NetBeans“ IDE palaiko automatinį kodo generavimą taip ženkliai palengvindama programuotojo darbą. Programuotojui nereikia rūpintis kodo blokais, kadangi sistema pati automatiškai juos sukuria.

Trečioji analizuojama aplinka – „Visual Studio“. „Visual Studio“ – Microsoft sukurta integruota kūrimo aplinka, palaikanti 36 programavimo kalbas, tokias kaip C, C++, Visual Basic, JavaScript, XML ir daugelį kitų. „Visual Studio“ IDE suteikia leidimą nemokamai naudotis pagrindinėse platformose, tokiose kaip „Windows“, „Linux“, „Mac OS X“.

Kaip ir kiekviena integruota programavimo kūrimo aplinka ji palaiko sintaksės paryškirimą ir tam tikrų elementų išsiskyrimą skirtingomis spalvomis, programos kodo užbaigimą ir generavimą. Programoje eilutės kodo pabaiga yra nurodoma kabliataškio ženklu. Kodo operacijos pradžioje ir pabaigoje dedami riestiniai skliaustai „{}“, kurie nurodo operacijos pradžią ir pabaigą. „Visual Studio“ IDE palaiko klaidų aptikimo funkciją – atsiranda raudonos banguotos pabraukimo linijos, įspėjančios apie kodo klaidą. Šie pabraukimai leidžia vartotojui nedelsiant išspręsti klaidas. Klaidos paaiškinimą galima pamatyti užvedus žymeklį ant raudonai pažymėtos vietos.

Kitas „Visual Studio“ IDE bruožas yra tai, kad programa turi derinimo sistemą. Ji naudinga, kad galima pažymėti tik tam tikrą vietą, iki kurios turi vykti kodo tikrinimas. Programoje jis vadinamas lūžio tašku. Norint sustabdyti programos vykdymą tam tikroje eilutėje, reikia ją pasirinkti ir spustelti kairįjį redaktoriaus langą – kai kairėje pusėje matomas raudonas apskritimas, o kodo eilutė pažymima geltona spalva ir lūžio taškas laikomas nustatytu.

## 2. SPECIFIKACIJA

Specifikacijoje yra aprašomi funkciniai ir nefunkciniai paprogramės veikimo reikalavimai. **Programos funkcionalumą nusako funkciniai reikalavimai:**

1. Programos įdiegimas – vartotojas nusiperka programą iš UAB „Debetas“, kuri yra įdiegiama prisijungus nuotoliniu būdu.
2. Programos paleidimas – programa yra paleidžiama greitai ir be jokių trikdžių.
3. Vartotojo sąsajos užkrovimas – paleidus programą yra užkraunama vartotojo sąsaja.
4. Meniu juostos valdymas – vartotojas gali pasirinkti norimos paprogramės lango atidarymą naudojantis meniu juosta.
5. Paprogramės lango užkrovimas – užkraunamas pasirinktas langas iš meniu juostos.
6. Duomenų patekimas į laukus – vartotojui pasirinkus tam tikrą sąskaitos pavadinimą ir sąskaitos tipą duomenys į laukus pereina automatiškai.
7. Įvestų duomenų įrašymas į duomenų bazę – išsaugojus pasirinktus duomenis, jie automatiškai užfiksuojami duomenų bazėje.

8. Duomenų tikrinimas – jeigu nėra duomenų arba jie yra blogai įvesti (blogai pasirinktas sąskaitos pavadinimas arba sąskaitos tipas) yra gražinama tuščia reikšmė.
9. Duomenų šalinimas – vartotojas, norintis pašalinti visus arba tik pasirinktus duomenis, turi tai patvirtinti iššokusiam lange.
10. Ataskaitų formavimas – teisingai suvedus visus duomenis yra suformuojama ataskaita.
11. Ataskaitos spausdinimas – ataskaita pateikiama spausdinimo formoje, kurią galima atspausdinti.

