



**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**INFORMATIKOS FAKULTETAS**  
**KOMPIUTERIŲ TINKLŲ KATEDRA**

Vaida Ulinskaitė

**Mokomoji aplinka programavimo pradmenų  
įsisavinimui**

Magistro darbas

**Vadovas:**

**Lekt. I. Lagzdinytė – Budnikė**

**2011-05-26**

KAUNAS, 2011



**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**INFORMATIKOS FAKULTETAS**  
**KOMPIUTERIŲ TINKLŲ KATEDRA**

Vaida Ulinskaitė

**Mokomoji aplinka programavimo pradmenų  
įsisavinimui**

Magistro darbas

**Recenzentas**

**doc. dr. S. Maciulevičius**

**2011-05-26**

**Vadovas**

**Lekt. I. Lagzdinytė – Budnikė**

**2011-05-26**

**Atliko**

**IFT-9 gr. stud.**

**Vaida Ulinskaitė**

**2011-05-26**

KAUNAS, 2011

## SUMMARY

### Educational environment for uptake of programming basics

At work analyzes the needs of students and teachers of programming elements in the teaching process. Examined the existing supply. Designed program for self-study or lessons time. Chosen to teach programming fundamentals FreePascal programming language, but it is not taught the language but the programming. Educational program consists of theory with examples, practical exercises and tests. On theoretical part student is indoctrinate with basic programming components (structure, cycles, conditional statements, etc.). The practical part is for students practice, when faced with different situations, he will see where he make the mistakes, try to mend them, know exactly what he need to learn more. Tests are designed for teachers with programming modules. This part will facilitate students assessment.

The program was created with Delphi 7 programming language. The educational material includes the use of MS PowerPoint presentation creation program.

## TURINYS

SUMMARY .....	3
TURINYS.....	4
Lentelių sąrašas .....	6
Paveikslų sąrašas .....	7
Įvadas.....	9
1. Mokomosios aplinkos programavimo pradmenų įsisavinimui poreikio ugdymo procese tyrimas.....	11
1.1. Problemos analizė .....	11
1.2. Mokytojų nuomonės tyrimas.....	11
1.3. Mokinių nuomonės tyrimas.....	12
1.4. Reikalavimai mokomosios programavimo pradmenų aplinkos pagrindinėms dalims .....	14
1.5. Esamų mokomųjų aplinkų internete analizė .....	16
2. Projektuojamos mokomosios aplinkos analizė.....	20
2.1. Sprendžiamos problemos principiniai sprendimai .....	22
2.1.1. Teorija.....	22
2.1.2. Praktika.....	22
2.1.3. Kontroliniai darbai (testai).....	22
2.2. Mokomosios medžiagos ir testų pavyzdžiai .....	22
2.2.1. Pratimai, testai.....	23
3. Mokomosios aplinkos projektavimas.....	27
3.1. Medžiagos išpildymo technologijų analizė, parinkimas ir pagrindimas.....	27
3.2. Sudėtinių elementų realizacija .....	28
3.2.1. Mokinių prisijungimo realizacija.....	31
3.2.2. Pagrindinis mokomosios aplinkos langas.....	32
3.2.3. Meniu punkto „Meniu“ realizavimas.....	33
3.2.4. Meniu punkto „Pamokos“ realizavimas.....	33
3.2.5. Meniu punkto „Testai“ realizavimas .....	38
3.2.6. Meniu punkto „Mano duomenys“ realizavimas .....	43

3.2.7. Meniu punkto „Apie“ realizavimas .....	43
4. Mokomosios aplinkos vartotojo dokumentacija .....	44
5. Mokomosios aplinkos panaudojimas mokymo procese.....	44
5.1. Mokomosios aplikos funkcijos ir galimybės.....	44
5.2. Mokomosios aplinkos įdiegimas.....	44
5.3. Mokomosios aplinkos testavimas .....	45
5.3.1. Pirminis testavimas .....	45
5.3.2. Testavimas mokykloje.....	45
5.3.3. Aplinkos testavimo apibendinimas .....	47
Išvados.....	48
Literatūros sąrašas .....	49
1 priedas. Vidurinio ugdymo bendrųjų programų 7 priedas (ištraukos).....	50
2 priedas. Vartotojo duomenų failo pavyzdys.....	56
3 priedas. Mokytojų poreikių programavimo mokyme tyrimo anketa .....	57
4 priedas. Mokinų poreikių tyrimo anketa .....	58
5 priedas. Programos vertinimo anketa (mokytojų).....	59
6 priedas. Programos vertinimo anketa (mokinų).....	60
7 priedas. Procedūros „Tikrinimas“ tekstas .....	61
8 priedas. Lukšių V. Grybo gimnazijos IT mokytojo atsiliepinimas apie mokomosios aplinkos realizuotą dalį .....	64

## LENTELIŲ SĄRAŠAS

1 lentelė. Mokomosios aplinkos svarbiausios dalys .....	15
2 lentelė. Esamų mokomųjų aplinkų palyginimas .....	19
3 lentelė. Technologijų palyginimas .....	27
4 lentelė. Medžiagos pateikimo technologijų palyginimas .....	28

## PAVEIKSLŲ SĄRAŠAS

1 pav. Kada mokotės programuoti?.....	12
2 pav. Kokias pagalbines priemones naudoja mokytojas?.....	13
3 pav. Ar reikalinga mokomoji aplinka programavimo pagrindų įsisavinimui? .....	13
4 pav. Kokias temas yra sunkiausia įsisavinti? .....	14
5 pav. Puslapio <a href="http://www.freewebs.com">www.freewebs.com</a> langas .....	16
6 pav. Puslapio <a href="http://ifreal.com">http://ifreal.com</a> langas .....	17
7 pav. Puslapio <a href="http://programavimopamokos.net">http://programavimopamokos.net</a> langas.....	17
8 pav. Puslapio <a href="http://www.emokykla.lt">www.emokykla.lt</a> langas .....	18
9 pav. Shema „Pamokų struktūra“ .....	21
10 pav. Shema „Testų struktūra“ .....	21
11 pav. Shema „Užduočių grupės“ .....	23
12 pav. Testo formos užduoties pavyzdys .....	24
13 pav. Interaktyvios užduoties pavyzdys (I dalis).....	24
14 pav. Interaktyvios užduoties pavyzdys (II dalis).....	25
15 pav. Atviro klausimo užduoties pavyzdys.....	26
16 pav. Mokomosios aplinkos panaudojimo atvejų diagrama .....	29
17 pav. Mokomosios aplinkos funkcijų hierarchija .....	30
18 pav. Prisijungimo langas .....	31
19 pav. Pagrindinis mokomosios aplinkos langas.....	32
20 pav. Pamokos „Teorija“ struktūra .....	34
21 pav. Pamokų katalogas .....	34
22 pav. Pamoka „Dvejetainiai skaičiai“ (Dešimtainio sk. vertimas į dvejetainį).....	35
23 pav. Treniruotės pradinis langas.....	35
24 pav. Vaizdas po skaičiaus pasirinkimo ir mygtuko „Dalinti“ paspaudimo ...	36
25 pav. Vaizdas po mygtuko „Tikrinti“ paspaudimo .....	36
26 pav. Pamokos „Pradžia“ dalies „Read ir ReadLn“ treniruotė.....	38
27 pav. Testo aiškinamasis langas.....	38
28 pav. Testo „Dvejetainiai skaičiai“ pirmas klausimas .....	39
29 pav. Testo „Dvejetainiai skaičiai“ pirmo klausimo eiga.....	39
30 pav. Testo „Dvejetainiai skaičiai“ antras klausimas .....	40

31 pav. Testo „Dvejetainiai skaičiai“ trečias klausimas .....	40
32 pav. Testo „Dvejetainiai skaičiai“ ketvirtas klausimas .....	41
33 pav. Testo „Dvejetainiai skaičiai“ penktas klausimas.....	41
34 pav. Raudonas kritinis laikas.....	42
35 pav. Testo „Dvejetainiai skaičiai“ rezultatas.....	42
36 pav. Mokinio duomenų langas .....	43
37 pav. Meniu punkto „Apie“ langas .....	43
38 pav. Kiek laiko mokotės programuoti? .....	45
39 pav. Ar mokomosios aplinkos navigacija aiški? .....	46
40 pav. Ar Jus tenkina mokomosios medžiagos pateikimas aplinkoje .....	46
41 pav. Ar tokia aplinka būtų naudinga mokantis programuoti? .....	47



## ĮVADAS

Šiais informacinių technologijų laikais, vis labiau pastebime, kad kompiuteriai ir jų valdomi įvairūs aparatai vis labiau „protingėja“, tampa „išmaniais“. Tik ir girdime aplink: išmanusis telefonas, išmanioji skalbyklė, išmanusis televizorius, ir dar daug daug visko. Jei daiktai, kurie mus supa kasdien, tampa tokie protingi, tai gal tuoj jiems nereikės mūsų – žmonių? Gali būti, kad vis labiau mažės darbo vietų srityse, kur darbus puikiausiai gali atlikti kompiuteriai (vienas prekybos tinklas jau dabar beveik visose parduotuvėse taiko savitarnos kasas, t.y. ten jau nereikia kasininkų). Tačiau tikrai ne visur.

Žmones, kurie bijo „protingų“ mašinų, galiu nuraminti – neužvaldys jos mūsų, kaip fantastiniame filme. Tai neįmanoma, nes kad ir kokios „protingos“ tos mašinos yra, be jokių išimčių, yra už jas protingesnių žmonių, kurie ir sukuria PROTA mašinoms. Tie žmonės – programuotojai. Jie dirba tam, kad mums būtų lengviau. Kad namų ruošoje ir kitur turėtume pagalbinių.

Kas tie programuotojai? Tai žmonės, kurie kuria programas – komandų sekas, mašinoms, o jos tik besąlygiškai jas vykdo.

Norint tapti tokiu žmogumi, reikia mokytis programavimo meno. Taip, būtent meno, nes programuoti negali žmogus, kuris, kaip mašina, tik vykdo komandas. Norint mokėti programuoti, reikia mokėti ne tik tam tikrus sakinius, žodžius, komandas, bet svarbiausia – reikia mokėti mąstyti, turėti vadinamąjį loginį mąstymą.

Mokyklose, pamokų metu yra sudėtinga to išmokyti vaikus. Turi kiekvienas pajauti savyje norą tobulėti, išmokti mąstyti, išmokti programuoti.

Atlikus tyrimą, paaiškėjo, kad savarankiškam programavimo mokymuisi yra pakankamai daug medžiagos (ypač internete), tačiau problema ta, kad visa pateikta medžiaga nėra sisteminga, visiškai išpildyta. Daugelyje vietų yra pateiktos tik atskiros dalys (pavyzdžiui, ciklai) su vienu pavyzdžiu.

Ypatingai aktuali ši problema man pasirodė 2009-2010 mokslo metais. Mokyklose programavimas prasideda nuo 9 klasės. Norint pasiekti rezultatų, reikia pradėti mokytis daug anksčiau. Turėjau mokinį (septintoką), su kuriuo neformaliojo ugdymo užsiėmimų metu mokėmės programuoti ir dalyvavome konkursuose. Valandų tokie veiklai bent mūsų mokykloje nėra skirta daug, tiksliau labai mažai – 1 val/sav. Pats vaikas labai noriai mokėsi ir namuose, ieškojosi informacijos. Tada ir kilo mintis, kad būtų daug efektyviau, jei pačius pagrindinius dalykus tokie vaikai galėtų turėti namuose.

Ieškant mokomosios medžiagos, paaiškėja dar viena problema – nėra sukurta nieko, kur mokinys galėtų pats bandyti kurti programą ir būtų pateikiama, ką jis padarė ne taip, ar jo kurta programa veikia ir pan. Trūksta grįžtamojo ryšio.

Aspektai, kurių labiausiai trūksta besimokant programuoti:

- ✓ Aiškiai, su pavyzdžiais išdėstyta teorinė dalis;
- ✓ Galimybė po kiekvienos temos atlikti įgytų žinių patikrinimą (taip lengviau įsisavinama medžiaga);
- ✓ Papildomo, įvairesnės užduotys savarankiškam sprendimui;
- ✓ Kontrolinės užduotys, apimančios 2-3 panašias temas arba po vieną išskirtinę;

Darbo *tikslas* – sukurti mokomąją aplinką, skirtą mokyklose ir namuose mokytis programavimo pagrindų (Free Pascal aplinkoje).

Darbo *uždaviniai*:

- ✓ Atlikti poreikio programavimo mokomajai aplinkai analizę mokytojų ir mokinių tarpe;
- ✓ Išanalizuoti kompiuterizavimo galimybes skirtingiems mokomiesiems aplinkos elementams;
- ✓ Parinkti tinkamas technologijas pamokų kompiuterizavimui;
- ✓ Suprojektuoti ir realizuoti mokomąją aplinką;
- ✓ Pritaikyti mokymo priemonę ugdymo procese bei išanalizuoti trūkumus ir tobulinimo galimybes;

# 1. MOKOMOSIOS APLINKOS PROGRAMAVIMO PRADMENŲ ĮSISAVINIMUI POREIKIO UGDYMO PROCESĖ TYRIMAS

## 1.1. Problemos analizė

Greitas informatikos mokslo kitimas iš esmės keičia ir informatikos mokymą, veikdamas tiek jo turinį, tiek pedagogiką. Tarptautinės informatikos organizacijos (ACM, IEEE CS) nuolat atnaujina informatikos mokymo programų rekomendacijas.[1]

Remiantis Lietuvos Respublikos švietimo ir mokslo ministro 2011 m. vasario 21 d. įsakymu Nr. V-269 patvirtintos vidurinio ugdymo bendrosios programos 7 priedu „**VIDURINIO UGDYMO BENDROSIOS PROGRAMOS: INFORMACINĖS TECHNOLOGIJOS**“, vienas iš pasirenkamųjų modulių nuo 10 klasės yra programavimas (žr. 1 priedas).

Programavimo mokymas, ypač mokymasis, – aktuali problema. Programavimo pradėdama mokytis įvairaus amžiaus: nuo pradinių klasių iki aukštosios mokyklos ir dar vėliau. Lietuvos bendrojo lavinimo mokyklose galimybė susipažinti su pirmaisiais programavimo žingsniais įgyvendinta informacinių technologijų programoje: jau penktoje klasėje mokytojai turi supažindinti mokinius su programų kūrimu, pavyzdžiui, pasitelkdami *Logo* arba *Scratch* aplinką. Vėliau, devintoje ar dešimtoje klasėje, numatyta galimybė pasirinkti programavimo modulį (čia skiriama dėmesio algoritmavimui).[2]

**Problema** – norintiems savarankiškai mokytis programuoti ar pagilinti, patikrinti žinias, reikia internete ar kitoje literatūroje ieškoti pavyzdžių, paaiškinimų. Tie pavyzdžiai dažniausiai būna sukurti savamokslių programuotojų, o jei profesionalų, tai viskas parašyta vaikams nesuprantama kalba ( pridėta daug papildomų, neesminių funkcijų).

Kita problema – mokytojų išsilavinimas programavimo srityje. Dažnai mokiniai neturi galimybių pasirinkti šio modulio dėl to, kad mokytojas jo neorganizuoja dėl savo žinių stygiaus.

## 1.2. Mokytojų nuomonės tyrimas

Pradėdama kurti mokomąją aplinką programavimo pagrindams įsisavinti, nusprendžiau atlikti tyrimą, kuris padėtų išsiaiškinti, ko trūksta informacinių technologijų mokytojams, mokant programavimo pagrindų, ką yra sudėtingiausia išsiaiškinti mokiniams, ar mokytojai norėtų turėti pagalbininką – mokomąją aplinką.

Dviejų mokyklų (Lukšių V. Grybo gimnazijos ir Kazlų Rūdos pagrindinės mokyklos) informacinių technologijų mokytojams pateikiau apklausos anketas (žr. 3 priedą). Deja, toliau bendradarbiauti sutiko tik Lukšių V. Grybo gimnazijos mokytojas (iš Kazlų Rūdos pagrindinės mokyklos anketų neatgavau), todėl aptarsiu šio mokytojo atsakymus.

Lukšių V. Grybo gimnazijoje galimybė mokytis programuoti yra ir pamokų, ir neformalaus ugdymo metu.

Mokytojas, kaip pagalbines priemones naudoja vadovėlius ir ieško informacijos (įvairesnių užduočių) internete.

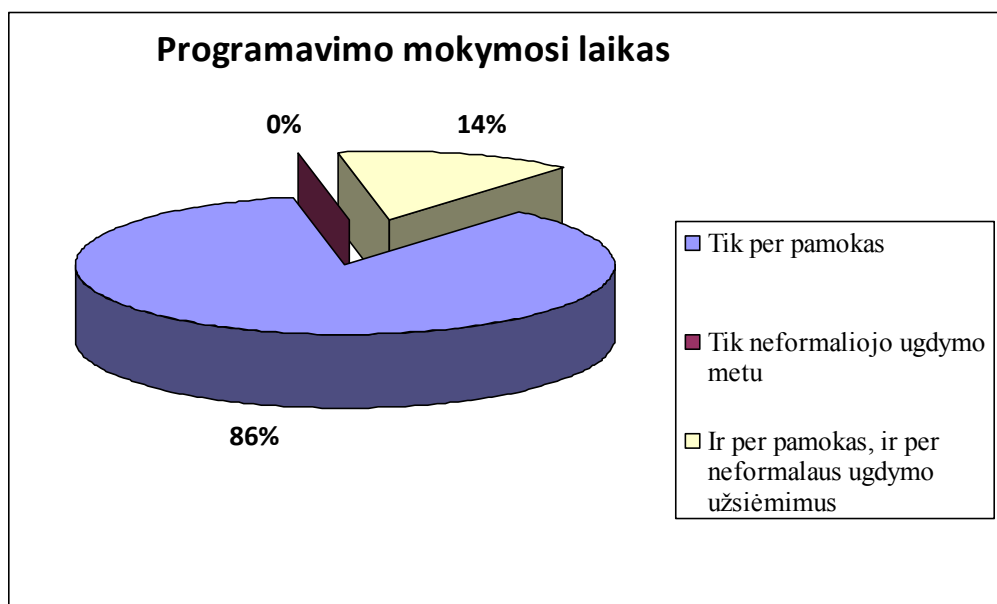
Mokytojo nuomone, mokomoji aplinka programavimo pagrindų mokymui(si) yra reikalinga. Ji turėtų būti apie Pascal arba C++ programavimo kalbą.

Mokiniam sunkiausia yra išaiškinti ciklo cikle bei įrašo tipo kintamųjų struktūrą ir sintaksę

### 1.3. Mokinių nuomonės tyrimas

Mokinių nuomonė apie programavimo mokymosi būdus ir tai, ko būtent jiems trūksta man pasirodė svarbi, kadangi mokomoji aplinka turėtų būti daugiau orientuota į mokinį. Norint sužinoti, ką apie programavimo mokymą(si) mano mokiniai, jiems taip pat buvo pateiktos anketos (žr. 4 priedas).

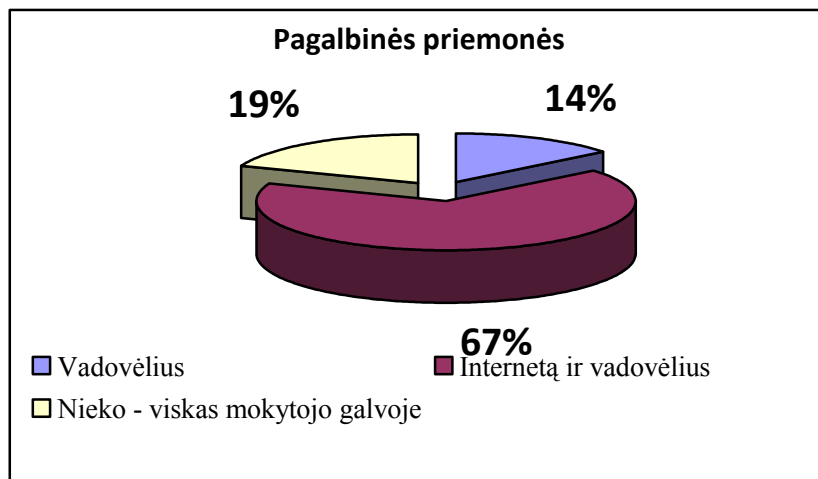
Apklausoje dalyvavo 21 mokinys (10 berniukų ir 11 mergaičių). Visi šie vaikai – 10 klasės mokiniai iš Lukšių V. Grybo gimnazijos.



