



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

Rima Čeplikienė

**Programavimo Paskal kalba mokomosios aplinkos
sudarymas ir tyrimas**

Magistro darbas

**Recenzentas
lekt. dr. I.Lagzdinytė
2008-05-19**

**Vadovas
doc. K.Baniulis
2008-05-19**

**Atliko
IFT 6 gr. stud.
R.Čeplikienė
2008-05-19**

KAUNAS, 2008

SUMMARY

Living in a fast moving world skills of using informative technologies are of great importance. Students' general skills to use informative technologies are more important than specific knowledge or being able to use computer programs.

Educational aid „Pascal programming language“ is a wonderful alternative not only for students choosing a programming module but for junior pupils who are interested in programming as well. The aim and content of created educational aid are introduced in this paper. Programming language Pascal governing constructions: attribution to sentence, conditional sentence and cycles are presented too.

An educational aid consists of theory, tasks and tests of 2 types: self control tests and tests.

Hot Potatoes programme of creating interactive tasks, HTML language and distance testing system TesTTool are used for making tests. Teaching material is presented in HTML formate, animated examples are made using Wink 2000 programme.

This teaching aid can be used:

1. Introducing a new topic, revising or answering a lesson
2. For students' individual, independent learning while extending programming knowledge.

Analyses of the use of programme in training process and results of experiment are presented in this paper.

Turinys

IVADAS	5
1 DARBO TIKSLAS IR UŽDAVINIAI	6
2 PROBLEMOS ANALIZĖ	7
2.1 MOKYTOJŲ NUOMONĖS TYRIMAS	8
2.2 MOKSLEIVIŲ NUOMONĖS TYRIMAS	11
3 MOKOMŲJŲ PROGRAMŲ IR METODŲ ANALIZĖ	15
3.1 MOKOMŲJŲ PROGRAMŲ VERTINIMO KRITERIJŲ SUDARYMAS	15
3.2 MOKOMŲJŲ PROGRAMŲ KLASIFIKACIJA.....	16
3.3 MOKYMO METODAI IR JŲ KLASIFIKACIJA	18
3.4 PROGRAMAVIMO DALYKO MOKOMŲJŲ VEIKSMŲ APIBRĖŽIMAS PAGAL BLOOMO KRITERIJUS	19
3.5 MOKOMŲJŲ PROGRAMŲ APŽVALGA.....	21
3.5.1 PROGRAMAVIMO PRADMENYS.....	22
3.5.2 <i>Automatizuotas apmokymo kursas Turbo Pascal 6.0</i>	23
3.5.3 <i>INFORMATIKA</i>	24
3.5.4 <i>PaskMok</i>	25
3.5.5 <i>ELEKTRONINIS INFORMATIKOS VADOVĒLIS</i>	26
3.5.6 „Komenskio Logo“.....	26
3.6 PROGRAMŲ PALYGINIMAS.....	27
4 ELEKTRONINĖS PRIEMONĖS PROJEKTAVIMAS	28
4.1 MOKOMOSIOS PROGRAMOS SUDARYMO MODELIS.....	29
4.2 MOKYMO PROCESO ORGANIZAVIMAS	30
4.3 PROGRAMOS PASKIRTIS.....	30
4.4 PROGRAMOS STRUKTŪRA.....	31
4.5 MOKYMO MEDŽIAGOS PAGRINDINĖS TEMOS.....	32
4.6 MOKOMOSIOS PROGRAMOS REIKALAVIMŲ SPECIFIKACIJA	32
4.6.1 <i>Vartotojų reikalavimai</i>	32
4.6.2 <i>Nefunkciniai reikalavimai</i>	33
4.6.3 <i>Funkciniai reikalavimai</i>	33
4.7 PROGRAMOS PANAUDOJIMO ATVEJAI.....	34
4.8 PROGRAMOS STRUKTŪRA.....	39
4.9 TEMOS „CIKLAI“ ONTOLOGIJA.....	42
4.10 CIKLO KONSTRAVIMO BŪSENŲ GRAFAS.....	43
5 MOKYMO PRIEMONĖS ĮGYVENDINIMO TECHNOLOGINIAI SPRENDIMAI	45
5.1 ŽINIATINKLIO KŪRIMO PROGRAMA MICROSOFT OFFICE FRONTPAGE	45
5.2 TESTŲ IR UŽDUOČIŲ KŪRIMO PRIEMONĖS	46
5.2.1 <i>Interaktyvių užduočių kūrimo programa „Hot Potatoes“</i>	47
5.2.2 <i>HTML kalba ir JavaScript</i>	48
5.2.3 <i>TestTool</i>	50
5.3 DEMONSTRACINIŲ OBJEKTŲ KŪRIMAS	52
6 MOKOMOSIOS PRIEMONĖS MOKYMO OBJEKTŲ ĮGYVENDINIMAS IR TESTAVIMAS	52
6.1 TECHNOLOGINIS MOKYMO OBJEKTŲ ĮGYVENDINIMAS.....	52
6.2 MOKOMOSIOS PRIEMONĖS TESTAVIMAS.....	53
6.2.1 <i>Eksperimentinio tyrimo tikslas</i>	53
6.2.2 <i>Tyrimo uždaviniai</i>	53
6.2.3 <i>Tyrimo eiga</i>	54
6.2.4 <i>Tyrimo rezultatai</i>	55
6.3 GALIMI PRIEMONĖS PATOBULINIMAI.....	56
7 IŠVADOS	57
8 LITERATŪROS SĄRAŠAS	58
9 PRIEDAI	59