Programos savybės yra aprašomos nefunkciniuose reikalavimuose. Jeigu aprašyti reikalavimai nėra įgyvendinti, tada sistema tampa nenaudinga. **Reikalavimai paprogramės išvaizdai:**

- Patrauklus vartotojo sąsajos dizainas – paprogramė turi būti patraukli vartotojo akiai, neužkrauta nereikalinga informacija.
- Lengvai vartotojui suprantama sąsaja – vartotojui turi būti lengvai suprantama kaip naudotis programa.
- Neutralios spalvos – spalvos neturi trukdyti arba nukreipti žmogaus dėmesio.
- Neįkyri sąsaja – vartotojui nereikia atlikinėti daug patvirtinimų programoje.
- Lengvai suprantama meniu juosta – programos meniu juosta turi būti kuo paprastesnė, tačiau turi atlikti tam skirtas funkcijas.
- Sąsaja lietuvių kalba – kadangi programos vartotojai yra lietuvių įmonės, tai programa turi būti parašyti lietuvių kalba.

Vadinasi, paprogramės išvaizda turi būti patraukli vartotojo akiai, nevarginti jo.

#### **Reikalavimai paprogramės naudojimui:**

- Paprastas ir nesudėtingas naudojimas – lengvai suprantamas programos veikimo principas.
- Greitas programos atsakas – programa turi greitai atlikti duomenų nuskaitymą iš failų ir įrašymą į duomenų bazę be jokių trikdžių.
- Galimybė naudotis paieška pagal sąskaitos numerį – taupant vartotojo laiką yra galimybė ieškoti sąskaitos pavadinimą pagal sąskaitos numerį.
- Veiksmų atšaukimas – paspaudus mygtuką „Uždaryti“ arba „Atšaukti“ galima grįžti į pradinę būseną.

Taigi, labai svarbu, kad naudotojui būtų nesudėtinga suprasti, kaip ja naudotis.

#### **Reikalavimai vartotojui:**

- Turėti kompiuterį ir jame instaliuotą „Ūkininkų ir žemės ūkio įmonių ekonominės veiklos analizė“ programą.
- Išmanyti buhalterinę dalį, kadangi laukams reikia priskirti duomenis pagal buhalterinės sąskaitos tipus ir pavadinimus.
- Turėti įmonės direktoriaus leidimą naudotis programa ir ją pildyti.

### **3. PAPROGRAMĖS KŪRIMAS**

Įmonėje UAB „Debetas“ yra sukurta „Ūkininkų ir žemės ūkio įmonių ekonominės veiklos analizė“ programa. Programa parašyta „Clarion“ programavimo kūrimo aplinkoje „Clarion“ programavimo kalba. Programos pagrindinis tikslas yra leisti ūkininkams stebėti savo veiklos rodiklius, daryti išvadas apie darbo produktyvumą, našumą. Kadangi „Ūkininkų ir žemės ūkio įmonių ekonominės veiklos analizė“ programoje nėra atskiros paprogramės, kuri leistų įvesti kiekvienių metų rodiklių duomenis, reikia kurti „Žinynai“ paprogramę. Pagrindinis paprogramės tikslas yra suteikti vartotojui galimybę suvesti kiekvieniems metams reikiamus duomenis apie įmonę, kurie yra skirti veiklos analizės lyginimui, siekiant taupyti programos vartotojų laiką.

Kaip buvo minima anksčiau, paprogramė yra kuriama „Clarion“ programavimo kūrimo aplinkoje, „Clarion“ programavimo kalba. „Žinynai“ paprogramė pradedama kurti, kai iš pagrindinio lango (*angl. Main Frame*) yra iškviečiama nauja procedūra – „IMT\_Z\_T“ – Ilgalaikio materialiojo turto žinynas. Veiksmas atliekamas pasižymėjus pagrindinį meniu → New Procedure → Suteikiamas vardas naujai procedūrai ir pažymima, kad tai bus „Window“ procedūra. Sugrįžtama į pradinį meniu → Actions → Call a Procedure → laukelyje „Procedure Name“ įrašome susikurtos naujos procedūros pavadinimą, t.y. IMT\_Z\_T ir spaudžiamas mygtukas OK. Turint šią procedūrą yra pradedamas naujos paprogramės projektavimas. Iš minėtos procedūros yra iškviečiamos dar kelios procedūros, svarbu paminėti tai, kad norint suformuoti spausdinimo ataskaitas, kuriant procedūrą reikia nurodyti, kad tai bus „Report“ forma.