1 pav. Kada mokotės programuoti?

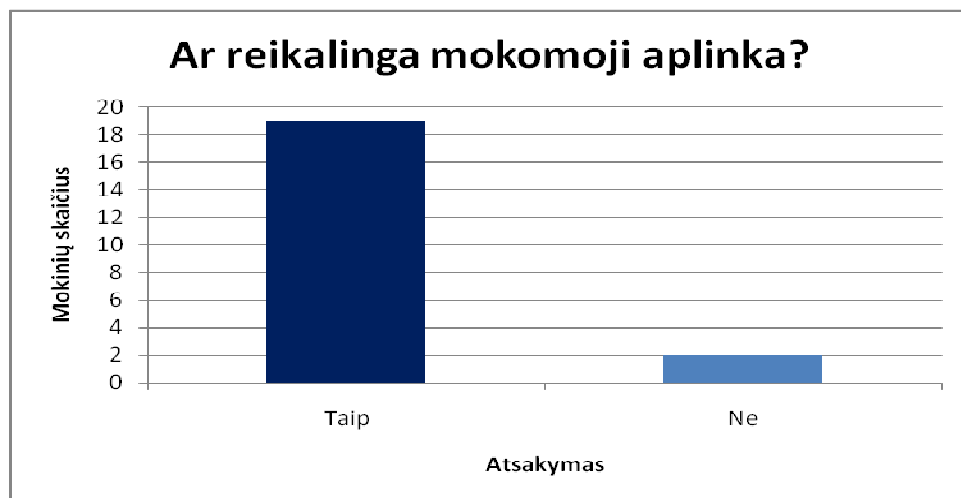
Visi apklausti vaikai mokosi programavimo. 18 iš 21 (85,7%) mokosi programuoti tik pamokų metu, likę 3 mokiniai mokosi ir per pamokas, ir per neformalaus ugdymo užsiėmimus (1 pav.).

Į klausimą „Kokias pagalbines priemones naudoja mokytojas programavimo mokymui“ dauguma mokinių (67%) atsakė, kad mokytojas naudojasi ir vadovėliais, ir internetu (2 pav.). Ten mokytojas ieško naujų pavyzdžių, įdomių užduočių, paprastesnių medžiagos pateikimo būdų.

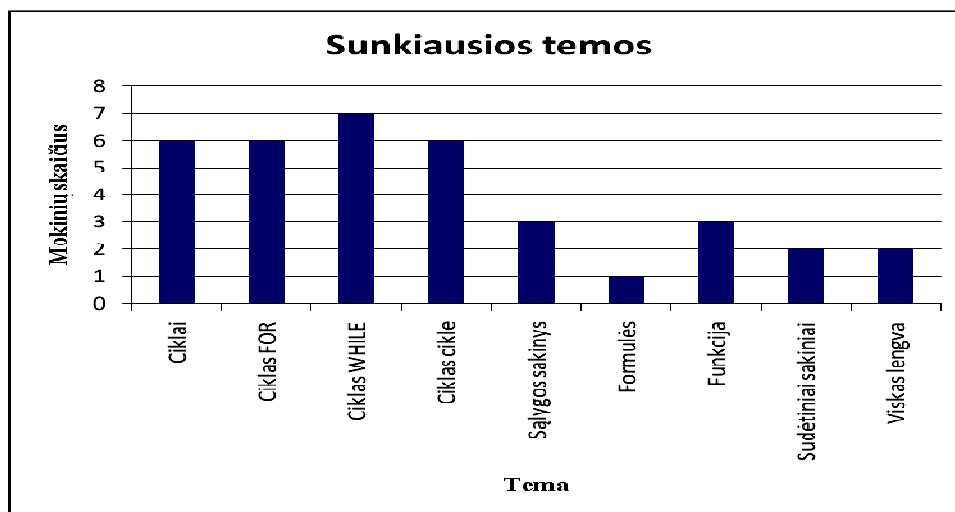


2 pav. Kokias pagalbines priemones naudoja mokytojas?

Paklausti, ar būtų reikalinga mokomoji aplinka programavimo mokymuisi, beveik visi mokiniai (19 iš 21) atsakė, kad būtų reikalinga (3 pav.). Galbūt tokia aplinka padėtų mokiniams lengviau įveikti temas, kurios sekasi šiuo metu sunkiausiai (4 pav.).



3 pav. Ar reikalinga mokomoji aplinka programavimo pagrindų įsisavinimui?



4 pav. Kokias temas yra sunkiausia įsisavinti?

Mokiniam buvo pateiktas visiškai atvikas klausimas „Kokias temas yra sunkiausia išmokyti, mokantis programuoti“. Jų atsakymai atsispindi 4 paveiksle. Galima daryti išvadą, kad daug dėmesio reikia skirti ciklams, nes, kaip matyti iš grafiko, visos temos susijusios su ciklais, kelia mokiniams rūpesčių.

#### 1.4. Reikalavimai mokomosios programavimo pradmenų aplinkos pagrindinėms dalims

Atlikus tyrimą, paaiškėjo, kokios mokomosios aplinkos reikia mokytojams ir mokiniams. Neužtenka, kad būtų tik bendrais bruožais aprašyti dalykai. Ypač mokiniai pageidauja, kad medžiaga būtų vaizdingai ir įdomiai pateikta, tuomet jiems būtų įdomiau ir aiškiau mokytis.

Visiems, tiek mokytojams, tiek mokiniams svarbu yra praktika – kad būtų galima įtvirtinti perskaitytą informaciją.

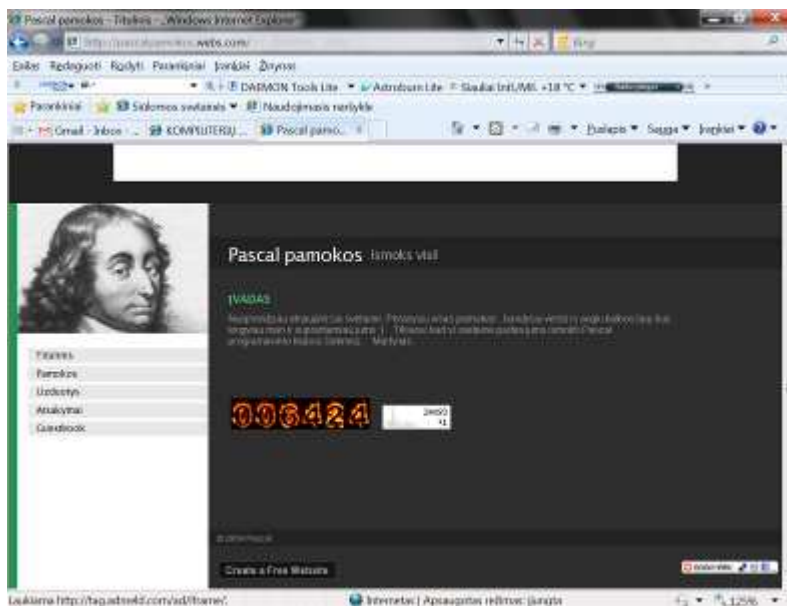
1-oje lentelėje pateikiamos mokomosios aplinkos dalys, kurios turi būti, kad aplinka būtų naudinga ir pritaikoma.

## Mokomosios aplinkos svarbiausios dalys

Dalis	Aprašymas
<u>Teorinė dalis</u>	<p>Aiškiai, su pavyzdžiais išnagrinėti teoriniai programavimo dalykai:</p> <ul style="list-style-type: none"> <li>✓ kas yra programavimas;</li> <li>✓ kas yra algoritmas;</li> <li>✓ programavimo kultūra;</li> <li>✓ trumpa programavimo kalbų apžvalga – panašumai ir skirtumai;</li> <li>✓ duomenys ir kintamieji;</li> </ul>
<u>Pamokos</u>	<p>Išsamiai aptariami ir išaiškinami pagrindiniai programavimo elementai:</p> <ul style="list-style-type: none"> <li>✓ Programos struktūra;</li> <li>✓ Skaitymo/rašymo komandos (Read/ReadLn ir Write/WriteLn);</li> <li>✓ Priskyrimo sakiniai;</li> <li>✓ Operatoriai DIV ir MOD;</li> <li>✓ Sąlygos sakiniai;</li> <li>✓ Ciklai;</li> <li>✓ Procedūros;</li> <li>✓ Funkcijos;</li> <li>✓ Masyvai;</li> <li>✓ Matricos;</li> <li>✓ Duomenų/rezultatų failai;</li> </ul>
<u>Pratybos</u>	<p>Reikia, kad mokinys turėtų galimybę, kiekvieną temą įtvirtinti praktiškai atlikdamas užduotis – matyti klaidas ir ieškoti teisingo sprendimo.</p>
<u>Testai</u>	<p>Testai iš kiekvienos temos padeda mokytojui įvertinti mokinio žinias, suprasti, kur mokinys dar nėra tvirtas, kad galėtų spręsti, ar galima pereiti prie naujos temos, ar reikia dar padirbėti prie senos.</p>

## 1.5. Esamų mokomųjų aplinkų internete analizė

### 1) [www.freewebs.com/pascalpamokos/](http://www.freewebs.com/pascalpamokos/)



5 pav. Puslapio [www.freewebs.com](http://www.freewebs.com) langas

Autorius pasistengė – įdėjo 6 pamokas, 1 užduotį ir atsakymą į tą užduotį.

1 pamoka. Siūlomas kompiliatorius, standartinis pavyzdys „Sveikas, pasauli“, paaiškinimai.

2 pamoka. „Kintamieji“. Labai išsamiai ir aiškiai pateikti kintamųjų tipai, aiškiai pateikti kai kurie pavyzdžiai, tačiau yra ir paskubomis pateiktų dalykų, netikslios sąvokos.

3 pamoka. „If, then, else“. Nepaminėta, kad sąlygos sakinyš gali būti ir be veiksmų, jei sąlyga tenkinama, o šiaip labai išsamu.

4 pamoka. „Div, mod“. Trumpas paaiškinimas, kas yra kas; užduotis – nustatyti ar skaičius lyginis ar ne; užduotis – keturženklis skaičiaus apvertimas.

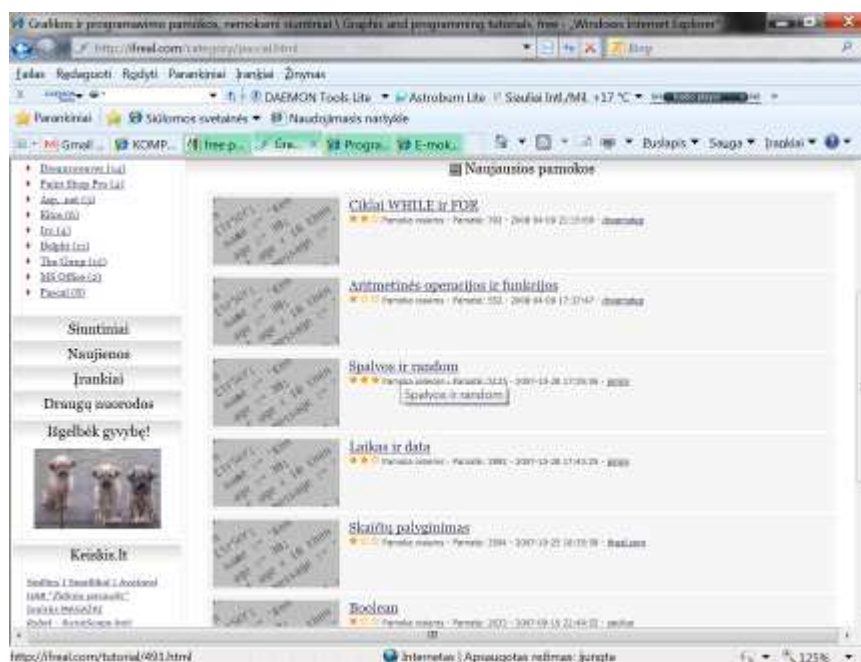
5 pamoka. „Ciklai“. Pakankamai aiškiai pateikti ciklai FOR, WHILE, REPEAT UNTIL.

6 pamoka. „Masyvai“. Paaiškinti 1-mačiai ir 2-mačiai masyvai. Manau to nepakanka, nes masyvai, pakankamai sudėtinga, paini tema pradedantiesiems.

Iš tokios „mini“ šio puslapio analizės susidarau vaizdą, kad susipažinimui čia medžiagos pakanka, tačiau trūksta praktikos, kai kur ir išsamesnio paaiškinimo. Pasigedau informacijos apie procedūras.



2) <http://ifreal.com/category/pascal.html>

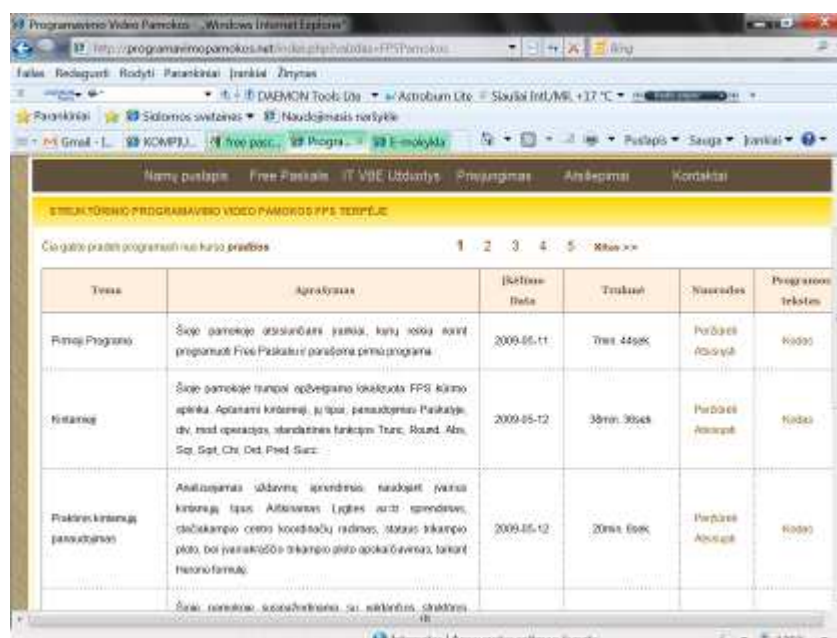


6 pav. Puslapio <http://ifreal.com> langas

Patalpintos 8 pamokos.

Visos pamokos panašios – tiesiog sudėti programų teksta. Skirta jau pažengusiems, kad sugebėtų patys, nusikopijavę tekstą į kompiliatorių, aiškintis, kuri eilutė, ką atlieka.

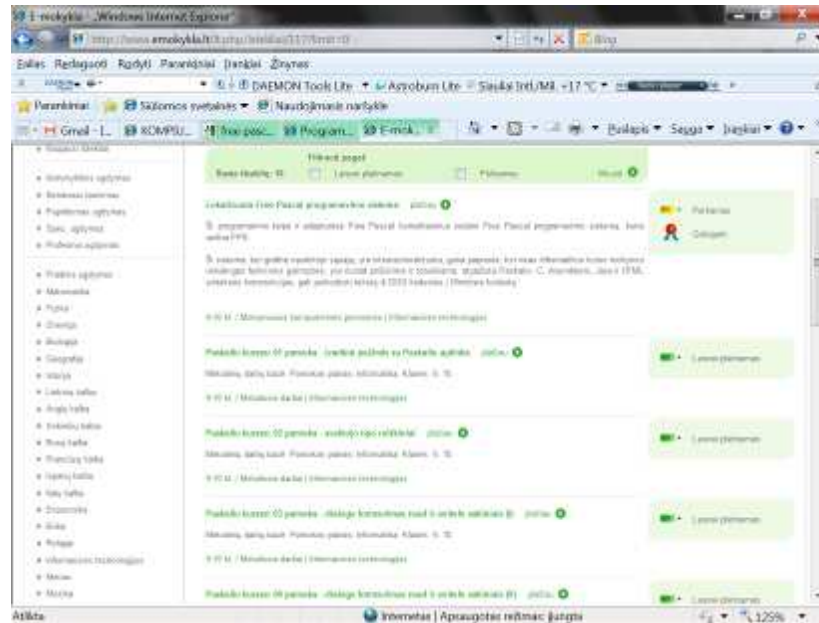
3) <http://programavimopamokos.net/index.php?vaizdas=FPSPamokos>



7 pav. Puslapio <http://programavimopamokos.net> langas

Tikriausiai padariausios pamokos iš visų rastų, nes čia sudėta filmuota medžiaga, kaip yra rašomos programos. Apima labai daug temų nuo kompiliatoriaus įdiegimo iki tekstų analizės. Galėčiau šį puslapį rekomenduoti besimokantiems programavimo.

- 4) [http://www.emokykla.lt/lt.php/istekliai/117?q\\_type\\_3%5B%5D=&q\\_type\\_6%5B%5D=&q\\_word=paskalio](http://www.emokykla.lt/lt.php/istekliai/117?q_type_3%5B%5D=&q_type_6%5B%5D=&q_word=paskalio)



8 pav. Puslapio [www.emokykla.lt](http://www.emokykla.lt) langas

Tinklapis skirtas mokytojams. Tai metodinių nurodymų rinkinys mokytojų darbo palengvinimo. Pamokų planai.

2 lentelėje pateiktas esamų mokomųjų aplinkų palyginimas.

2 lentelė.