„Žinynai“ paprogramės pagrindinis langas susideda iš ID, kuris yra automatiškai sugeneruojamas ir iš metų, kuriuos vartotojas pats įveda. Metai įvedami paspaudus mygtuką „Įterpti“. Kompiuterio ekrane atsiranda naujas langas „IMT analizės duomenys“, tuomet į laukelį „Metai“ programos vartotojas pats įveda metus, už

kuriuos nori formuoti ekonominės veiklos analizės ataskaitą. Svarbu paminėti tai, kad ataskaitos daromos tik pasibaigusiais metų laikotarpiais. „IMT analizės duomenys“ langas turi dvi korteles: „Bendros vertės“ ir „IMT detalizacija pagal grupes“. Abiejose kortelėse reikšmių įvedimas yra vienodo principo.

Taupant programos vartotojo laiką, yra sukurtas specialus mygtukas „Naujas žinynas seno pagrindu“ - jo dėka pirmas užpildytas žinynas susikonfigūroja naujiems žinynams. Ši programos funkcija palengvina programos vartotojo darbą, kadangi naujo žinyno nereikia pildyti iš naujo, užtenka paspausti minėtą mygtuką ir naujas žinynas susiformuoja automatiškai. Paprogramės naudotojas įveda kitus norimus metus ir spaudžia mygtuką „Atrinkti“.

Duomenys yra priskiriami pagal sąskaitos tipą, kurį vartotojas pasirenka iš sąskaitų tipų sąrašo ir pagal sąskaitos pavadinimą. Vartotojui taip pat yra suteikta galimybė išsirinkti pavadinimą iš sąskaitų pavadinimų sąrašo. Nežinant sąskaitos pavadinimo, bet žinant sąskaitos numerį galima atlikti paiešką pagal numerį, taip lengviau yra randamas pavadinimas. Pasirinkus 10 sąskaitos tipą – „Kitas šaltinis“, vartotojas įvedą norimą pavadinimą pagal prasmę ir užpildo „Suma“ laukelį. Tai nurodo, kad tokios sąskaitos nėra. Jeigu vartotojas pasirenką sąskaitos tipą nuo 1 iki 8, tada jam įvedinėti reikšmių nereikia. Pavadinimas yra pasirenkamas iš sąskaitų sąrašo paspaudus ant piktogramos, esančios šalia lauko. Sumos laukelyje vartotojui nereikia vesti sumos, kadangi programa nuskaito reikšmę iš failo, priskiria duomenis tam tikram laukui ir įrašo į duomenų bazę.

Renkantis sąskaitos pavadinimą būtina paminėti, kad galima rinktis tik bendrą sąskaitos tipą arba detalių sąskaitos tipą. Pasirinkus bendrą sąskaitą vartotojui nebereikia žymėti punktų, kurie yra detalūs, ir atvirkščiai – jeigu renkamasi detalią sąskaitą, tada yra pasirenkami visos tos klasės detalios sąskaitos, o bendra sąskaita lieka nepažymėta. Pavyzdžiui, įmonės darbuotojas nori įvesti informaciją apie materialųjį ilgalaikį turtą, kurios sąskaitos numeris yra 12. Tuomet renkantis bendrą sąskaitą vartotojas pasirinks tik 12 sąskaitos numerį, o renkantis detalias sąskaitas vartotojas pasirinks sąskaitas, kurios priklauso 12-am numeriui: 120 – Žemė, 121 – Pastatai ir kiti statiniai, 122 – Mašinos ir įrenginiai, 123 – Transporto priemonės, 124 – Kita įranga, prietaisai, įrankiai, 125 – Kitas materialus turtas, bet nebus įtrauktas bendras sąskaitos tipas.

Užpildytas rodiklio laukas įgauna žalios spalvos rėmelį tam, kad paprogramės vartotojas nesuklystų ir kelis kartus neįvestų į tą patį lauką duomenų. Suvedus duomenis visiems laukams yra būtina paspausti mygtuką „Atrinkti“, tada visi duomenys įsirašo į jiems skirtus laukus.

Naudojantis paprograme vartotojas gali pasirinkti kelis elgesio su įvestais duomenimis būdus:

1. Redaguoti duomenis. Paspaudus pagrindiniame lange mygtuką „Redaguoti“ vartotojas gali pakeisti įvestus duomenis: sąskaitos tipą, sąskaitos pavadinimą arba sumą.
2. Šalinti duomenis. Vartotojas gali tai daryti būdamas pradiniam lange – paspaudus mygtuką „Šalinti“ ant pažymėto žinyno yra pašalinamas visas žinynas. Detalizacijos lange pasirinkus mygtuką „Šalinti“ galima ištrinti pasirinktus arba visus laukus. Abiem atvejais pasirodo pranešimo langas, kuriame vartotojas turi patvirtinti, kad tikrai nori ištrinti duomenis.

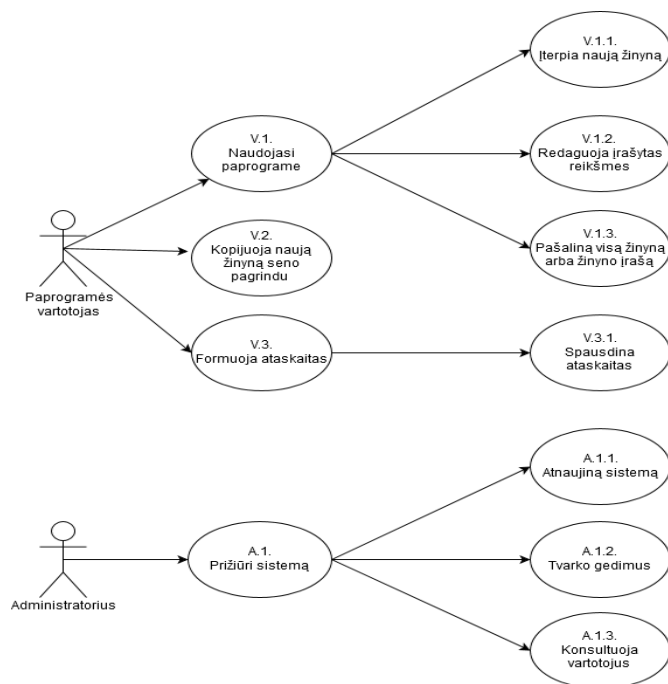
Užpildytas žinyno formas įmonės darbuotojai gali peržiūrėti juos priešspausdinimo režime ir, jeigu yra poreikis, gali atspausdinti formą, pasirinkdamas mygtuką „Spausdinti“.

Naudotojų kategorijos pateikiamos 1 lentelėje.

**1 lentelė.** Naudotojų kategorijos

Naudotojo kategorija	Sprendžiami uždaviniai	Patirtis dalykinėje srityje	Patirtis IT	Prioritetas
<b>VI.</b> Paprogramės vartotojas	1. Įterpia naują žinyną. 2. Redaguoja įvestus duomenis. 3. Šalina visa žinyną arba pasirinktą žinyno įrašą. 4. Kopijuoja naują žinyną seno pagrindu. 5. Formuoja ataskaitas ir jas spausdina.	Srities specialistas	Naujokai arba patyrę	Naudotojas
<b>A1.</b> Administratorius	1. Prižiūri sistemą. 2. Atnaujina sistemą. 3. Tvarko gedimus. 4. Konsultuoja paprogramės vartotojus.	Srities specialistas	Patyrę	Svarbiausias naudotojas

„Žinynai“ paprogramės vartotojų funkcijų galimybės yra pateikiamos panaudos atvejų diagramoje (angl. *Use Case Diagram*), kuri vaizduojama 1 paveikslėlyje.