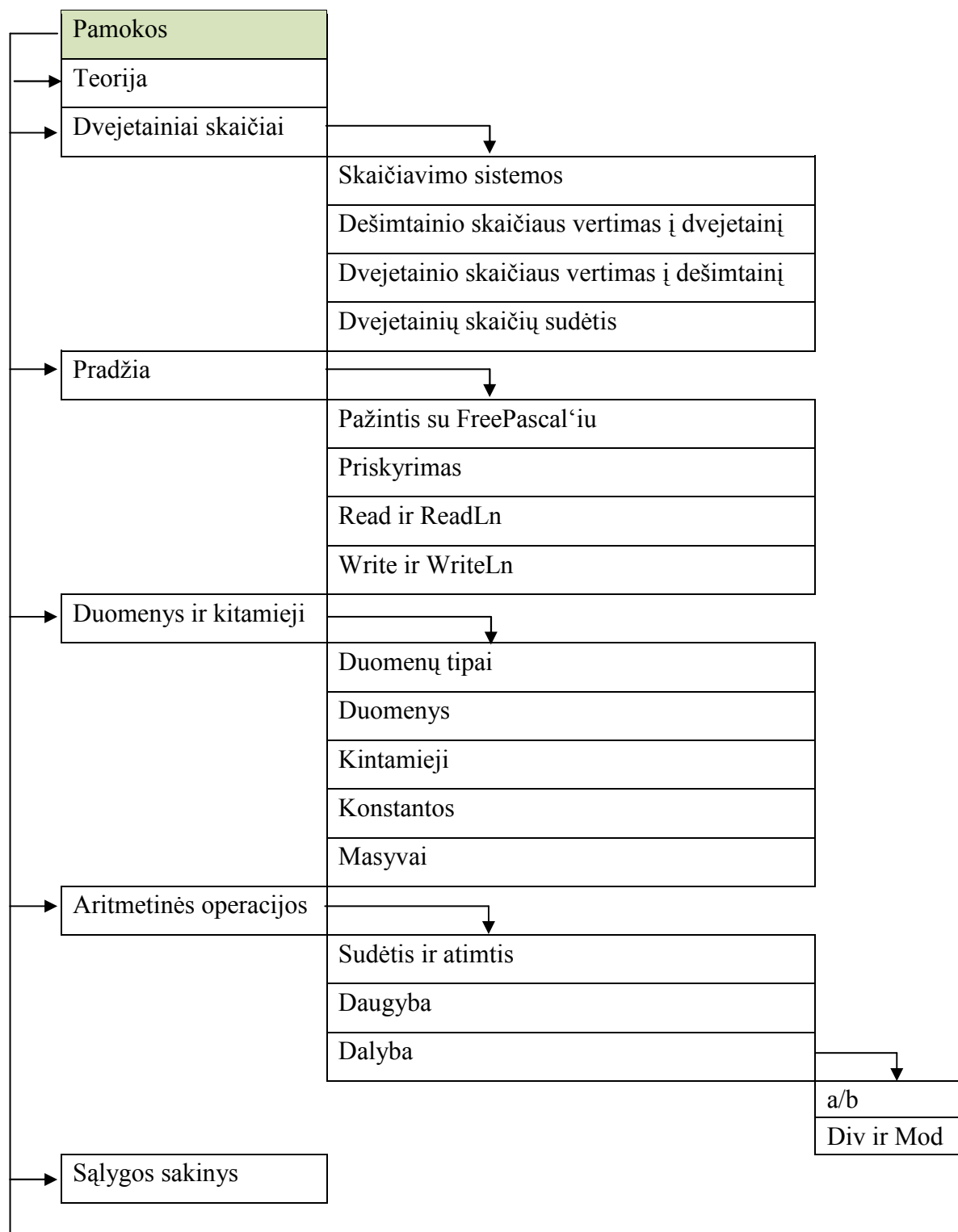
Esamų mokomųjų aplinkų palyginimas

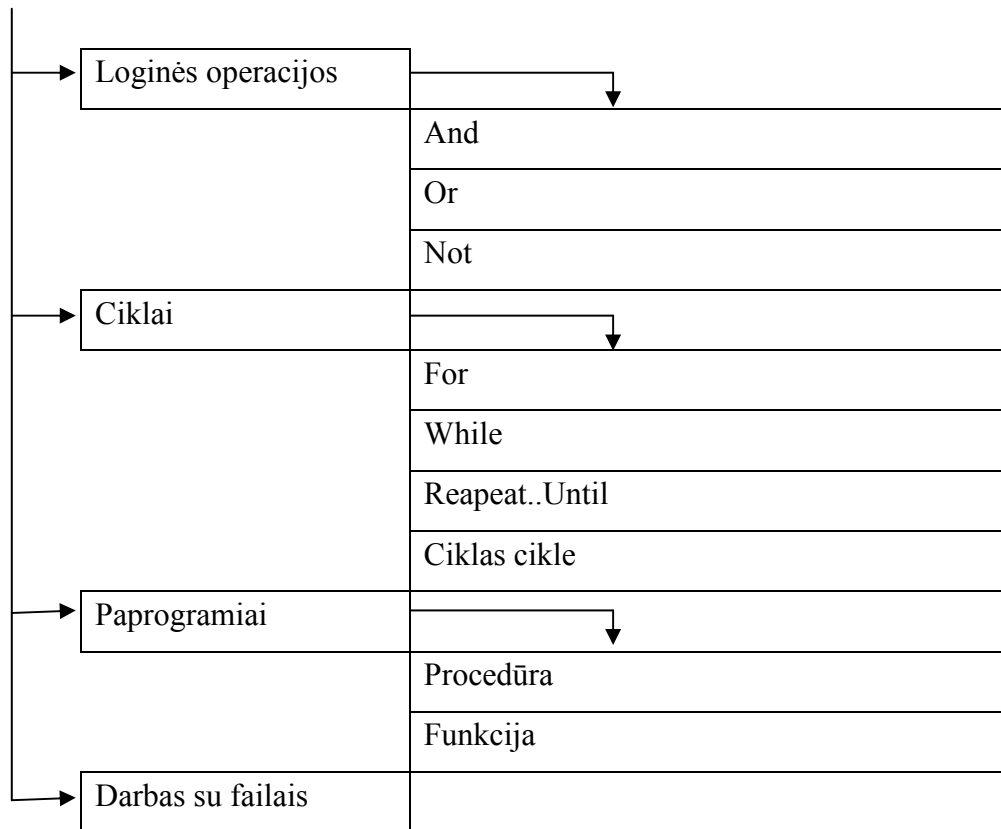
Aplinka	Kriterijus/	Teo rija	Pam okos	Prat ybos	Test ai	Pastabos
<a href="http://www.freewebs.com/pascalpamokos">www.freewebs.com/pascalpamokos</a>		-	±	-	-	Kai kurios pamokos pateiktos labai atmestinais – nepritaikyta pradedantiesiems
<a href="http://ifreal.com/category/pascal.html">http://ifreal.com/category/pascal.html</a>		-	-	-	-	Pateikti tik programų tekstai. Mokymuisi netinka, nes nėra paaiškinimų
<a href="http://programavimopamokos.net/index.php?vaizdas=FPSPamokos">http://programavimopamokos.net/index.php?vaizdas=FPSPamokos</a>		-	+	-	-	Puikiai pateikta medžiaga – video pamokos. Didžiulis trūkumas, kad esant prastam interneto ryšiui, atidaryti ir peržiūrėti yra neįmanoma

Bendrai apžvelgiant visą rastą medžiagą, galima daryti išvadą, kad reikalinga ir mokiniams ir mokytojams yra taikomoji mokomoji aplinka, kurioje būtų sudėta teorija, pamokos, praktinės užduotys, testai.

## 2. PROJEKTUOJAMOS MOKOMOSIOS APLINKOS ANALIZĖ

Mokomojoje aplinkoje dvi pagrindinės dalys – pamokos ir testai. Tam, kad kiekviena temą būtų lengviau mokytis, temos yra suskirstytos į mažesnes temas. Kad mokiniui būtų patogiau ir nereikėtų blaškytis po aplinką, pratybos yra pateikiamos kartu su mokomąja medžiaga. Paskaitęs medžiagą ir peržiūrėjęs pavyzdžius, mokinys iš karto gali pasitreniruoti pats pritaikydamas naują informaciją.





9 pav. Shema „Pamokų struktūra“

Testai
Tema „Teorija“
Tema „Dvejetainiai skaičiai“
Tema „Pradžia“
Tema „Duomenys ir kitamieji“
Tema „Aritmetinės operacijos“
Tema „Sąlygos sakiny“
Tema „Loginės operacijos“
Tema „Ciklai“
Tema „Paprogramiai“
Tema „Darbas su failais“

10 pav. Shema „Testų struktūra“

## **2.1. Sprendžiamos problemos principiniai sprendimai**

### **2.1.1. Teorija**

Programoje pateikta teorinė medžiaga su vaizdingais ir aiškiais pavyzdžiais. Teorija susijusi ne su pačiu programavimo procesu, bet su bendrais dalykais: kas yra programavimas, kas yra algoritmas, kam reikalingas, kur naudojamas programavimas. Medžiaga imama iš vadovėlio „Šiuolaikiškas žvilgsnis į programavimo pagrindus“ [4].

### **2.1.2. Praktika**

Praktika – tai teorinių žinių pritaikymas sprendžiant uždavinius. Čia mokinys interaktyviai, testo būdu ar kitaip pateiks savo sprendimus (priklausomai nuo temos). Jei atsakymas bus teisingas, „pasipils“ sveikinimai, jei neteisingas – bus patarta prisiminti teorinę temos dalį, pabandyti ištaisyti klaidas.

Praktikoje daug dėmesio bus pradmenims (kintamieji, jų tipai), taip pat ir sudėtingesniems dalykams, kuriems dėmesio niekada nebus per daug:

- a) Dvejetainiai skaičiai;
- b) DIV ir MOD;
- c) Sąlygos sakiniai;
- d) Ciklo sakiniai;
- e) Masyvai;
- f) Procedūros;

### **2.1.3. Kontroliniai darbai (testai)**

Kontroliniai darbai įvairūs:

- a) teoriniai – pateikta programa ar jos dalis, mokinys turi atsakyti, kokį rezultatą pateiks programa;
- b) testai – pateikti keli atsakymų variantai, mokinys turi pasirinkti vieną ar kelis teisingus atsakymus;
- c) interaktyvūs – mokinys turi galimybę sudėlioti programos elementus taip, kad gautų reikiamą rezultatą;

## **2.2. Mokomosios medžiagos ir testų pavyzdžiai**

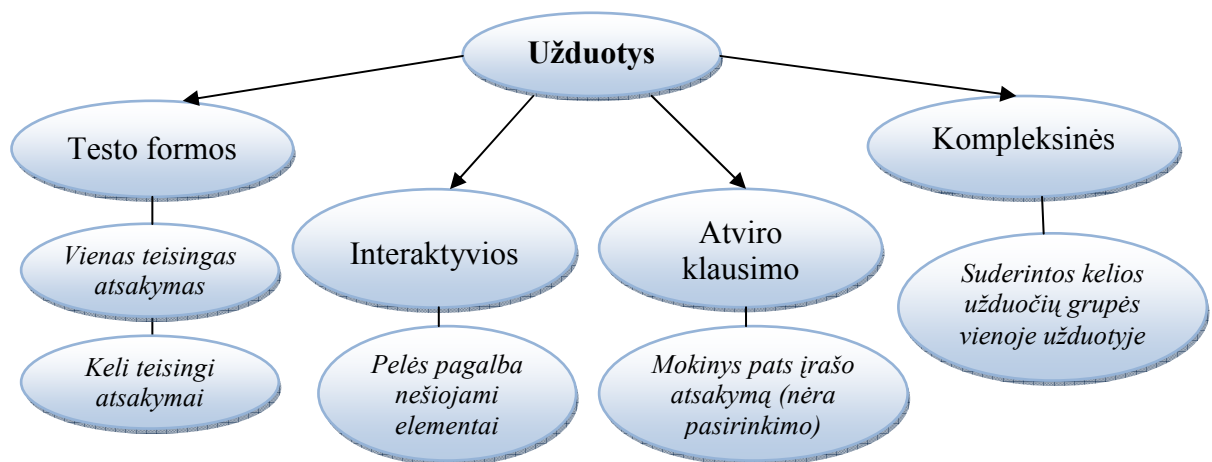
Pirmiausiai, norint pradėti programuoti, kaip ir pradėdant atlikti kokį kitą darbą reikia turėti teorinių žinių. Pražioje reikia išsiaiškinti, kas yra *programavimas*, kur galima jį

pritaikyti. Išmokti apibrėžimus, kurie bus reikalingi programuojant (*algoritmas, duomenys, reiškiniai, dvejetainiai ir dešimtainiai skaičiai*).

Vėliau galima pradėti aiškintis, kas yra *programa, funkcija, procedūra*; pradėti spęsti paprastus uždavinius; mokytis tam tikrų veiksmų: dalyba su DIV ir MOD operatoriais, ciklai, sąlygos sakiniai ir t.t.

### 2.2.1. Pratimai, testai

Mokinių žinių įtvirtinimui ir žinių patikrinimui reikia naudoti skirtingų tipų ir sudėtingumų uždavinius. Negalima pateikti vien testinių klausimų, kur reikia pasirinkti vieną/kelias atsakymus iš pateiktų, nes tuomet atsiranda galimybė tiesiog atspėti atsakymą net negalvojant. Todėl pratyboms ir testams siūlomos tokios užduočių grupės:



11 pav. Shema „Užduočių grupės“

#### 1. Testo formos uždavotys

Šios uždavotys dažniausiai naudojamos teorinių žinių – apibrėžimų tikrinimui. Testo formos uždavotys gali būti skirtingos:

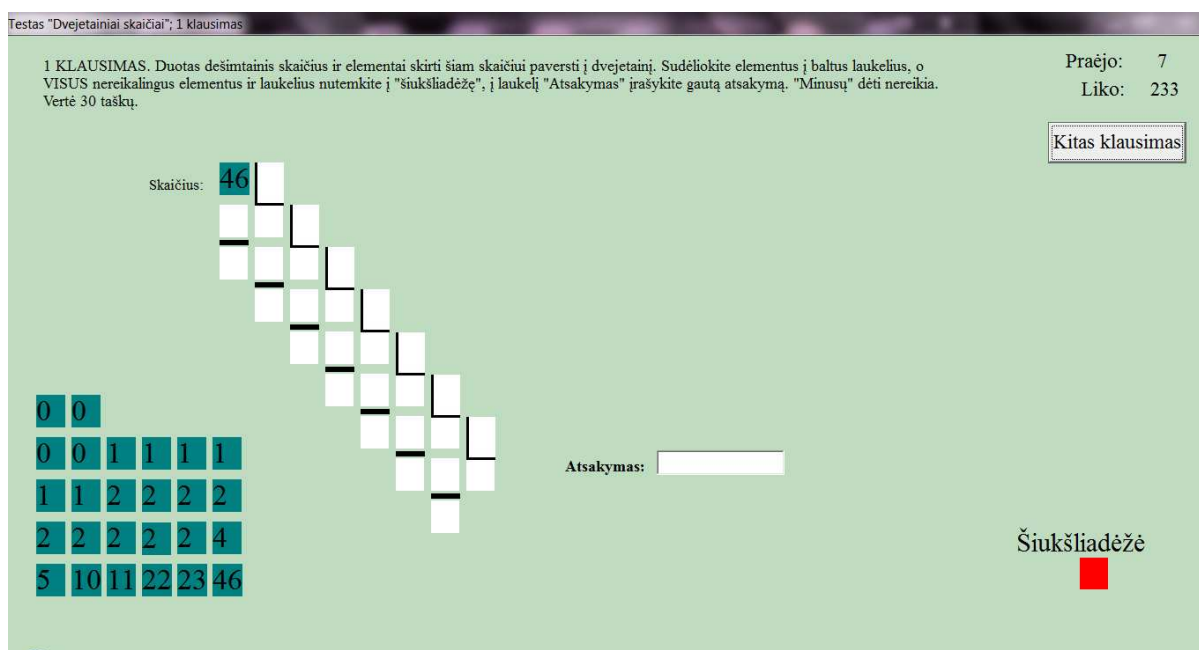
- ✓ vienas teisingas atsakymas iš kelių pateiktų;
- ✓ keli/visi teisingi atsakymai iš kelių pateiktų;



12 pav. Testo formos užduoties pavyzdys

## 2. Interaktyvios užduotys

Interaktyvios užduotys, tai užduotys, kai mokinys, naudodamasis pele, turi „sutampyti“ pateiktus elementus į reikiamas vietas. Šio tipo užduotis galima naudoti plačiai – programos teksto sudėliojimui, skaičiavimų atlikimui ir pan. Svarbu, kad, jei yra keli vienodi elementai, turi nebūti skirtumo, kurį elementą mokinys į kurią vietą padės. Pavyzdyje pateikta dešimtainio skaičiaus vertimo į dvejetainį skaičių užduotis (13 pav.) ir sprendimo dalis, parodanti, kad nereikia vienodų elementų dėti nustatyta tvarka (14 pav.)



13 pav. Interaktyvios užduoties pavyzdys (I dalis)



1 KLAUSIMAS. Duotas dešimtainis skaičius ir elementai skirti šiam skaičiui paversti į dvejetainį. Sudėliokite elementus į baltus laukelius, o VISUS nereikalingus elementus ir laukelius nutemkite į "šiuksšliadėžę", į laukelį "Atsakymas" įrašykite gautą atsakymą. "Minusų" dėti nereikia. Vertė 30 taškų.

Praėjo: 63

Liko: 177

[Kitas klausimas](#)

Skaičius: 462


4	6	2			
4	6	2	3	2	
0	2	2	1	1	2
1	1	0	5		
1					

0 0  
0  
1  
2 2

1 1 1  
2 2  
2 2 4

Atsakymas:

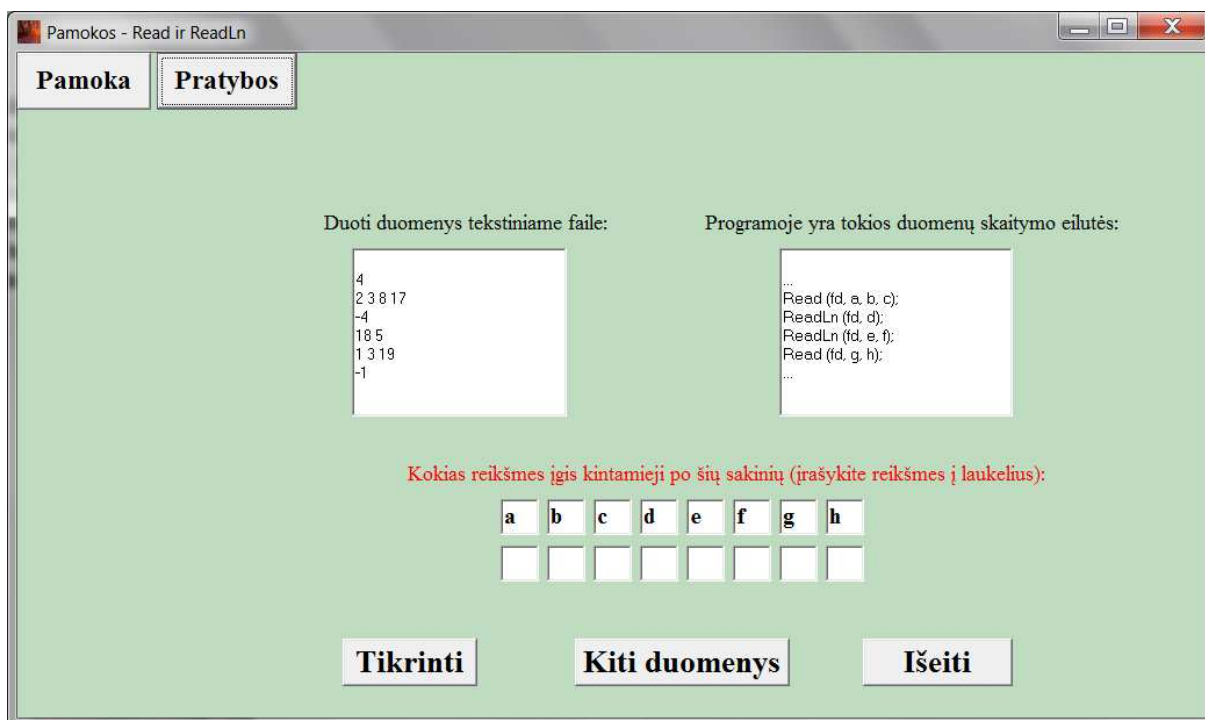
Šiuksšliadėžė



14 pav. Interaktyvios užduoties pavyzdys (II dalis)

### 3. Atviro klausimo užduotys

Šios grupės užduotys, padeda priversti mokinį galvoti, taip kaip ir interaktyvios užduotys. Tokių pratimų tikslas – mokinys pats įrašo reikiamą skaičių ar žodį į tam skirtą vietą (nėra jokių pasirinkimų).



15 pav. Atviro klausimo užduoties pavyzdys

### 4. Kompleksinės užduotys

Kompleksinės užduotys – tai užduotys, kur yra apjungiamos kelios užduočių grupės, pavyzdžiui, dalis užduoties yra interaktyvi, dalis – atviro klausimo. Kompleksinės užduoties pavyzdys gali būti jau anksčiau minėtas dešimtainio skaičiaus vertimas į dvejetainį (13 pav.). Čia, visą užduotį atlikus interaktyviai – nešiojant objektus, atsakymą reikia įrašyti pačiam mokiniui.