1 pav. Panaudos atvejų diagrama.

Kuriamoje paprogramėje naudojama „Clarion“ duomenų bazė. Duomenų bazėje įrašyti duomenys yra saugomi .dat formato faile.

„Dictionary“ (liet. Žodynas) formoje yra aprašoma duomenų bazė su joje esančiais laukais bei nurodomi raktai. Ilgalaikio materialiojo turto analizei yra sukuriama IMT duomenų bazė. Duomenų bazę sudaro 39 laukai, pirminis ir unikalus duomenų bazės raktas yra ID.

Nenaudojant standartinio mechanizmo duomenų bazės ryšiai yra aprašomi naudojantis funkcija. Taip yra todėl, kad duomenų bazėje yra daug laukų ir jų visų tipai yra beveik vienodi, todėl nuspręsta taupyti laiką ir nekurti papildomų lentelių ryšiams. Programoje yra sukuriama funkcija IIGAFF\_T – laukelio žinynas su parametrais : par\_tipas, par\_kodas, par\_id. Naudojantis šia funkcija yra sukuriamas duomenų bazės ryšys: visų pirma yra išvalomi įrašai, kurie yra žinyne. Tada žinyno tipui yra priskiriama parametro tipo reikšmė, žinyno kodui priskiriama parametro kodo reikšmė, žinyno ID įgauna parametro ID reikšmę. Programos kodas:

```
clear( iig:record )
IIG:TIPAS = par_tipas
IIG:KODAS = par_kodas
iig:id = par_id
```

Duomenys, kurie turi atsispindėti duomenų bazių laukuose, yra saugomi balansinių formų ataskaitose. Programoje duomenys yra nuskaitomi iš failų, įrašomi į tam tikrus laukus ir išsaugomi duomenų bazėje, t.y. renkantis sąskaitos tipą yra nurodoma, kokia tai sąskaita (iš kur ją reikia paimti), nurodomas pavadinimas, tai kreditinė ar debetinė sąskaita ir, kai surandamas laukas su reikšme, pagal nurodytus kriterijus jis yra grąžinamas į pildomą lauką.

Norint peržiūrėti kaip atrodo duomenų bazė yra galimi du būdai:

1. Naudojantis „CapeSoft File Scanner“ programa. Ši programa leidžia perskaityti norimus failus vartotojui ir peržiūrėti kaip atrodo aprašyta duomenų bazės forma. Svarbu paminėti tai, kad programa nuskaito .dat ir .csv plėtinių failus.
2. Naudojant „Clarion“ programoje esančią „Tables“ sekciją. Svarbu tai, kad ji taip pat parodo įrašytas reikšmes į duomenų bazę. Ja naudotis yra patogiau, negu su „CapeSoft File Scanner“ programa, kadangi, jeigu yra labai daug laukų duomenų bazėje, ji leidžia susikurti filtrus, pagal ką ieškoti lauko reikšmės, taip palengvinant ir pagreitinant įrašų radimą.

#### 4. Testavimas

Norint užtikrinti programos veikimo kokybę, svarbu testuoti programinę įrangą. Kaip teigia Sigitas Dapkūnas (2007), „Testuojant nustatoma, ar programinė įranga atitinka jos specifikaciją ir reikalavimus. Radus neatitikimus, programinis kodas turi būti pataisytas, kas gali pareikalauti netgi programinės įrangos perprojektavimo.“ Tai reiškia, kad testavimo metu rasti neatitikimai sudaro galimybę taisyti iškilusius defektus, gerinti programos veikimą ir kokybę. Sigito Dapkūno nuomone, neatitikimų neradimas įrodo, kad programa veikia pagal specifikaciją ir yra tinkamai paruošta naudoti. Taigi, sistemos testavimas yra neatsiejamas siekiant pateikti vartotojui kokybišką programą.