### 3. MOKOMOSIOS APLINKOS PROJEKTAVIMAS

#### 3.1. Medžiagos išpildymo technologijų analizė, parinkimas ir pagrindimas

Pradedant projektuoti mokomąją aplinką, svarbiausia pasirinkti tinkamas realizavimo technologijas. Tam kad pasirinktume tinkamą technologiją, reikia žinoti, ką norėsime sukurti, t.y. kokius kriterijus turi atitikti technologija. Pagrindiniai kriterijai, ko reikia mokomosios aplinkos kūrimui:

- ✓ Vartotojo sąsaja;
- ✓ Interaktyvumas;
- ✓ Duomenų kaupimas;
- ✓ Laikmatis (testų laiko ribojimui);

Kadangi tikslas yra sukurti mokomąją aplinką, kuriai nereikėtų prieigos prie interneto, tai tokias technologijas, kaip HTML ar PHP atmečiau iš karto. Pagrindinės mokomosios aplinkos dalies (pačios aplinkos, kur talpinama medžiaga, bendraujama su vartotoju ir pan.) kūrimui rinkausi tarp dviejų programavimo aplinkų: Delphi ir Java.

**3 lentelė.**

#### **Technologijų palyginimas**

<b>Kriterijai \ Technologija</b>	<b>Dephi</b>	<b>Java</b>
Vartotojo sąsaja;	+	+
Interaktyvumas;	+	+
Duomenų kaupimas;	+	+
Laikmatis (testų laiko ribojimui);	+	+

Palyginus šias dvi technologijas, paaiškėja, kad jos abi yra tinkamos mokomosios aplinkos kūrimui. Kadangi reikėjo pasirinkti vieną iš dviejų, tai pasirinkimą nulėmė didesnė mano darbo su Delphi patirtis. Taigi mokomosios aplinkos kūrimui pasirinkau Delphi7 programavimo kalbą.

Dar vienas sprendimas – kaip kuo vaizdingiau ir įdomiau mokiniui pateikti mokomąją medžiagą bei pavyzdžius. Šiam tikslui įgyvendinti rinkausi tarp MS PowerPoint pateikčių rengimo programos ir Delphi7.

## Medžiagos pateikimo technologijų palyginimas

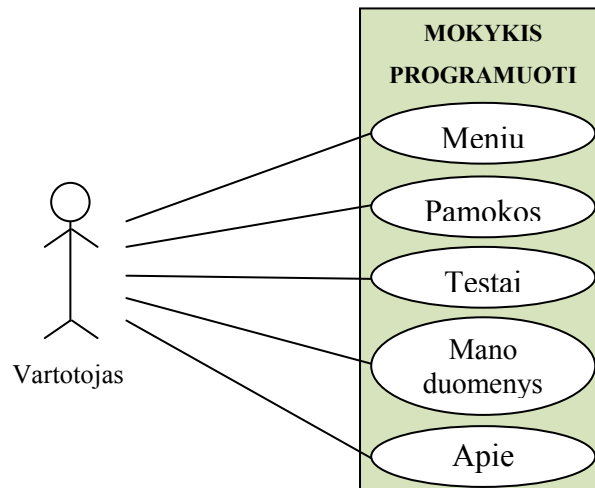
Kriterijus \ Technologija	MS PowerPoint	Delphi
Vaizdingumas	+	+
Elementų įvairovė	+	±
Paprastumas (imituojant judėjimą)	+	±
Paprastumas kuriant ankstesnio žingsnio peržiūrą	+	-
Veikimo garantija	-	+

Atlikus palyginimą, nusprendžiau, kad reikia medžiagos pateikimą kurti, naudojant abi technologijas. Tokio pasirinkimo priežastimi tapo kriterijus „Veikimo garantija“. Prie MS PowerPoint šioje vietoje atsiranda minusas, nes mokinio kompiuteryje gali ir nebūti šios programos, o jei jos nėra – pristatymas neveikia. Kadangi norėjosi medžiagą pateikti kuo aiškiau ir vaizdingiau, tai dėl veikimo garantijos nebuvimo neatsisakiau mokomosios medžiagos ir pavyzdžių pateikimo su MS PowerPoint. Iškilusią problemą išsprendžiau taip: mokiniui pasirinkus temą, mokomosios aplikos lange pasirodo medžiaga, tuo pačiu metu atsiveria naujas langas su pristatymu (PowerPoint). Jei mokinio kompiuteryje nėra įdiegtas MS Office paketas (ar tik MS PowerPoint), tai jis mato tik tą medžiagą, kuri pateikta mokomojoje aplinkoje (kurta su Delphi7).

### 3.2. Sudėtinųjų elementų realizacija

Norint identifikuoti, kas naudojasi mokomąja aplinka, tik paleidus programą, reikalingas prisijungimas. Tuomet prisijungusiam pirmą kartą yra sukuriamas duomenų failas, o kitą kartą, tiesiog prisijungiama prie to failo ir toliau kaupiami duomenys.

Po sėkmingo prisijungimo atsiveria pagrindinis programos langas, kur vartotojas iš karto yra trumpai supažindinamas su aplinka ir gali pradėti naudotis aplinka pasitelkdamas lango viršuje esančią meniu juostą.



16 pav. Mokomosios aplinkos panaudojimo atvejų diagrama

Mokomosios aplinkos „Mokykis programuoti“ meniu punktai *Menu*, *Pamokos* ir *Testai* išskaidomi į antro, trečio ir ketvirto lygio smulkesnius pasirinkimus.

Punktas „Menu“ skirtas pradinio lango rodymui (*Pradžia*), atsijungimui nuo aplinkos (*Atsijungti*) ir išėjimui (*Išeiti*).

Menu punktas „Pamokos“ suskaidytas į skirtingas temas:

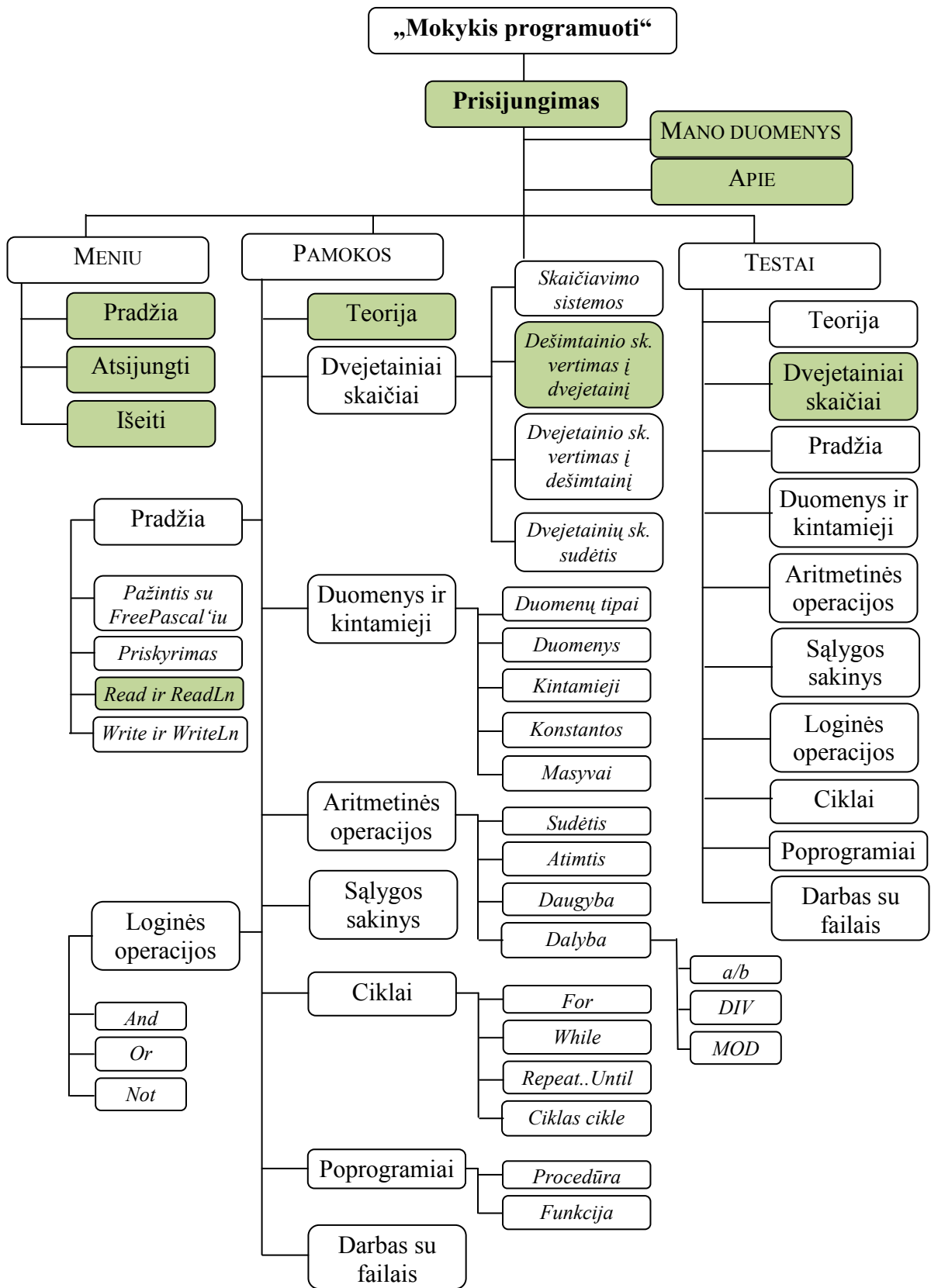
- |                           |                       |
|---------------------------|-----------------------|
| ✓ Teorija;                | ✓ Sąlygos sakiny;     |
| ✓ Dvejetainiai skaičiai;  | ✓ Loginės operacijos; |
| ✓ Pradžia;                | ✓ Ciklai;             |
| ✓ Duomenys ir kintamieji; | ✓ Poprogramiai;       |
| ✓ Aritmetinės operacijos; | ✓ Darbas su failais;  |

Daugelis iš šių temų dar suskaidytos į potemes, nes iš karto mokytis didelę temą yra sudėtinga. Pavyzdžiui tema „Ciklai“ yra suskaidyta į potemes *For*, *While*, *Repeat..Until* ir *Ciklas cikle*. Kadangi kiekvienos šios potemės įsisavinimui reikia laiko ir praktikos. Negalima visko sudėti į vieną vietą.

Punktas „Testai“ skaidomas į tokia pačias dalis kaip ir „Pamokos“, tačiau čia jau nėra potemių, nes testai laikomi iš visos temos, o ne iš jos dalies.

Punktas „Mano duomenys“ nesiskaido. Pasirinkus šį punktą, atveriamas langas, kur mokinys mato, koks failas jam sukurtas, taip pat failo turinį – kada buvo prisijungęs, kokius pamokas peržiūrėjo, kokius testus laikė, kiek užtruko, kokį įvertinimą gavo.

Punktas „Apie“ taip pat nesiskaido. Naujai atsivėrusiame lange yra pateikiama lakoniška informacija apie autorę bei padėkos žodis autoriams, leidusiems naudoti jų surinktą medžiagą. Mokomosios aplinkos funkcijų hierarchija pateikta 13 paveiksle:



17 pav. Mokomosios aplinkos funkcijų hierarchija  
(pažymėtos dalys yra realizuotos)

### 3.2.1. Mokinių prisijungimo realizacija

Šiai daliai nusistačiau tikslą, kad mokinys po pirmo prisijungimo turėtų savo duomenų dokumentą, kur būtų saugoma informacija apie:

- a) Prisijungimus;
- b) Atsijungimus;
- c) Peržiūrėtas pamokas;
- d) Laikytus testus;
- e) Testų rezultatus;

Tam, kad kiekvieną kartą prisijungdamas, mokinys vienodai rašytų savo duomenis ir jo duomenys būtų kaupiami tame pačiame dokumente, nustačiau, kad nepriklausomai nuo to, ar įjungtos mokinio klaviatūroje didžiosios raidės, ar ne, visada būtų rašoma didžiosiomis raidėmis.

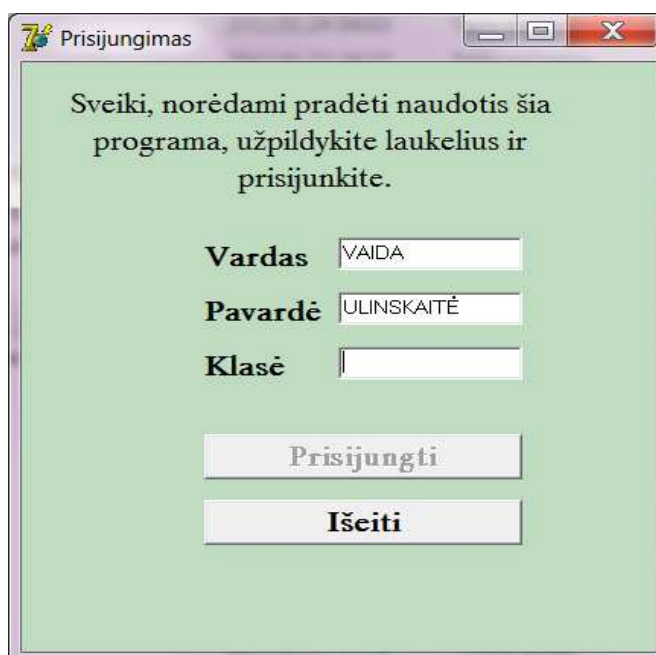
Tam, kad būtų naudingas prisijungimo ir atsijungimo fiksavimas, reikėjo išsiaiškinti, kaip fiksuoti datą ir laiką. Radau ir sėkmingai pritaikiau funkcijas[9]:

```
Copy (DateToStr (Date) , 0 , 11)
```

```
Copy (TimeToStr (Time) , 0 , 8)
```

Visa kita informacija dokumente fiksuojama automatiškai, kai vartotojas prisijungia ir atlieka veiksmus: peržiūri pamoką, laiko testą, atsijungia ir pan. Tokio dokumento pavyzdys pateiktas 2 priede.

Vartotojo prisijungimo langas pateiktas 18 paveiksle. Prisijungdamas vartotojas, būtinai turi užpildyti visus laukus, kitaip negali paspausti mygtuko „Prisijungti“. Ši sąlyga taip pat skirta tam, kad būtų lengviau įgyvendinti registraciją duomenų dokumente.



18 pav. Prisijungimo langas

Tikrinimas, ar visi laukeliai užpildyti, yra atliekamas, kai parašoma kas nors į bet kurį laukelį (jei visi laukai yra užpildyti, aktyvuojamas mygtukas prisijungti):

```
if (Edit1.Text > ' ') and (Edit2.Text > ' ') and (Edit3.Text > ' ')
  then Button1.Enabled:= true;
```

Veiksmai, kuriuos programa atlieka, kai vartotojas paspaudžia mygtuką „Prisijungti“:

```
Vardas:= Edit1.Text; // Įsimenamas vardas
Pavarde:= Edit2.Text; // Įsimenama pavardė
Klase:= Edit3.Text; // Įsimenama klasė
// Sudaromas duomenų failo vardas
FailoVardas:= 'Mokiniai/' + Vardas + '_' + Pavarde + '_' + Klase + '.txt';
Tikrinimas(); // Kreipiamasi į tikrinimo procedūrą, kur tikrinama
// ar reikia sukurti naują failą, ar užtenka prisijungti
```

Procedūra, į kurią kreipiamasi, kai reikia užregistruoti vartotojo atliktus veiksmus:

```
procedure TForm1.Registravimas(a: string);
var F: TextFile;
begin
  AssignFile(F, FailoVardas); // Prisijungiama prie duomenų failo
  Append(F);
  Write (F, a); // Įrašoma, kas buvo atlikta, pvz. „Prisijungta“
  Write (F, Copy(DateToStr(Date), 0, 11) + ' '); // Įrašoma data
  WriteLn (F, Copy(TimeToStr(Time), 0, 8)); // Įrašomas laikas
  WriteLn (F);
  CloseFile(F); // Duomenų failas uždaromas
end;
```

Tai universali procedūra, į ją kreipiamasi visada, kai reikia užregistruoti vartotojo veiklą, tiesiog skiriasi įrašomas tekstas („Prisijungta“, „Atsijungta“, „Peržiūrėta pamoka X“)

### 3.2.2. Pagrindinis mokomosios aplinkos langas

Tik prisijungęs mokinys patenka į langą, kur yra trumpas pasisveikinimas, supažindinamas su programa. Šiame lange (19 pav.) yra galimybė įsidiegti savo kompiuteryje FreePascal aplinką (tereikia paspausti mygtuką „Įdiegti kompiliatorių“).



19 pav. Pagrindinis mokomosios aplinkos langas



Kompiliatoriaus diegimo failas yra patalpintas kataloge priklausančiame mokomajai aplinkai. Jo paleidimas vykdomas paprasta komanda[9]:

```
ShellExecute(Handle, 'open', 'Nuorodos\fps 0.6.4a_lt.exe', nil, nil, SW_SHOWNORMAL);
```

Tolimesni pasirinkimai priklauso tik nuo paties vartotojo – ką jis nori (ar jam reikia) veikti. Navigacijai skirta viršuje esanti meniu juosta.

### 3.2.3. Meniu punkto „Meniu“ realizavimas

Tai viena iš paprasčiausių realizacijų. Šiame punkte yra 3 pasirinkimai: „Pradžią“, „Atsijungti“, „Išeiti“.

- a) Pasirinkus „Pradžią“, vartotojas iš bet kurios aplinkos vietos yra grąžinamas į pradinį langą (19 pav.):

```
Form2.Show;
```

- b) Pasirinkus „Atsijungti“, vartotojas yra atjungiamas iš mokomosios aplinkos ir jam pateikiamas prisijungimo langas (18 pav.)

```
Form1.Show();
```

```
Form2.Close;
```

```
Form1.Registravimas('Atsijungta: '); //Įrašas duomenų faile
```

- c) Pasirinkus „Išeiti“, yra uždaroma mokomoji aplinka. Vartotojas darbą baigė. Duomenų faile įrašoma, kad jis atsijungė.

```
Form1.Close;
```

```
Form2.Close;
```

```
Form1.Registravimas('Atsijungta: ');
```

### 3.2.4. Meniu punkto „Pamokos“ realizavimas

Realizuotos trijų skirtingų pamokų dalys:

- ✓ Teorija;
- ✓ Dvejetainiai skaičiai → Dešimtainio skaičiaus vertimas į dvejetainį;
- ✓ Pradžią → Read ir ReadLn;

### 3.2.4.1. Pamoka „Teorija“

**Kas yra programavimas?**

- Algoritmas
- Programavimo kultūra

**Programavimas**

**PROGRAMAVIMAS – veiksmų, atlikti, apibrėžimas (planavimas), kompiuterini su**

- Kad būtų lengviau suprasti, reik paprasčia pavyzdį: susitinkate nemoka kalbėti lietuviškai ir būtinors pasakyti. Ką darote? Pasite jumis abiemis suprantama bend užsienio kalba, gestų kalba ir p kad tas žmogus jus suprastų.

**Pavyzdys**

1 žingsnis. Į keptuvę įdėti 4 koldus ir kepti 5 minutes.

- Neskepti koldai
- Keptuvė
- Įskaita vieno pusė
- Įskaita koldai

**Kultūra (2)**

Tekstą tikslingai rašyti taip, kad aiškios dalyje pagal prasme būtų atpažįstamos ir suvokiamos. Tam reikia jį išdėstyti lape lyg kur reikia paleikant didesnes įtraukas.

Pavyzdžiui, šitaip parašytą tekstą sunku skaityti:

begia  
Kad patsim, amsi, raišio, raišio,  
Witold, al, p, rmas, amsi, raišio,  
...  
end.

Patartume jį rašyti taip:

Vėr patsim, amsi, raišio, raišio,  
begia  
Witold, al, p, rmas, amsi, raišio,  
Witold, al, p, rmas, amsi, raišio,  
...  
end.

20 pav. Pamokos „Teorija“ struktūra

Jau buvo minėta anksčiau, kad lygiagrečiai medžiaga yra pateikiama paprastame aplinkos lange ir su pateikimais. Tokį medžiagos pateikimo būdą ir atspindi 20 paveikslas.

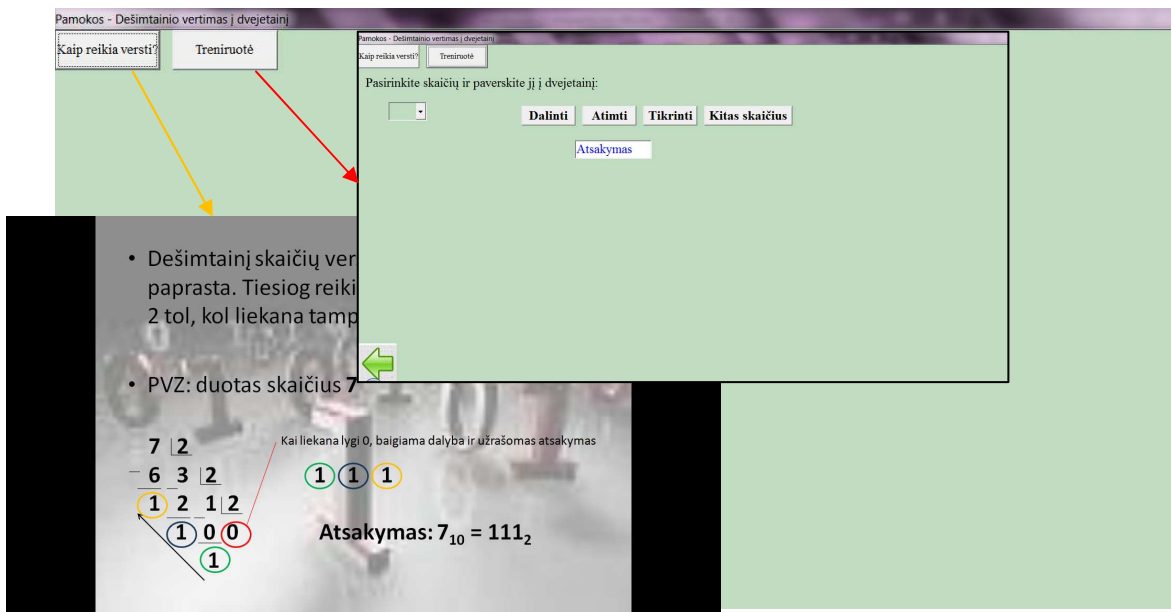
Mokomoji medžiaga yra įkeliama į aplinką iš tekstinio dokumento, kuris yra kataloge, priklausančiame mokomajai aplinkai (21 pav). Ten sudėta visa medžiaga, todėl mokytojui pakankamai patogiu ir paprastu kažką pakeisti (kiekvienas turi savo aiškinimo metodų).

Pavadinimas	Modifikavimo data	Tipas	Dydis
Algoritmas	2011.04.29 01:01	Microsoft Office P...	359 KB
Algoritmas	2011.04.29 00:57	Microsoft Office P...	317 KB
Algoritmas	2011.01.22 18:23	Tekstinis dokumen...	1 KB
Kas yra programavimas	2011.04.29 01:04	Microsoft Office P...	251 KB
Kas yra programavimas	2011.04.29 01:04	Microsoft Office P...	251 KB
Programavimas	2011.01.21 23:56	Tekstinis dokumen...	2 KB
Programavimo kultūra	2011.04.29 02:04	Microsoft Office P...	232 KB
Programavimo kultūra	2011.04.29 02:03	Microsoft Office P...	232 KB
Programavimo kultūra	2011.04.29 02:25	Tekstinis dokumen...	1 KB

21 pav. Pamokų katalogas

Kaip matome, kiekvienai temai yra skirta po 3 failus: parengto pristatymo, PowerPoint rengyklės ir tekstinis. Visa tai tam, kad mokytojas galėtų savo nuožiūra papildyti medžiagą jam patogesniu būdu.

### 3.2.4.2. Pamoka „Dvejetainiai skaičiai“ . Dalis „Dešimtainio skaičiaus vertimas į dvejetainį“



22 pav. Pamoka „Dvejetainiai skaičiai“ (Dešimtainio sk. vertimas į dvejetainį)

18-ame paveiksle matome pateiktą teorinę medžiagą apie tai, kaip reikia atlikti vertimą iš dešimtainio skaičiaus į dvejetainį. Taip pat čia yra ir treniruotės galimybė. Tai pratybos, kurios yra prie kiekvienos temos. Trumpai aptarsiu pratybų veikimo principą.

### 3.2.4.3. Pratybų „Dešimtainio skaičiaus vertimas į dvejetainį“ realizavimas

Norint pradėti treniruotę, reikia paspausti mygtuką „Treniruotė“ (22 pav.). Tuomet atsiveria langas (23 pav.), kuriame reikia pasirinkti norimą skaičių ir spaudant mygtukus „Dalinti“ ir „Atimti“, atlikti vertimą iš dešimtainio į dvejetainį.



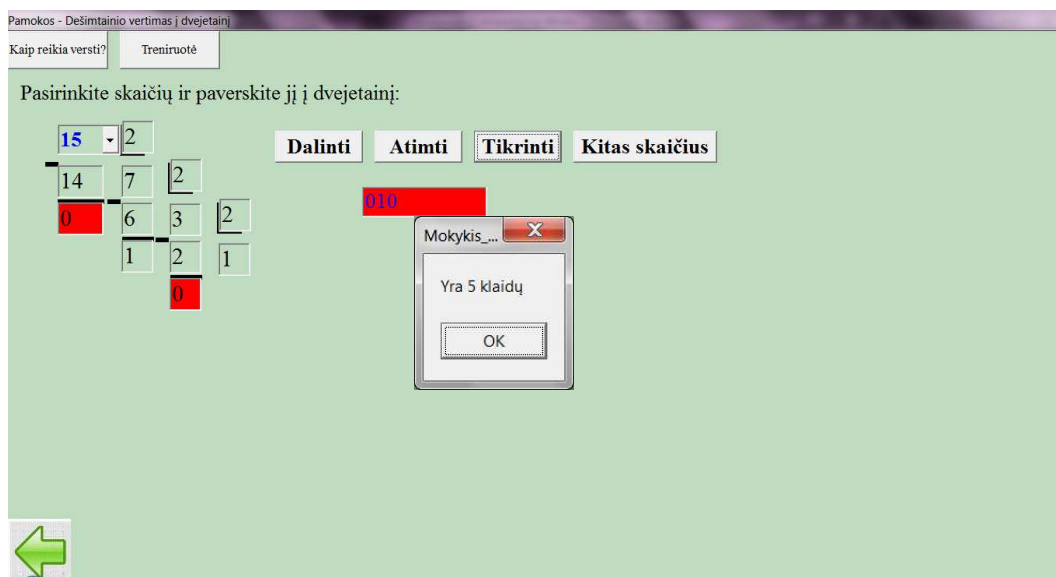
23 pav. Treniruotės pradinis langas

Pasirinkus skaičių ir paspaudus mygtuką „Dalinti“ ar „Atimti“ atsiranda laukeliai atitinkamose vietos, į kuriuos reikia įrašyti skaičius.



24 pav. Vaizdas po skaičiaus pasirinkimo ir mygtuko „Dalinti“ paspaudimo

Kai mokinys mano, kad jau baigė vertimą ir užduotis atlikta, jis paspaudžia mygtuką „Tikrinti“. Tuomet jam parodoma, kiek klaidų padarė, Klaidingos vietos nudažomos raudonai (25 pav.)



25 pav. Vaizdas po mygtuko „Tikrinti“ paspaudimo

Jei raudonų laukų yra mažiau, nei klaidų skaičius pranešime, reiškia, kad mokinys dar nebaigė skaičiaus versti iki galo. Išjungęs pranešimo lentelę, mokinys gali pasitaisyti savo padarytas klaidas.

Paspaudus mygtuką „Kitas skaičius“, ankstesni veiksmai pranyksta, atsiranda pradinis treniruotės langas, kur vėl galima viską pradėti iš pradžių, pasirinkus kitą skaičių.

Treniruotės realizacija paprasta. Spaudant mygtukus „Dalinti“ ir „Atimti“, tiesiog parodomi langai, kurie iki tol buvo nematomi:

```

Edit27.Visible:= True;
Edit27.SetFocus;           // Žymeklio vietos nustatymas
Edit27.Color:= clWindow;

```

```

Edit27.Text:= '0';
Image23.Visible:= True;
Edit28.Visible:= True;
Edit28.Color:= clWindow;
Edit28.Text:= '0';

```

Paspaudus mygtuką tikrinti, programa atlieka tokius veiksmus:

```

Vertimas; // Kreipiasi į procedūrą, kuri, mokinio pasirinktą skaičių paverčia į dvejetainį
Tikrinimas; // Kreipiasi į procedūrą, kuri tikrina, ar mokinio surašyti skaičiai atitinka gautus
programiškai

```

Procedūros „Vertimas“ tekstas:

```

procedure TForm14.Vertimas;
begin
  sk:= StrToInt(ComboBox1.Text); // Nuskaitomas mokinio pasirinktas skaičius
  i:= 0;
  while sk > 0 do
  begin
    i:= i + 1;
    Dalyba[i]:= 2; // Dalmenuų masyvas
    Liekana[i]:= sk div 2; // Liekanų masyvas
    Atimti[i]:= Liekana[i] * 2; // Atimčių masyvas (skaičių kuriuos reikia atimti)
    Skirtumas[i]:= sk - Atimti[i]; // Gautų skirtumų masyvas
    sk:= Liekana[i];
  end;
end;

```

Procedūros „Tikrinimas“ tekstas pateiktas 7 priede. Šioje procedūroje tikrinami visi laukai su anksčiau gautais masyvais ir skaičiuojamos klaidos, dažomi klaidingi laukai.

#### 3.2.4.4. Pamokos „Pradžia“ dalies „Read ir ReadLn“ pratybų realizavimas

Šioje temoje, kaip ir anksčiau aptartose, pateikiama teorija ir pratybos. Šių pratybų tikslas paprastas – kad mokinys pajautų skirtumą tarp komandų „Read“ ir „ReadLn“. 26 paveiksle yra pateikta užduotis, kur mokinys, matydamas, kas yra duomenų faile ir programos tekste, turi užpildyti laukelius, kokias reikšmes įgyja kintamieji. Tekstinių failų pavyzdžiai, kaip ir programų dalys yra saugomi tekstiniuose failuose („1t.txt“, „2t.txt“, ..., „1p.txt“, „2p.txt“, ...), kurie atidaromi atsitiktine tvarka, kai mokinys pasirenka „Kiti duomenys“. Tikrinimo uždavinį šiuo atveju dar labiau supaprastinau. Mokinio duomenys lyginami su dar vieno failo („1ats.txt“, „2ats.txt“, ..., „nats.txt“) duomenimis, kur yra surašyti atsakymai į užduotis.

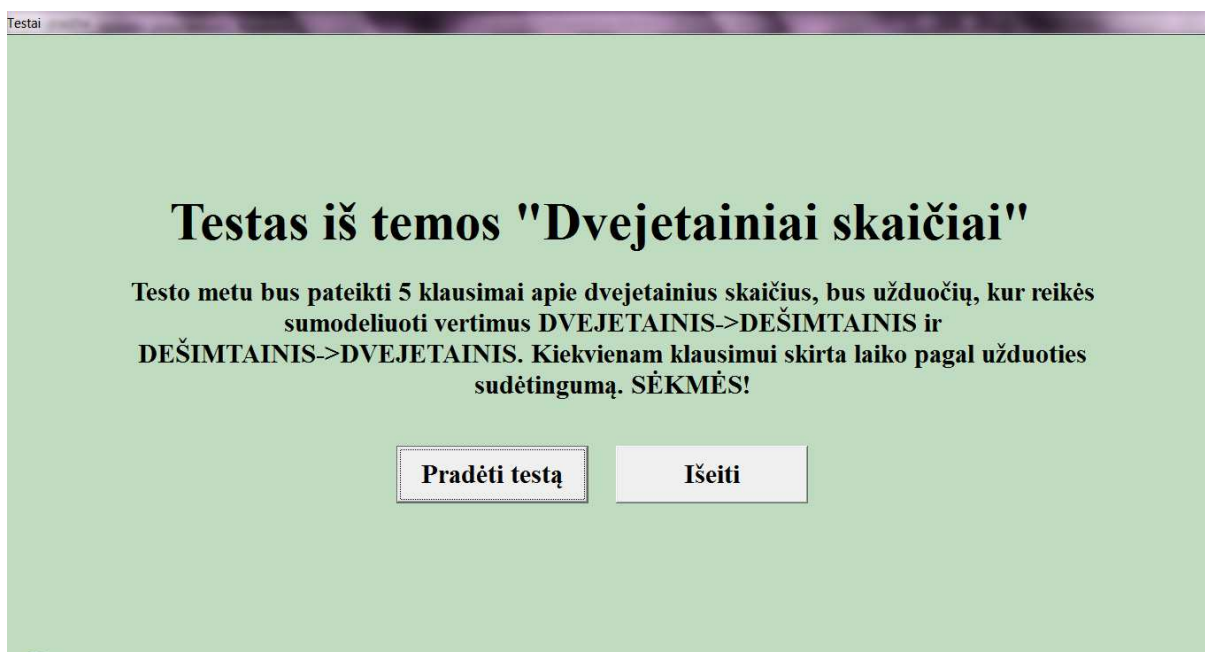


26 pav. Pamokos „Pradžia“ dalies „Read ir ReadLn“ treniruotė

### 3.2.5. Meniu punkto „Testai“ realizavimas

#### 3.2.5.1. Testo iš temos „Dvejetainiai skaičiai“ realizavimas

Menu juostoje pasirinkus punktą „Testai“ ir paspaudus ant „Tema „Dvejetainiai skaičiai““, atsiveria langas (27 pav.). Tai aiškinamasis langas, ko galima tikėtis, kai bus pradėtas testas. Mokinys gali rinktis ar nori pradėti testą, ar išeiti.



27 pav. Testo aiškinamasis langas

Paspaudus mygtuką „Pradėti testą“, atsidaro pirmas klausimas (28 pav.). Visuose klausimuose yra tie patys elementai: dešiniame viršutiniame kampe *praėjęs ir likęs laikas*, mygtukas „*Kitas klausimas*“, viršuje – *užduotis*.

Pirmasis klausimas – mano manymu, labai interaktyvus. Čia reikia su pele „sunešioti“ žalius laukus esančius apačioje į reikalingas vietas. Svarbu, kad nėra reikšmės eilės tvarkai. Visai neturi skirtumo, kurį dvejetą ar kitą skaičių nutemsime, svarbu tik, kad toje vietoje turi būti tikrai toks skaičius (29 pav.).

28 pav. Testo „Dvejetainiai skaičiai“ pirmas klausimas

29 pav. Testo „Dvejetainiai skaičiai“ pirmo klausimo eiga

Jei mokinys atsako į klausimą greičiau nei baigiasi laikas, jis gali spausti mygtuką „Kitas klausimas“, tada atsakytas klausimas išsaugomas ir patikrinamas, o mokiniui pateikiamas sekantis. Jei atsitinka taip, kad mokinys nespėja atsakyti į klausimą, per duotą laiką, yra įjungiamas kitas klausimas, o nespėtas pilnai atsakyti klausimas tikrinamas tiek, kiek yra atsakyta.

Antras klausimas (30 pav.) – atvirkščias pirmajam. Čia reikia iš dvejetainio paversti į dešimtainį. Naudoju objektus *ComboBox*. Reikia pasirinkti viską – *skaičius, laipsnius, sandaugas*, ranka reikia įrašyti gautą *sumą*.

30 pav. Testo „Dvejetainiai skaičiai“ antras klausimas

Trečias klausimas (31 pav.) yra lengviausias, tereikia sužymėti 3 atsakymus. Norintys sukčiauti nusivils – jei pažymėtas atsakymas neteisingas, iš visų taškų, atimamas 1 taškas.

31 pav. Testo „Dvejetainiai skaičiai“ trečias klausimas



Ketvirtas klausimas (32 pav.) apie skaičiavimo sistemas. Reikia pažymėti, kuriai sistemai, kokie skaičiai (raidės) priklauso.

Testas "Dvejetainiai skaičiai"; 4 klausimas

Prašė: 22  
Liko: 98

[Kitas klausimas](#)

4 KLAUSIMAS. Parinkite kiekvienai skaičiavimo sistemai būdingus ženklus. Vertė: 4 taškai.

Dvejetainė sistema	Aštuntinė sistema	Dešimtainė sistema	Šešiolkinė sistema
<input type="radio"/> 1; 2; 3; 4; 5; 6; 7; 8;	<input checked="" type="radio"/> 1; 2; 3; 4; 5; 6; 7; 8;	<input checked="" type="radio"/> 1; 2; 3; 4; 5; 6; 7; 8; 9; 10;	<input type="radio"/> 1; 2; 3; 4; 5; 6; 7; 8;
<input type="radio"/> 2;	<input type="radio"/> 0; 1; 2; 3; 4; 5; 6; 7;	<input type="radio"/> 2;	<input type="radio"/> 2;
<input checked="" type="radio"/> 0; 1;	<input type="radio"/> 0; 1;	<input type="radio"/> 0; 1; 2; 3; 4; 5; 6; 7; 8; 9;	<input type="radio"/> 0; 1;
<input type="radio"/> 0; 1; 2;	<input type="radio"/> 0; 1; 2;	<input type="radio"/> 0; 1; 2;	<input type="radio"/> 0; 1; 2;
<input type="radio"/> 0; 1; 2; 3; 4; 5; 6; 7; 8; 9; A; B; C; D; E;	<input type="radio"/> 0; 1; 2; 3; 4; 5; 6; 7; 8; 9; A; B; C; D; E; F;	<input type="radio"/> 0; 1; 2; 3; 4; 5; 6; 7; 8; 9; A; B; C; D; E; F;	<input checked="" type="radio"/> 0; 1; 2; 3; 4; 5; 6; 7; 8; 9; A; B; C; D; E; F;

32 pav. Testo „Dvejetainiai skaičiai“ ketvirtas klausimas

Penktame klausime (33 pav.) reikia suskaičiuoti sumas dvejetainiame kode – į kiekvieną baltą langelį reikia įrašyti skaičių.

Testas "Dvejetainiai skaičiai"; 5 klausimas

Prašė: 68  
Liko: 232

[Baigti testą](#)

5 KLAUSIMAS. Suskaičiuokite sumas. Jei langelyje nieko neturi būti, įrašykite "0".  
Vertė: 14 taškų.

$\begin{array}{r} 1100011 \\ + 1011111 \\ \hline \end{array}$ <div style="display: flex; justify-content: center; gap: 10px;"> <div style="border: 1px solid gray; width: 20px; height: 20px; display: inline-block;"></div> <div style="border: 1px solid gray; width: 20px; height: 20px; display: inline-block;"></div> <div style="border: 1px solid gray; width: 20px; height: 20px; display: inline-block;"></div> <div style="border: 1px solid gray; width: 20px; height: 20px; display: inline-block;"></div> <div style="border: 1px solid gray; width: 20px; height: 20px; display: inline-block;"></div> <div style="border: 1px solid gray; width: 20px; height: 20px; display: inline-block;"></div> <div style="border: 1px solid gray; width: 20px; height: 20px; display: inline-block;"></div> <div style="border: 1px solid gray; width: 20px; height: 20px; display: inline-block;"></div> <div style="border: 1px solid gray; width: 20px; height: 20px; display: inline-block;"></div> <div style="border: 1px solid gray; width: 20px; height: 20px; display: inline-block;"></div> <div style="border: 1px solid gray; width: 20px; height: 20px; display: inline-block;"></div> <div style="border: 1px solid gray; width: 20px; height: 20px; display: inline-block;"></div> </div>	$\begin{array}{r} 10101 \\ + 1010 \\ \hline \end{array}$ <div style="display: flex; justify-content: center; gap: 10px;"> <div style="border: 1px solid gray; width: 20px; height: 20px; display: inline-block;"></div> <div style="border: 1px solid gray; width: 20px; height: 20px; display: inline-block;"></div> <div style="border: 1px solid gray; width: 20px; height: 20px; display: inline-block;"></div> <div style="border: 1px solid gray; width: 20px; height: 20px; display: inline-block;"></div> <div style="border: 1px solid gray; width: 20px; height: 20px; display: inline-block;"></div> <div style="border: 1px solid gray; width: 20px; height: 20px; display: inline-block;"></div> </div>
---	--

33 pav. Testo „Dvejetainiai skaičiai“ penktas klausimas

Kadangi raudona spalva traukia akį, tai tada, kai laiko lieka 10 sekundžių, likęs laikas yra nudažomas raudonai (34 pav.) Iš patirties žinau, kad tai paskatina sparčiau atsakinėti.