##### Testavimo tikslai:

1. Rasti sistemos klaidas, trūkumus ir juos ištaisyti.
2. Patikrinti, ar sistema atitinka iškeltus funkcinius ir nefunkcinius reikalavimus.
3. Patikrinti, ar vartotojui bus patogu dirbti su programa.

Atliekant testavimą, reikia patikrinti klaidas, kurias gali įverti vartotojas, nežinodamas, kaip naudotis programa. Norint išvengti nesklandumų programoje yra patikrinama, ar yra galimybė įrašyti raidines reikšmes į laukus, kuriuose turi būti saugomos skaitinės reikšmės – sumos. Jeigu į lauką bandoma parašyti raidę arba simbolį, paprogramė to daryti neleidžia ir neužfiksuoja įvestos klaidingos informacijos. Svarbu paminėti ir tai, kad, kol nėra užpildytas nei vienas žinytas, neveiks žinytų kopijavimo mygtukas. Jeigu nėra nei vienas užpildytas reikšmių laukas, tada „Spausdinti“ mygtukas nebus aktyvus – privalomas nors vienos reikšmės įvedimas arba priskyrimas.

#### IŠVADOS

1. Baigiamajame darbe išanalizuotos trys skirtingos programavimo kūrimo aplinkos: Clarion IDE, NetBeans IDE ir Visual Studio IDE. Atlikus integruotų programavimo kūrimo aplinkų lyginimą galima teigti, kad tai programos, kurios palengvina ir pagreitina programinės įrangos procesą.
2. Sukurta „Žinytai“ – tai paprogramė, kurios tikslas yra suteikti vartotojui galimybę suvesti kiekvieniems metams reikiamus duomenis apie įmonę, kurie yra skirti veiklos analizės lyginimui, siekiant taupyti programos vartotojų laiką.
3. Atlikus testavimą įsitikinta, kad paprogramė atitinka specifikacijoje iškeltus reikalavimus: paprogramė veikia sklandžiai, nėra apkrauta nereikalinga informacija.

#### LITERATŪRA

1. Clarion - Home. [žiūrėta 2019 m. spalio 14d.]. Prieiga per internetą: <<http://softvelocity.com/>>
2. NetBeans IDE: Features, tips & future. [žiūrėta 2019 m. spalio 14d.]. Prieiga per internetą: <<https://jaxenter.com/NetBeans-ide-features-tips-future-149599.html>>
3. Programų sistemų kokybė. Mokomoji medžiaga. [žiūrėta 2019m. lapkričio 13d.]. Prieiga per internetą: <[https://klevas.mif.vu.lt/~sigitas/Kokybe/PS\\_Kokybe.pdf?fbclid=IwAR3NrlVciR21BYL-FQ71Mg7aaCvDsfqJckVNkxe6xDwT7p-jGxxfUJB89Kc](https://klevas.mif.vu.lt/~sigitas/Kokybe/PS_Kokybe.pdf?fbclid=IwAR3NrlVciR21BYL-FQ71Mg7aaCvDsfqJckVNkxe6xDwT7p-jGxxfUJB89Kc)>
4. Visual Studio. [žiūrėta 2019 m. spalio 5d.]. Prieiga per internetą: <<https://docs.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2019>>
5. Muslu, K., Brun, Y., Holmes, R., Ernst, M.D, Notkin, D. (2012). Speculative Analysis of Integrated Development Environment Recommendations. Tuskonas, JAV.

#### DEVELOPMENT OF A "ŽINYNAI" SUB-PROGRAM FOR ANALYSIS OF FARMERS' ECONOMIC ACTIVITY

Alberta Gudauskytė, supervisor Marijus Montvilas

*Kaunas University of Applied Sciences*

##### Summary

The article analyzes three different integrated programming development environments and compares them against several different criteria. Introduces the functional and non-functional requirements for a subroutine to run smoothly and not distract the user's attention when working with it.