Testas "Dvejetainiai skaičiai"; 5 klausimas

Praėjo: 293  
Liko: 7

5 KLAUSIMAS. Suskaičiuokite sumas. Jei langelyje nieko neturi būti, įrašykite "0".  
Vertė: 14 taškų;

Baigti testą

$$\begin{array}{r} 1100011 \\ + 1011111 \\ \hline 11000110 \end{array}$$
$$\begin{array}{r} 10101 \\ + 1010 \\ \hline 011111 \end{array}$$

34 pav. Raudonas kritinis laikas

Kai testas baigiamas, pasirodo langas (35 pav.) kuriame parašytas mokinio *vardas*, *pavardė*, *surinkti taškai*, *pažymys*. Dešinėje pusėje parašyta kiek taškų už kuri klausimą mokinys surinko (ši informacija iš karto po testo baigimo, įrašoma į mokinio duomenų failą). Paspaudus mygtuką „Išeiti“, grįžtama į darbo pradžios langą.

Testas "Dvejetainiai skaičiai"; 5 klausimas

Testą laikė: A A  
Surinkote 24 iš 84  
Pažymys: 4

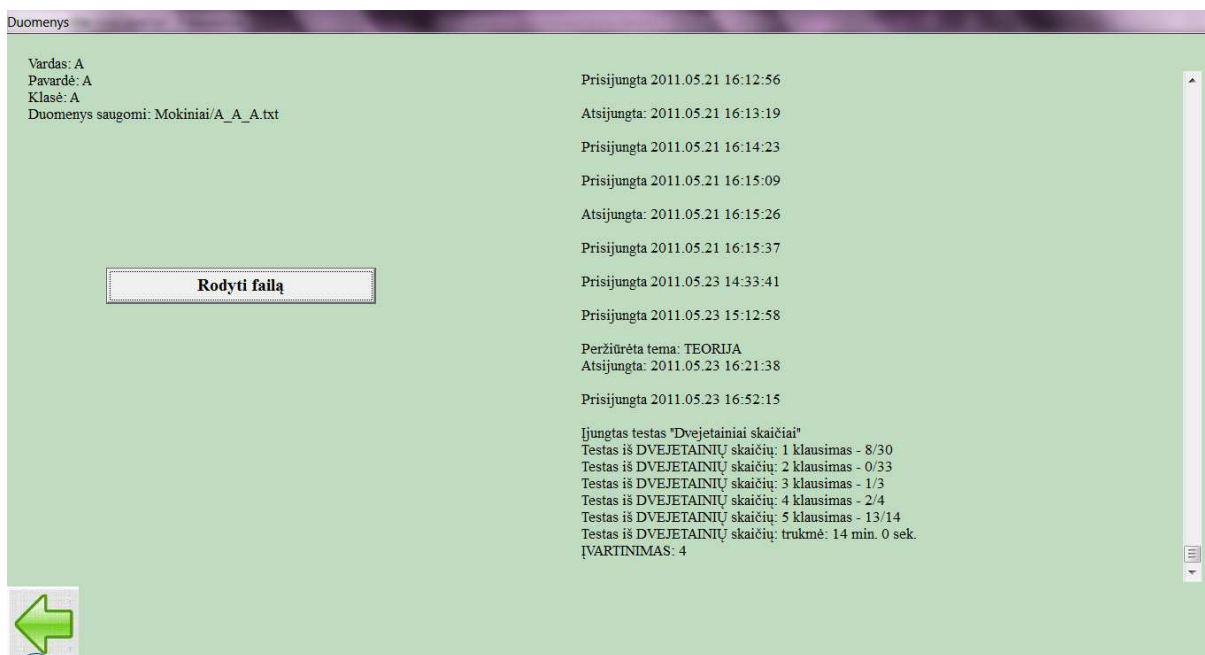
Išeiti

Gauti taškai:  
1kl.: 8  
2kl.: 0  
3kl.: 1  
4kl.: 0  
5kl.: 13

35 pav. Testo „Dvejetainiai skaičiai“ rezultatas

### 3.2.6. Meniu punkto „Mano duomenys“ realizavimas

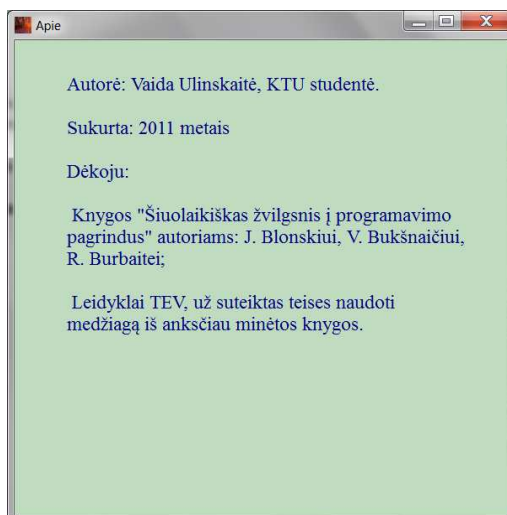
Per meniu juostą galime patekti į mokinio duomenų langą (36 pav.). Lange „Duomenys“ yra matoma mokinio informacija: *vardas, pavardė, klasė, dokumento pavadinimas*, kuriame saugoma visa informacija apie mokinį. Paspaudus mygtuką „Rodyti failą“, dešinėje pusėje yra pateikiamas dokumentas. Tekstinis dokumento pavyzdys pateiktas 2 priede.



36 pav. Mokinio duomenų langas

### 3.2.7. Meniu punkto „Apie“ realizavimas

Šiame meniu punkte (37 pav.) pateikta glausta informacija apie autorę. Trumpas padėkos žodis asmenims ir įstaigoms, sutikusioms pasidalinti jų turima autorine medžiaga.



37 pav. Meniu punkto „Apie“ langas

## 4. MOKOMOSIOS APLINKOS VARTOTOJO DOKUMENTACIJA

Mokomoji aplinka programavimo pradmenų įsisavinimui skirta mokiniams, pradedantiems mokytis programuoti. Aplinkoje pateikiama teorinė medžiaga apie pagrindinius programavimo elementus: programos struktūrą, programos kultūrą, dvejetainius skaičius, kintamuosius, skaitymo/rašymo sakinius, sąlygos sakinius, ciklus. Aplinkoje sudarytos galimybės mokiniams įtvirtinti žinias praktikos pagalba, taip pat yra kontrolinė dalis, kuria gali pasinaudoti ir mokytojas, norėdamas įvertinti mokinių žinias.

Aplinka yra lietuvių kalba, todėl norint kažką atlikti, nereikia daug galvoti – viskas parašyta, tereikia sekti instrukcijas.

- a) Tik įjungus aplinką, prašoma įvesti vardą, pavardę ir klasę.
- b) Atsivėrus pagrindiniam aplinkos langui, rekomenduojama perskaityti aplinkos naudojimo instrukciją, kuria vadovaujantis, darbas su aplinkas bus aiškus ir patogus.

Mokytojas, noėdamas pažiūrėti, kaip sekasi jo mokiniams, turi atsidaryti programos aplanką ir ten susirasti aplanką „Mokiniai“.

## 5. MOKOMOSIOS APLINKOS PANAUDOJIMAS MOKYMO PROCESSE

### 5.1. Mokomosios aplikos funkcijos ir galimybės

Mokomoji aplinka yra skirta programavimo pagrindų mokymuisi savarankiškai ir su mokytojo pagalba. Aplinka orientuota į pradedančiuosius programuotojus, kurie dar neturi jokių pagrindų. Pagrindinės funkcijos ir galimybės:

- ✓ Vaizdžiai, įdomiai, aiškiai pateikti mokomąją medžiagą;
- ✓ Besimokant, žinias įtvirtinti pratybomis;
- ✓ Atlikti testus;
- ✓ Viskas vienoje vietoje, be ypatingų reikalavimų kompiuteriui.

### 5.2. Mokomosios aplinkos įdiegimas

1. Išskleisti archyvą;
2. Aplanke su failais ir kitais aplankais rasti failą *Mokykis\_programuoti.exe* ir jį paleisti.

### 5.3. Mokomosios aplinkos testavimas

#### 5.3.1. Pirminis testavimas

Sukurta aplinka buvo testuojama su įvairiomis operacinėmis sistemomis. Pastebėta, kad iškilo problemų dėl šriftų. Kai kurios operacinės sistemos (tarp jų ir WindowsXP) neatpažindavo šriftų ir tekstai buvo pateikiami nesuprantamais simboliais. Ši problema yra išspręsta – visa aplinka sukurta naudojant standartinį šriftą TimeNewRoman.

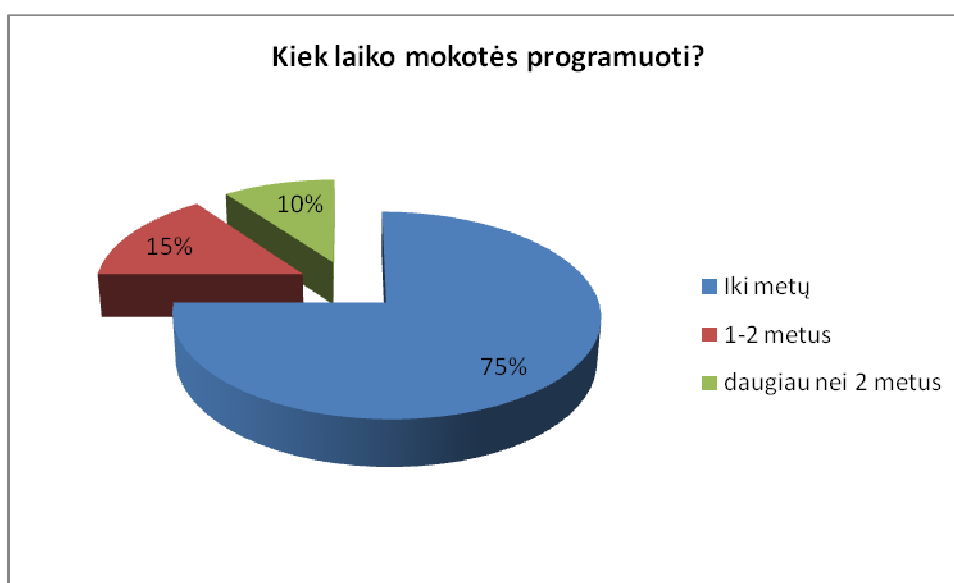
Atliekant testavimą, buvo pastebėta ir keletas programavimo klaidų. Atlikus veiksmus, kurie kuriant nebuvo apgalvoti, neveikė testų laikmatis, neišsivalydavo langai ir pan. Visos problemos kurios buvo pastebėtos, buvo ištaisytos.

#### 5.3.2. Testavimas mokykloje

Mokomoji aplinka buvo testuojama Šakių raj. Lukšių Vinco Grybo gimnazijoje. Testavimui buvo patrenktos anketos informacinių technologijų mokytojų ir mokinių nuomonės tyrimui (žr. 5, 6 prieduose).

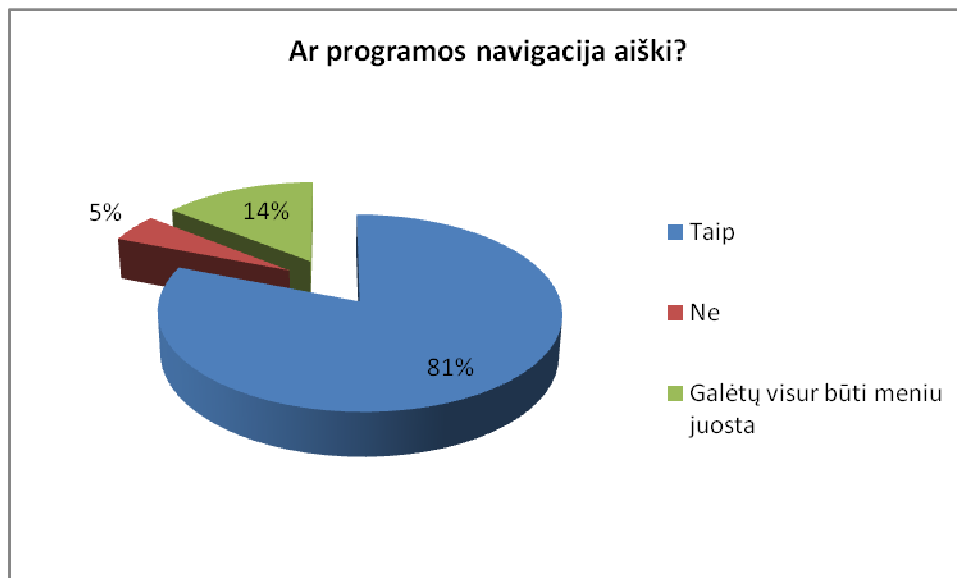
Į anketas buvo įtraukti klausimai apie sukurta mokomosios aplinkos dalį. Klausiau mokinių ir mokytojo, ar aiškus medžiagos pateikimas aplinkoje, ar aiški aplinkos navigacija, ar gali tokia aplinka pagelbėti besimokant programavimo pagrindų.

Apklausoje dalyvavo 21 mokiny besimokantis 10 klasėje. Į klausimą, kiek mokosi programuoti, 15 mokinių (75%) atsakė, kad mokosi dar trumpai (mažiau nei metus). Tik 10% programuoti mokosi ilgiau nei 2 metus (38 pav.).



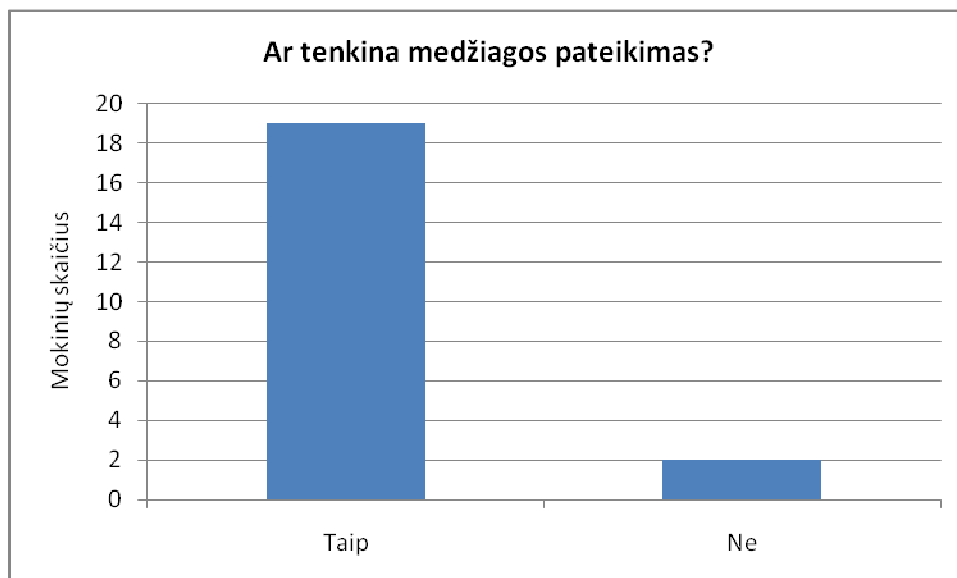
38 pav. Kiek laiko mokotės programuoti?

Mokiniai pabandę dirbti su mokomąja aplinka programavimo pagrindų įsisavinimui, gavo užduotį, įvertinti, kai kuriuos aplinkos aspektus. Vienas tokių aspektų – ar aiški aplinkos navigacija, t.y. ar patogiu pereiti iš vieno lango į kitą. Net 17 mokinių (81%) nesusidūrė su jokiais navigacijos problemomis, 3 mokiniai pageidauja, kad visoje aplinkoje, navigacija būtų vienoda – meniu juosta lango viršuje (39 pav).



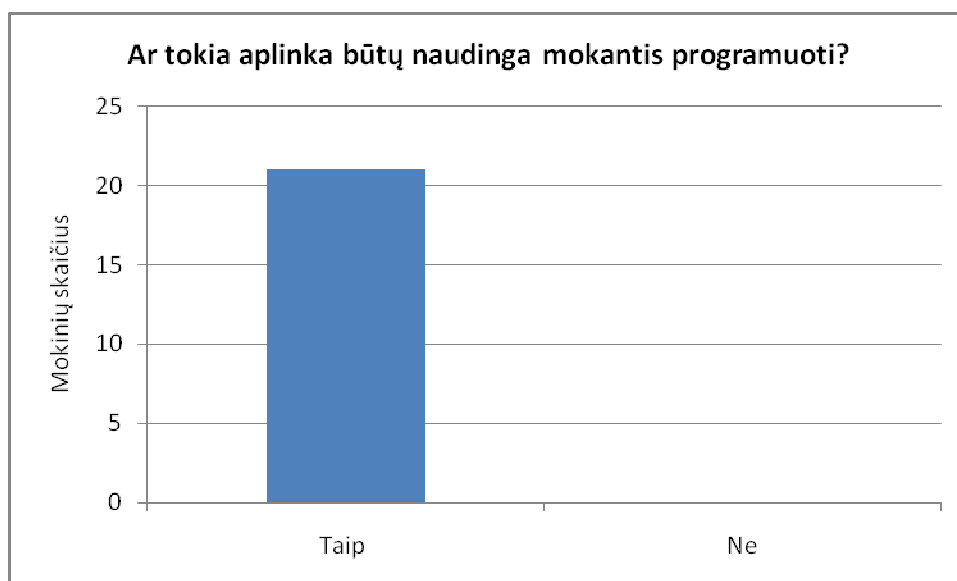
**39 pav. Ar mokomosios aplinkos navigacija aiški?**

Daugumai apklaustųjų patiko ir medžiagos pateikimo būdai aplinkoje. 19 mokinių pritarė tokiam būdai (40 pav.)



**40 pav. Ar Jus tenkina mokomosios medžiagos pateikimas aplinkoje**

Net ir rasdami didesnių ar mažesnių trūkumų, visi mokiniai teigia, kad tokia aplinka būtų naudinga mokantis programavimo pagrindų (41 pav.).



41 pav. Ar tokia aplinka būtų naudinga mokantis programuoti?

Lukšių Vinco Grybo gimnazijos informacinių technologijų mokytojas vietoj anketos, pateikė savo atsiliepimą, kuris pateiktas 8 priede.

### 5.3.3. Aplinkos testavimo apibendinimas

Atlikus pirminį testavimą, rastos klaidos yra pataisytos. Atliktas testavimas mokykloje parodė, kad mokiniai ir mokytojas yra patenkinti kuriamos aplinkos funkcijomis, medžiagos pateikimu, testų forma, aiškumu.

Smulkios klaidos (rašybos, programinės) buvo pataisytos iš karto, rimtesni pasiūlymai dar svarstomi, apmąstomi ir bus įgyvendinti.

Atlikusi šiuos tyrimus tik dar labiau įsitikinau, kad ir mokiniams ir mokytojams tokia aplinka labai reikalinga.

## IŠVADOS

1. Atlikta mokytojų ir mokinių poreikių tyrimas, po kurio paaiškėjo apklaustųjų poreikiai ir lūkesčiai kuriamai mokomajai aplinkai.
2. Išanalizuotos esamos mokomosios aplinkos, laisvai prieinamos internete. Nustatyti šių aplinkų privalumai ir trūkumai. Į tai atsižvelgta kuriant mokomąją aplinką.
3. Atlikta analizė, kokia turi būti kuriama mokomoji aplinka, nustatyta jos struktūra. Tai padeda lengviau realizuoti kūrinį.
4. Suprojektuota ir realizuota dalis mokomosios aplinkos. Realizuotos tokios dalys, kurios atspindi būsimos visos mokomosios aplinkos bruožus – koku būdu pateikiama informacija, kaip vykdomi testai ir pan.
5. Sukurta vartotojo dokumentacija, kuri trumpai apibūdina, kam yra skirta mokomoji aplinka, kaip ja naudotis.
6. Atliktas mokomosios aplinkos testavimas mokykloje. Jo metu nustatytos smulkios klaidos pataisytos. Gauta nemažai naudingų pastabų, kaip galima dar tobulinti mokomąją aplinką.



## LITERATŪROS SĄRAŠAS

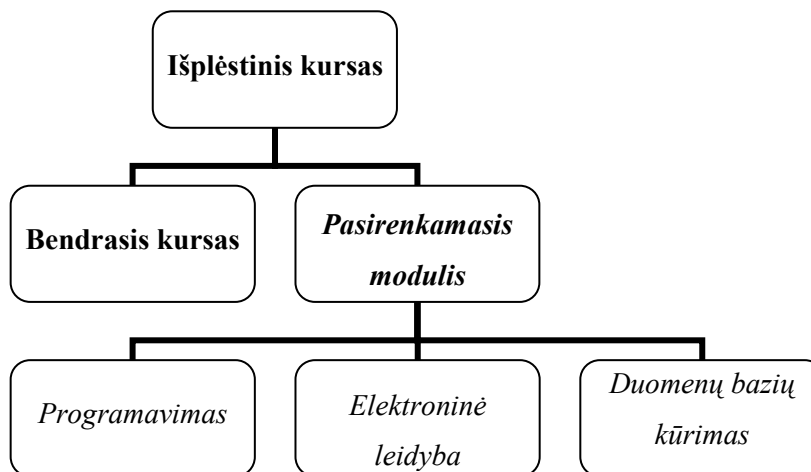
1. Joana Lipeikienė, Tomas Petkus „Informatikos mokytojų virtualios mokomosios aplinkos kūrimas ir tyrimas“ ISSN 1392-0561. INFORMACIJOS MOKSLAI. 2007 42–43
2. Valentina Dagienė, Jūratė Urbonienė, „Programavimo mokymasis: lyginamoji kalbos ir aplinkos analizė“, ISSN 1392-0561. INFORMACIJOS MOKSLAI. 2010 54
3. VIDURINIO UGDYMO BENDROSIOS PROGRAMOS. Patvirtintos ministro 2011 m. vasario 21 d. įsakymu Nr. V-269
4. Jonas Blonskis, Vytautas Bukšnaitis, Renata Burbaitė „ŠIUOLAIKIŠKAS ŽVILGSNIS Į PROGRAMAVIMO PAGRINDUS“ Informacinių technologijų vadovėlis pasirenkamajam kursui IX–X klasėse. 2010
5. Šaltenis V. „OBJEKTINIO PROGRAMAVIMO METODAI (Delphi programavimo terpėje)“, Vilnius, 2004.
6. [www.freewebs.com/pascalpamokos/](http://www.freewebs.com/pascalpamokos/) Mokomoji programavimo aplinka internete
7. <http://ifreal.com/category/pascal.html> Mokomoji programavimo aplinka internete
8. <http://programavimopamokos.net/index.php?vaizdas=FPSPamokos> Mokomoji programavimo aplinka internete
9. <http://83.171.17.211/intronet/informatika/vadoveliai/delphi/atsakymai.htm> Pagalba programuojantiems Delpfi. Čia pateikti atsakymai į beveik visus klausimus.
10. [http://vmc.ppf.ktu.lt/saulius/v1/inf\\_03\\_aritmetika/ar1\\_sudetis/win01.html](http://vmc.ppf.ktu.lt/saulius/v1/inf_03_aritmetika/ar1_sudetis/win01.html)

Autorius: Sinkevičius Saulius; 2001 m. išleista „*Turbo Pascal 7.0 trumpai. Žinynas*“ knyga. Ši knyga gavo grifą, t. y. ji yra rekomenduojama Lietuvos Respublikos švietimo ir mokslo ministerijos. Recenzavo dr. Jonas Blonskis, dr. Valentina Dagienė, mok. metodininkas Arvydas Verseckas [4]

# 1 PRIEDAS. VIDURINIO UGDYMO BENDRŪJŲ PROGRAMŲ 7 PRIEDAS (IŠTRAUKOS)

<...>

4.7. Išplėstinio kurso kiekvieno modulio programą sudaro kelios teminės dalys (veiklos sritys), tačiau dėstant temų eilės tvarkos nebūtina laikytis. Kai kurios temos gali būti išdėstytos kartu, keičiant tvarką ir pan.



**2 schema.** Informacinių technologijų išplėstinio kurso struktūra

4.8. Pasirenkamųjų modulių ugdymo turinys parengtas taip, kad nė vienas iš jų nėra susietas su konkrečia programine įranga. Mokytojas laisvas pasirinkti programinę įrangą atsižvelgdamas į savo ir mokinių pasirengimą, mokinių pageidavimą, mokyklos galimybes ir pan.

4.9. Išplėstinis kursas ugdo mokinio informacinę komunikacinę kompetenciją, padeda plėtoti gebėjimus naudotis informacinėmis ir komunikacinėmis technologijomis pagal pasirinktą mokymosi kryptį. Kursas padeda mokiniui pasirengti spręsti su mokymusi, laisvalaikio susijusius uždavinius, taip pat renkantis būsimą profesiją, suteikia galimybę susipažinti su įvairesnėmis technologijomis.

<...>

## **11. Išplėstinis kursas. Programavimas**

### **11.1. Mokinių pasiekimai. Išplėstinis kursas. Programavimas**

11.1.1. Šiame skyriuje aprašomi išplėstinio kurso *Programavimo* modulio mokinių pasiekimams keliami reikalavimai. Lentelėje aprašoma, kokios turi būti mokinių žinios ir supratimas, kokie ugdomi gebėjimai; vėliau nurodoma turinio apimtis: užrašoma tema ir

atskleidžiama jos apimtis. Skyriaus pabaigoje pateikiamas mokinių pasiekimų lygių požymių aprašas.

11.1.2. Šioje lentelėje aprašomi mokinių pasiekimai: nuostatos, gebėjimai, žinios ir supratimas.

<i><b>Programavimas</b></i>	
<b>Mokinių pasiekimai</b>	
<p><b>Nuostatos</b></p> <p><i>Planuoti savo veiklą, pasirinkti uždavinius ir apgalvoti jų sprendimo būdus, numatyti ir įvertinti rezultatus.</i></p> <p><b>Esminis gebėjimas:</b> kurti programas uždaviniams spręsti: parinkti, pritaikyti, sudaryti uždavinių algoritmus, užrašyti juos programavimo kalba, atlikti kompiuteriu, testuoti, apipavidalinti rezultatus.</p>	
<b>Gebėjimai</b>	<b>Žinios ir supratimas</b>
<p>1.1. Analizuoti užduotį: įvertinti jos sudėtingumo lygį, skaidyti į dalis, nustatyti ryšius tarp šių dalių, parengti sprendimo eskizą (schemą).</p>	<p>1.1.1. Analizuoti užduotį, skaidyti ją į struktūrines dalis.</p> <p>1.1.2. Apibūdinti užduoties skaidymo dalimis pranašumus.</p> <p>1.1.3. Aprašyti sprendimą kuriuo nors formaliu būdu (schemomis, programavimo kalbos žymenimis).</p> <p>1.1.4. Parinkti sprendžiamam uždaviniui tinkamas duomenų struktūras, sprendimo metodus ir algoritmus.</p>
<p>1.2. Pradiniams duomenims ir rezultatams laikyti (saugoti) naudoti tekstinius failus.</p>	<p>1.2.1. Apibūdinti darbo su dideliais duomenų rinkiniais specifiką.</p> <p>1.2.2. Nusakyti tekstinių failų paskirtį.</p> <p>1.2.3. Parengti failą duomenų įvedimui ir rašymui.</p> <p>1.2.4. Skaityti duomenis iš failo.</p> <p>1.2.5. Rašyti duomenis į failą.</p>

<b>Programavimas</b>	
<b>Mokinių pasiekimai</b>	
<p><b>Nuostatos</b></p> <p><i>Planuoti savo veiklą, pasirinkti uždavinius ir apgalvoti jų sprendimo būdus, numatyti ir įvertinti rezultatus.</i></p> <p><b>Esminis gebėjimas:</b> kurti programas uždaviniams spręsti: parinkti, pritaikyti, sudaryti uždavinių algoritmus, užrašyti juos programavimo kalba, atlikti kompiuteriu, testuoti, apipavidalinti rezultatus.</p>	
<b>Gebėjimai</b>	<b>Žinios ir supratimas</b>
<p>1.3. Aprašyti veiksmus procedūromis arba funkcijomis, taikyti jas sudarant programas.</p>	<p>1.3.1. Paaiškinti procedūros ir / ar funkcijos sąvoką, paskirtį.</p> <p>1.3.2. Išskirti uždavinio dalis, kurias galima užrašyti procedūromis arba funkcijomis.</p> <p>1.3.3. Apibūdinti parametrus, naudoti juos rašant procedūras arba funkcijas.</p> <p>1.3.4. Rašant programas naudoti procedūras ir / ar funkcijas.</p> <p>1.3.5. Nusakyti programos, procedūrų ir / ar funkcijų vardų, parametrų ir kintamųjų galiojimo sritis.</p> <p>1.3.6. Paaiškinti procedūrų ir / ar funkcijų parametrų ir kreipinių argumentų ryšį.</p>
<p>1.4. Suformuoti masyvą ir atlikti veiksmus su jo duomenimis.</p>	<p>1.4.1. Apibūdinti masyvo sąvoką.</p> <p>1.4.2. Skirti masyvo elemento indeksą ir masyvo elemento reikšmę.</p> <p>1.4.3. Aprašyti skaičių ir simbolių masyvo duomenų tipą.</p> <p>1.4.4. Aprašyti masyvo duomenų tipo kintamuosius.</p> <p>1.4.5. Suteikti masyvo elementams reikšmes (priskyrimas, skaitymas iš failo).</p> <p>1.4.6. Atlikti veiksmus su masyvo elementų reikšmėmis.</p> <p>1.4.7. Naudoti masyvo tipo parametrus procedūrose arba funkcijose.</p>
<p>1.5. Atlikti veiksmus su simbolių eilutėmis.</p>	<p>1.5.1. Apibūdinti simbolių eilutės sąvoką.</p> <p>1.5.2. Aprašyti simbolių eilutės duomenų tipo kintamuosius.</p> <p>1.5.3. Atlikti veiksmus su simbolių eilučių kintamųjų reikšmėmis.</p>

<i><b>Programavimas</b></i>	
<b>Mokinių pasiekimai</b>	
<p><b>Nuostatos</b></p> <p><i>Planuoti savo veiklą, pasirinkti uždavinius ir apgalvoti jų sprendimo būdus, numatyti ir įvertinti rezultatus.</i></p> <p><b>Esminis gebėjimas:</b> kurti programas uždaviniams spręsti: parinkti, pritaikyti, sudaryti uždavinių algoritmus, užrašyti juos programavimo kalba, atlikti kompiuteriu, testuoti, apipavidalinti rezultatus.</p>	
<b>Gebėjimai</b>	<b>Žinios ir supratimas</b>
<p>1.6. Atlikti veiksmus su įrašais (struktūromis) ir masyvais su įrašo (struktūros) tipo elementais.</p>	<p>1.6.1. Apibūdinti įrašo (struktūros) sąvoką, nusakyti paskirtį.</p> <p>1.6.2. Aprašyti įrašo (struktūros) duomenų tipą.</p> <p>1.6.3. Aprašyti įrašo (struktūros) duomenų tipo kintamuosius.</p> <p>1.6.4. Suteikti įrašo (struktūros) tipo kintamiesiems reikšmes (įvesti, priskirti duomenis), išvesti.</p> <p>1.6.5. Atlikti veiksmus su įrašo (struktūros) tipo kintamųjų reikšmėmis.</p> <p>1.6.6. Atlikti veiksmus su įrašo (struktūros) tipo masyvo elementų reikšmėmis.</p> <p>1.6.7. Sudaryti nesudėtingas duomenų struktūras, panaudoti įrašo (struktūros) ir masyvo duomenų tipus.</p>
<p>1.7. Sudaryti uždavinių programas.</p>	<p>1.7.1. Parinkti uždaviniui spręsti tinkamas duomenų struktūras ir algoritmus.</p> <p>1.7.2. Programose naudoti algoritmus: sumos, sandaugos, kiekio, aritmetinio vidurkio skaičiavimo; rikiavimo; didžiausios (mažiausios) reikšmės radimo, paieškos surikiuotame ir nerikiuotame masyve, masyvo papildymo naujomis reikšmėmis, reikšmių pašalinimo iš masyvo.</p> <p>1.7.3. Parinktą, modifikuotą ar sudarytą algoritmą užrašyti programavimo kalba.</p> <p>1.7.4. Kurti programas įvairiems taikomiesiems uždaviniams spręsti.</p>

<i>Programavimas</i>	
<b>Mokinių pasiekimai</b>	
<p><b>Nuostatos</b></p> <p><i>Planuoti savo veiklą, pasirinkti uždavinius ir apgalvoti jų sprendimo būdus, numatyti ir įvertinti rezultatus.</i></p> <p><b>Esminis gebėjimas:</b> kurti programas uždaviniams spręsti: parinkti, pritaikyti, sudaryti uždavinių algoritmus, užrašyti juos programavimo kalba, atlikti kompiuteriu, testuoti, apipavidalinti rezultatus.</p>	
<b>Gebėjimai</b>	<b>Žinios ir supratimas</b>
<p>1.8. Įvaldyti programavimo technologijos (metodikos) elementus.</p>	<p>1.8.1. Įvardyti programos sudarymo etapus, laikytis jų rašant programas.</p> <p>1.8.2. Parengti kontrolinius duomenis programos darbo teisingumui patikrinti.</p> <p>1.8.3. Skirti programos darbo patikrinimą ir programos testavimą.</p> <p>1.8.4. Komentuoti programos dalis.</p>

## **11.2. Turinio apimtis. Išplėstinis kursas. Programavimas**

**11.2.1. Algoritmai.** Primenami sumos, sandaugos, kiekio, aritmetinio vidurkio skaičiavimo, didžiausios (mažiausios) reikšmės paieškos algoritmai. Supažindinama su rikiavimo, paieškos surikiuotame ir nerikiuotame masyve, masyvo reikšmių šalinimo iš masyvo, masyvo papildymo naujomis reikšmėmis algoritmais. Mokoma užrašyti loginius reiškinius ir juos panaudoti masyvo rikiavimo, paieškos, reikšmių šalinimo ir papildymo veiksmuose, kai duomenų tipas yra masyvas ir / ar įrašas (struktūra). Mokoma užrašyti algoritmus programavimo kalbos žymenimis ir taikyti darbui su duomenimis, saugomais skirtingo tipo duomenų kintamuosiuose. Mokiniai skatinami kurti algoritmus nesudėtingiems skaičiavimams, įvairioms taikomiesiems uždaviniams spręsti.

**11.2.2. Tekstiniai failai.** Paaiškinama tekstinių failų paskirtis, tekstinio failo tipo kintamieji. Mokoma failus paruošti skaitymui bei rašymui, skaityti iš tekstinių failų ir į juos rašyti paprastuosius, masyvo bei įrašų tipų duomenis. Mokoma užrašyti tai programavimo kalbos žymenimis, taikyti programose.

**11.2.3. Procedūros ir / ar funkcijos.** Apibūdinama procedūrų ir / ar funkcijų paskirtis programoje. Paaiškinama, kaip jos aprašomos. Mokoma tinkamai užrašyti aprašymą, struktūrą, parametrus. Paaiškinama, kaip užrašomas kreipinys, kaip duomenys perduodami procedūrai ir / ar funkcijai, kaip grąžinami skaičiavimo rezultatai. Mokoma apibrėžti vardų

galiojimo sritis. Mokoma užrašyti šias konstrukcijas programavimo kalbos žymenimis. Mokiniai skatinami skaidyti programą dalimis, kurios apiforminamos procedūromis ir / ar funkcijomis.

**11.2.4. Vienmatis masyvas.** Paaiškinamos masyvo, masyvo elemento indekso ir reikšmės sąvokos. Mokoma aprašyti masyvo duomenų tipą, jį naudoti kuriant masyvo tipo kintamuosius. Mokoma priskirti reikšmes masyvo elementams, skaityti duomenis iš failo į masyvą bei juos išvesti į failą, atlikti veiksmus su masyvo reikšmėmis. Paaiškinama, kaip masyvai perduodami procedūros (funkcijos) parametrais. Mokiniai skatinami taikyti žinomus ir kurti nesudėtingus algoritmus darbui su masyvo elementų reikšmėmis.

**11.2.5. Simbolių eilutės.** Paaiškinamos simbolių eilutės, duomenų tipo ir eilutės tipo kintamųjų sąvokos. Apibūdinamas simbolių eilučių masyvo duomenų tipas ir kintamieji. Mokoma atlikti veiksmus su simbolių eilutėmis: įvesti, išvesti, palyginti, sujungti. Mokiniai skatinami taikyti žinomus algoritmus atliekant veiksmus su simbolių eilučių tipo masyvo elementų reikšmėmis.

**11.2.6. Įrašas (struktūra).** Paaiškinama įrašo (struktūros) prasmė ir taikymas. Mokoma sukurti įrašo (struktūros) duomenų tipą ir jį aprašyti. Mokoma sukurti masyvo tipą su įrašo (struktūros) elementais, aprašyti įrašo (masyvo) tipo kintamuosius, masyvus su įrašo tipo reikšmėmis, atlikti veiksmus su įrašo tipo duomenimis. Paaiškinama, kaip įrašo (struktūros) tipo kintamieji perduodami procedūros (funkcijos) parametrais. Mokoma, kaip įrašo (struktūroje) panaudoti jau sukurtus kitus įrašus (struktūras) bei masyvus. Mokiniai skatinami taikyti žinomus ir kurti nesudėtingus algoritmus darbui su įrašo (masyvo) elementų reikšmėmis.

**11.2.7. Programavimo technologija.** Pabrėžiami programos sudarymo etapai: rašymas, derinimas, patikrinimas, testavimas, tobulinimas. Mokiniai skatinami sudarant programas laikytis šių etapų. Paaiškinama kontrolinių duomenų sąvoka ir svarba programos teisingumui užtikrinti.

Mokytojas atkreipia dėmesį į programos stilių ir kalbos kultūrą. Nagrinėjami konkretūs pačių mokinių sudarytų programų pavyzdžiai. Mokytojas paaiškina, kur ir kokius komentarus rašyti, kaip geriau išdėstyti vieną ar kitą sakinį, kokius vardus geriau parinkti programoje, ir pan. Mokiniai mokomi rašyti programas tvarkingai, skatinami laikytis programavimo kultūros.

<...>

## 2 PRIEDAS. VARTOTOJO DUOMENŲ FAILO PAVYZDYS

Mokinio duomenų failo *VAIDA\_ULINSKAITĖ\_12A.txt* ištrauka:

<...>

Prisijungta 2011.01.24 03:39:23

Įjungtas testas "Dvejetainiai skaičiai"

Testas iš DVEJETAINIŲ skaičių: 1 klausimas - 29/30

Testas iš DVEJETAINIŲ skaičių: 2 klausimas - 33/33

Testas iš DVEJETAINIŲ skaičių: 3 klausimas - 3/3

Testas iš DVEJETAINIŲ skaičių: 2 klausimas - 33/33

Testas iš DVEJETAINIŲ skaičių: 3 klausimas - 3/3

Testas iš DVEJETAINIŲ skaičių: 4 klausimas - 4/3

Testas iš DVEJETAINIŲ skaičių: 5 klausimas - 14/14

Testas iš DVEJETAINIŲ skaičių: trukmė: 9 min. 33 sek.

Testas iš DVEJETAINIŲ skaičių:

Atsijungta: 2011.01.24 03:46:47

Prisijungta 2011.01.24 03:47:05

Atsijungta: 2011.01.24 03:48:21

Prisijungta 2011.01.24 03:51:42

Įjungtas testas "Dvejetainiai skaičiai"

Testas iš DVEJETAINIŲ skaičių: 1 klausimas - 5/30

Testas iš DVEJETAINIŲ skaičių: 2 klausimas - 5/33

Testas iš DVEJETAINIŲ skaičių: 3 klausimas - 3/3

Testas iš DVEJETAINIŲ skaičių: 4 klausimas - 4/3

Testas iš DVEJETAINIŲ skaičių: 5 klausimas - 14/14

Testas iš DVEJETAINIŲ skaičių: trukmė: 2 min. 58 sek.

Testas iš DVEJETAINIŲ skaičių: 5

Atsijungta: 2011.01.24 03:54:40

Prisijungta 2011.01.24 04:04:34

Pamoka: DVEJETAINIŲ SKAIČIŲ SUDĖTIS

Pamoka: DVEJETAINIŲ SKAIČIŲ SUDĖTIS

Atsijungta: 2011.01.24 04:06:45

Prisijungta 2011.01.24 07:28:37

Peržiūrėta tema: TEORIJA

Prisijungta 2011.01.24 07:58:58

Įjungtas testas "Dvejetainiai skaičiai"

Testas iš DVEJETAINIŲ skaičių: 1 klausimas - 10/30

Testas iš DVEJETAINIŲ skaičių: 2 klausimas - 0/33

Testas iš DVEJETAINIŲ skaičių: 3 klausimas - 0/3

Testas iš DVEJETAINIŲ skaičių: 2 klausimas - 0/33

Testas iš DVEJETAINIŲ skaičių: 3 klausimas - 0/3

Testas iš DVEJETAINIŲ skaičių: 4 klausimas - 0/4

Testas iš DVEJETAINIŲ skaičių: 5 klausimas - 14/14

Testas iš DVEJETAINIŲ skaičių: trukmė: 11 min. 46 sek.

Testas iš DVEJETAINIŲ skaičių: 5



### 3 PRIEDAS. MOKYTOJŲ POREIKIŲ PROGRAMAVIMO MOKYME TYRIMO ANKETA

Sveiki. Esu Kauno Technologijos Universiteto magistrantė. Magistriniam darbui rašau programą, skirtą mokyti (-is) programavimo pagrindų. Šiuo metu atlieku tyrimą apie tokios programos poreikį informacinių technologijų mokytojams. Maloniai prašau atsakyti į keletą klausimų. Anketa yra anoniminė, duomenys bus naudojami tik statistikai.

1. **Jūsų amžius:** \_\_\_\_\_
2. **Lytis:** \_\_\_\_\_
3. **Kategorija:** \_\_\_\_\_
4. **Pedagoginio darbo stažas:** \_\_\_\_\_
5. **Ugdymo institucijos, kurioje dirbate, tipas:** \_\_\_\_\_
6. **Gyvenamoji vieta (rajono centras, miestelis, kaimas):** \_\_\_\_\_
7. **Ar mokote vaikus programavimo?** (Jei atsakėte „Taip“, pereikite prie 12 klausimo, jei atsakėte „Ne“, atsakykite į 8 ir 9 klausimus ir viskas)
  - Taip;
  - Ne;
8. **Dėl kokių priežasčių nemokote?**  
\_\_\_\_\_  
\_\_\_\_\_
9. **Ar pradėtumėte mokyti, jei būtų sukurta programa, kur galima būtų rasti esminius dalykus apie programavimo pagrindus? Kokia informacija turėtų ten būti?**  
\_\_\_\_\_  
\_\_\_\_\_
10. **Ar programuoti mokote pamokų metu, ar neformalaus ugdymo metu?**
  - Tik per pamokas;
  - Tik neformalaus ugdymo užsiėmimų metu;
  - Ir per pamokas, ir neformalaus ugdymo metu;
11. **Kokias pagalbines priemones naudojate programavimo mokymui?**
  - Vadovėlius;
  - Ieškau informacijos (įdomesnių užduočių) internete;
  - Nieko nenaudoju – viskas, ko reikia, yra mano galvoje;
  - Kita (įrašykite)
12. **Jūsų nuomone, ar būtų reikalinga programa, kuria galėtų mokiniai naudotis savarankiškai, kuri pasitarnautų vertinant mokinių žinias (pamokų metu);**
  - Taip, reikalinga;
  - Nereikalinga;
13. **Kokias temas mokiniams yra sudėtingiausia išaiškinti, reikia daug pavyzdžių?**  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_
14. **Kokia programavimo kalba, Jūsų nuomone, labiausiai tinka programavimo mokymui (pagrindams)** \_\_\_\_\_

Dėkoju už skirtą laiką.

## 4 PRIEDAS. MOKINIŲ POREIKIŲ TYRIMO ANKETA

Sveiki. Esu Kauno Technologijos Universiteto magistrantė. Magistriniam darbui rašau programą, skirtą mokyti (-is) programavimo pagrindų. Šiuo metu atlieku tyrimą apie tokios programos poreikį 8-10 klasių mokiniams. Maloniai prašau atsakyti į keletą klausimų. Anketa yra anoniminė, duomenys bus naudojami tik statistikai.

1. Jūsų amžius: \_\_\_\_\_
2. Lytis: \_\_\_\_\_
3. Klasė: \_\_\_\_\_
4. Mokyklos tipas: \_\_\_\_\_
5. Gyvenamoji vieta (rajono centras, miestelis, kaimas): \_\_\_\_\_
6. Ar mokotės programavimo? (Jei atsakėte „Taip“, pereikite prie 9 klausimo, jei atsakėte „Ne“, atsakykite į 7 ir 8 klausimus ir viskas)  
 Taip;  
 Ne;
7. Dėl kokių priežasčių nesimokote?  
\_\_\_\_\_  
\_\_\_\_\_

8. Ar pradėtumėte mokytis, jei būtų sukurta programa, kur galima būtų rasti esminius dalykus apie programavimo pagrindus. Būtų galimybė pasimokyti savarankiškai.

9. Ar programuoti mokotės pamokų metu, ar neformalaus ugdymo metu?  
 Tik per pamokas;  
 Tik neformalaus ugdymo užsiėmimų metu;  
 Ir per pamokas, ir neformalaus ugdymo metu;
10. Kokias pagalbines priemones naudoja mokytojas (-a) programavimo mokymui?  
 Vadovėlius;  
 Ieško informacijos (įdomesnių užduočių) internete;  
 Nieko nenaudoja – viskas, ko reikia, yra mokytojo galvoje;  
 Kita (įrašykite)

11. Jūsų nuomone, ar būtų reikalinga programa, kuria galėtų mokiniai naudotis savarankiškai, kuri pasitarnautų mokytojui vertinant mokinių žinias (pamokų metu);

- Taip, reikalinga;  
 Nereikalinga;

12. Kokias temas yra sudėtingiausia išmokyti, mokantis programuoti?  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Dėkoju už skirtą laiką.

## 5 PRIEDAS. PROGRAMOS VERTINIMO ANKETA (MOKYTOJŲ)

Sveiki. Esu Kauno Technologijos Universiteto magistrantė. Magistriniam darbui rašau programą, skirtą mokyti (-is) programavimo pagrindų. Šiuo metu atlieku tyrimą apie tokios programos poreikį informacinių technologijų mokytojams. Maloniai prašau atsakyti į keletą klausimų. Anketa yra anoniminė, duomenys bus naudojami tik statistikai.

1. Jūsų amžius: \_\_\_\_\_
2. Lytis: \_\_\_\_\_
3. Kategorija: \_\_\_\_\_
4. Pedagoginio darbo stažas: \_\_\_\_\_
5. Ugdymo institucijos, kurioje dirbate, tipas:  
\_\_\_\_\_
6. Gyvenamoji vieta (rajono centras, miestelis, kaimas): \_\_\_\_\_
7. Ar programos navigacija aiški ir patogi:

Taip;

Ne;

Pasiūlymai:

---

---

---

8. Ar tenkina medžiagos pateikimas ir išsamumas?

Taip;

Ne;

Pasiūlymai:

---

---

---

9. Ar gali tokia programa (pilnai užbaigta) būti naudinga mokantis programuoti (pradedandiesiems)

Taip;

Ne;

Trūkumai: \_\_\_\_\_

---

---

---

10. Bendri pasiūlymai ir pastabos (trūkumai ir privalumai) \_\_\_\_\_

---

---

---

---

Dėkoju už skirtą laiką.

## 6 PRIEDAS. PROGRAMOS VERTINIMO ANKETA (MOKINIŲ)

Sveiki. Esu Kauno Technologijos Universiteto magistrantė. Magistriniam darbui rašau programą, skirtą mokytis (-is) programavimo pagrindų. Šiuo metu atlieku tyrimą apie tokios programos poreikį 8-10 klasių mokiniams. Maloniai prašau atsakyti į keletą klausimų. Anketa yra anoniminė, duomenys bus naudojami tik statistikai.

1. Jūsų amžius: \_\_\_\_\_

2. Lytis: \_\_\_\_\_

3. Klasė: \_\_\_\_\_

4. Mokyklos tipas: \_\_\_\_\_

5. Gyvenamoji vieta (rajono centras, miestelis, kaimas): \_\_\_\_\_

6. Kiek laiko mokotės programuoti? \_\_\_\_\_

7. Ar programos navigacija aiški ir patogi:

Taip;

Ne;

Pasiūlymai: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

8. Ar tenkina medžiagos pateikimas ir išsamumas?

Taip;

Ne;

Pasiūlymai: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

9. Ar gali tokia programa (pilnai užbaigta) būti naudinga mokantis programuoti (pradedandiesiems)

Taip;

Ne;

Trūkumai: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

10. Bendri pasiūlymai ir pastabos (trūkumai ir privalumai) \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Dėkoju už skirtą laiką.

## 7 PRIEDAS. PROCEDŪROS „TIKRINIMAS“ TEKSTAS

```
procedure TForm14.Tikrinimas;
var Klaidos, k, Ats: integer;
begin
    Klaidos:= 0;
    Ats:= 0;
    if StrToInt(Edit1.Text)<> Dalyba[1] then begin
        Klaidos:= Klaidos + 1;
        Edit1.Color:= clRed;
    end
    else Edit1.Color:= clMoneyGreen;
    if StrToInt(Edit7.Text)<> Dalyba[2] then begin
        Klaidos:= Klaidos + 1;
        Edit7.Color:= clRed;
    end
    else Edit7.Color:= clMoneyGreen;
    if StrToInt(Edit11.Text)<> Dalyba[3] then begin
        Klaidos:= Klaidos + 1;
        Edit11.Color:= clRed;
    end
    else Edit11.Color:= clMoneyGreen;
    if StrToInt(Edit15.Text)<> Dalyba[4] then begin
        Klaidos:= Klaidos + 1;
        Edit15.Color:= clRed;
    end
    else Edit15.Color:= clMoneyGreen;
    if StrToInt(Edit19.Text)<> Dalyba[5] then begin
        Klaidos:= Klaidos + 1;
        Edit19.Color:= clRed;
    end
    else Edit19.Color:= clMoneyGreen;
    if StrToInt(Edit23.Text)<> Dalyba[6] then begin
        Klaidos:= Klaidos + 1;
        Edit23.Color:= clRed;
    end
    else Edit23.Color:= clMoneyGreen;
    if StrToInt(Edit27.Text)<> Dalyba[7] then begin
        Klaidos:= Klaidos + 1;
        Edit27.Color:= clRed;
    end
    else Edit27.Color:= clMoneyGreen;
    if StrToInt(Edit3.Text)<> Liekana[1] then begin
        Klaidos:= Klaidos + 1;
        Edit3.Color:= clRed;
    end
    else Edit3.Color:= clMoneyGreen;
    if StrToInt(Edit8.Text)<> Liekana[2] then begin
        Klaidos:= Klaidos + 1;
        Edit8.Color:= clRed;
    end
    else Edit8.Color:= clMoneyGreen;
    if StrToInt(Edit12.Text)<> Liekana[3] then begin
```

```

        Klaidos:= Klaidos + 1;
        Edit12.Color:= clRed;
    end
    else Edit12.Color:= clMoneyGreen;
if StrToInt(Edit16.Text)<> Liekana[4] then begin
    Klaidos:= Klaidos + 1;
    Edit16.Color:= clRed;
    end
    else Edit16.Color:= clMoneyGreen;
if StrToInt(Edit20.Text)<> Liekana[5] then begin
    Klaidos:= Klaidos + 1;
    Edit20.Color:= clRed;
    end
    else Edit20.Color:= clMoneyGreen;
if StrToInt(Edit24.Text)<> Liekana[6] then begin
    Klaidos:= Klaidos + 1;
    Edit24.Color:= clRed;
    end
    else Edit24.Color:= clMoneyGreen;
if StrToInt(Edit28.Text)<> Liekana[7] then begin
    Klaidos:= Klaidos + 1;
    Edit28.Color:= clRed;
    end
    else Edit28.Color:= clMoneyGreen;

if StrToInt(Edit2.Text)<> Atimti[1] then begin
    Klaidos:= Klaidos + 1;
    Edit2.Color:= clRed;
    end
    else Edit2.Color:= clMoneyGreen;
if StrToInt(Edit4.Text)<> Atimti[2] then begin
    Klaidos:= Klaidos + 1;
    Edit4.Color:= clRed;
    end
    else Edit4.Color:= clMoneyGreen;
if StrToInt(Edit9.Text)<> Atimti[3] then begin
    Klaidos:= Klaidos + 1;
    Edit9.Color:= clRed;
    end
    else Edit9.Color:= clMoneyGreen;
if StrToInt(Edit13.Text)<> Atimti[4] then begin
    Klaidos:= Klaidos + 1;
    Edit13.Color:= clRed;
    end
    else Edit13.Color:= clMoneyGreen;
if StrToInt(Edit17.Text)<> Atimti[5] then begin
    Klaidos:= Klaidos + 1;
    Edit17.Color:= clRed;
    end
    else Edit17.Color:= clMoneyGreen;
if StrToInt(Edit21.Text)<> Atimti[6] then begin
    Klaidos:= Klaidos + 1;
    Edit21.Color:= clRed;
    end
    else Edit21.Color:= clMoneyGreen;
if StrToInt(Edit25.Text)<> Atimti[7] then begin

```

```

        Klaidos:= Klaidos + 1;
        Edit25.Color:= clRed;
    end
    else Edit25.Color:= clMoneyGreen;

if StrToInt(Edit5.Text)<> Skirtumas[1] then begin
    Klaidos:= Klaidos + 1;
    Edit5.Color:= clRed;
end
    else Edit5.Color:= clMoneyGreen;
if StrToInt(Edit6.Text)<> Skirtumas[2] then begin
    Klaidos:= Klaidos + 1;
    Edit6.Color:= clRed;
end
    else Edit6.Color:= clMoneyGreen;
if StrToInt(Edit10.Text)<> Skirtumas[3] then begin
    Klaidos:= Klaidos + 1;
    Edit10.Color:= clRed;
end
    else Edit10.Color:= clMoneyGreen;
if StrToInt(Edit14.Text)<> Skirtumas[4] then begin
    Klaidos:= Klaidos + 1;
    Edit14.Color:= clRed;
end
    else Edit14.Color:= clMoneyGreen;
if StrToInt(Edit18.Text)<> Skirtumas[5] then begin
    Klaidos:= Klaidos + 1;
    Edit18.Color:= clRed;
end
    else Edit18.Color:= clMoneyGreen;
if StrToInt(Edit22.Text)<> Skirtumas[6] then begin
    Klaidos:= Klaidos + 1;
    Edit22.Color:= clRed;
end
    else Edit22.Color:= clMoneyGreen;
if StrToInt(Edit26.Text)<> Skirtumas[7] then begin
    Klaidos:= Klaidos + 1;
    Edit26.Color:= clRed;
end
    else Edit26.Color:= clMoneyGreen;

for k:= i downto 1 do
    Ats:= Ats * 10 + Skirtumas[k];

if Edit29.Text <> 'Atsakymas' then
    if Ats <> StrToInt(Edit29.Text) then begin
        Klaidos:= Klaidos + 1;
        Edit29.Color:= clRed;
    end
    else Edit29.Color:= clMoneyGreen
else begin
    ShowMessage ('Parašykite atsakymą ');
    Edit29.Color:= clRed;
end;
ShowMessage ('Yra ' + IntToStr(Klaidos) + ' klaidų');

end;

```

## 8 PRIEDAS. LUKŠIŲ V. GRYBO GIMNAZIJOS IT MOKYTOJO ATSILIEPIMAS APIE MOKOMOSIOS APLINKOS REALIZUOTĄ DALĮ

Antanas Burkšaitis,  
informacinių technologijų mokytojas metodininkas,  
Lukšių Vinco Grybo gimnazija

### Recenzija

Malonu, kada kažkas mėgina arba pradeda kurti mokomąsias programas lietuvių kalba. Ši sritis, mano manymu, šiek tiek nepelnytai pamiršta. Mokiniai visada mėgsta naudotis šiomis programomis - net ir tada, kai jos neišbaigtos arba ne visai tobulos. Todėl pagiriamasis žodis programos kūrėjui būtinas.

Magistranto sukurta programa paprastai įdiegiama, nereikalauja didelių resursų, veikia visose paskutinėse Windows operacinėse sistemose, gali būti paleidžiama tinkle.

Programa nesunkiai valdoma ir neperkrauta nereikalinga informacija. Mano manymu, ja galima būtų ateityje „pažaišti“ įvairiomis kryptimis. Dabar ji labiau orientuota pagrindinei mokyklai. Galima būtų ją praplėsti iki pasiruošimo valstybiniam informacinių technologijų egzaminui lygmens. Be to, nemanau, kad būtų labai sudėtinga „prikabinti“ C++ teoriją ir testus. Nuo to programa taptų universalesnė. Dažnai, kad greičiau vyktų darbas bei programuotojui būtų paprasčiau, galima mokinių priversti pasiimti rašymo priemonę (kartais jiems pravartu tai padaryti, nes visai užmirš rašyti). Kai kuriuose testuose įdėta daug gana painaus darbo (malonu matyti, kad įsisavinta programavimo technika), bet jeigu būtų nueita paprastesniu keliu, programa, mano manymu, jau turėtų žymiai daugiau teorijos ir testų.

Mūsų mokyklos dešimtos klasės moksleiviai pamėgino dirbti su šia programėle. Panašios taikomosios programos visada sukelia daug emocijų. Atsiranda žaidybinis azartas net ir tada, kai ne viskas pavyksta. Moksleiviai žūtbūt stengiasi susigrumti su kompiuteriu. Nebuvo išimtis ir ši programa. Tad magistrantui pridėjus dar šiek tiek darbo, išeitų pakankamai geras programinis produktas. Mokinių atsiliepimas – duokit daugiau tokių programų.



Rašyti pastabas ir kritikuoti tikrai nesunku, todėl, ko gero, šiandien rašančių programos yra daug mažiau negu jas kritiškai vertinančių. Jaučiu didelę pagarbą kuriantiems ir linkiu jiems sėkmės.

Lukšiai,

2011-05-25