PAVEIKSLĖLIŲ SĄRAŠAS

2.1 PAV. DIAGRAMA NR.1	9
2.2 PAV. DIAGRAMA NR.2	9
2.3 PAV. DIAGRAMA NR. 3	10
2.4 PAV. DIAGRAMA NR.4	10
2.5 PAV. DIAGRAMA NR.5	11
2.6 PAV. DIAGRAMA NR.6	11
2.7 PAV. DIAGRAMA NR. 7	12
2.8 PAV. DIAGRAMA NR. 8	13
2.9 PAV. DIAGRAMA NR.9	14
2.10 PAV. DIAGRAMA NR.10	14
3.1 PAV. MOKOMŪJŲ PROGRAMŲ SKIRSTYMAS PAGAL TURINĮ	16
3.2 PAV. MOKOMŪJŲ PROGRAMŲ SKIRSTYMAS	17
3.3 PAV. PROGRAMOS „PROGRAMAVIMO PRADMENYS“ PAGRINDINIS LANGAS	22
3.4 PAV. PROGRAMOS „PROGRAMAVIMO PRADMENYS“ INTERAKTYVIOS UŽDUOTIES LANGAS ...	22
3.5 PAV. PROGRAMOS „PROGRAMAVIMO PRADMENYS“ TESTO LANGAS	23
3.6 PAV. AUTOMATIZUOTO APMOKYMO KURSO TURBO PASCAL 6.0 TURINYS	23
3.7 PAV. PROGRAMOS „INFORMATIKA“ TEMŲ IR UŽDUOTIES LANGAI	24
3.8 PAV. PROGRAMOS „INFORMATIKA“ TESTO LANGAS	24
3.9 PAV. PROGRAMOS PASMOK TEMŲ LANGAS	25
3.10 PAV. ELEKTRONINIO INFORMATIKOS VADOVĖLIO INTERAKTYVIOS UŽDUOTIES LANGAS ...	26
4.1 PAV. PROGRAMOS VARTOTOJAI IR PANAUDOJIMO ATVEJAI	35
4.2 PAV. PROGRAMOS STRUKTŪRA	40
4.3 PAV. PROGRAMOS LANGO TARPUSAVIO RYŠIAI	41
4.4 PAV. PROGRAMOS TEMŲ LOGINIS IŠDĖSTYMAS	41
4.5 PAV. PROGRAMOS TEMOS „CIKLAI“ LANGAS	42
4.6 PAV. CIKLO ANTOLOGIJA	43
5.1 PAV. TEMOS „PRISKYRIMO SAKINYS“ PAGRINDINIS LANGAS	46
5.2 PAV. TESTO, SUKURTO SU HOT POTATOES, PAVYZDYS	47
5.3 PAV. TESTO PAVYZDYS	48
5.4 PAV. JAWASCRIPT PANAUDOJIMAS MOKINIO REGISTRACIJAI	49
5.5 PAV. TESTO TEISINGUMUI NUSTATYTI FUNKCIJOS PANAUDOJIMAS HTML KALBOJE	49
5.6 PAV. TESTO SUKURTO SU TESTTOOL PAVYZDYS NR.1	51
5.7 PAV. TESTO SUKURTO SU TESTTOOL PAVYZDYS NR.2	51
5.8 PAV. PROGRAMOS ANIMUOTŲ UŽDUOČIŲ PAVYZDŽIAI	52

LENTELIŲ SĄRAŠAS

3.1 LENTELĖ PAŽINIMO SRITYS	19
3.2 LENTELĖ MOKOMŪJŲ PROGRAMŲ PALYGINIMAS	28
4.1 LENTELĖ VEIKLOS „PROGRAMOS ĮDIEGIMAS“ VARTOTOJŲ VEIKSMAI IR VEIKLA	36
4.2 LENTELĖ VEIKLOS „KURSO KŪRIMAS“ VARTOTOJŲ VEIKSMAI IR VEIKLA	36
4.3 LENTELĖ VEIKLOS „TEMOS PASIRINKIMAS“ VARTOTOJŲ VEIKSMAI IR VEIKLA	37
4.4 LENTELĖ VEIKLOS „PAMOKOS PASIRINKIMAS“ VARTOTOJŲ VEIKSMAI IR VEIKLA	37
4.5 LENTELĖ VEIKLOS „TEORIJOS SKAITYMAS“ VARTOTOJŲ VEIKSMAI IR VEIKLA	38
4.6 LENTELĖ VEIKLOS „VAIZDINĖS MEDŽIAGOS PERŽIŪRA“ VARTOTOJŲ VEIKSMAI IR VEIKLA ...	38
4.7 LENTELĖ VEIKLOS „TESTŲ ATLIKIMAS“ VARTOTOJŲ VEIKSMAI IR VEIKLA	39
4.8 LENTELĖ CIKLO KONSTRAVIMO EIGA IR TAISYKLĖS	44
6.1 LENTELĖ MOKYMO OBJEKTŲ TECHNOLOGINIS ĮGYVENDINIMAS	53
6.2 LENTELĖ TYRIME DALYVAVUSIŲ TESTŲ REZULTATŲ LENTELĖ	54
6.3 LENTELĖ TYRIME NEDALYVAVUSIŲ TESTŲ REZULTATŲ LENTELĖ	55
6.4 LENTELĖ ATSILIEPIMŲ APIE PROGRAMĄ LENTELĖ	55

Įvadas

Šiandieninį pasaulį sunku įsivaizduoti be kompiuterio ir be naujausių informacinių technologijų laimėjimų. Informacinės technologijos turi didžiulę įtaką šiuolaikiniam gyvenimui, o ypač švietimo sistemai. Mūsų mokiniai moka dirbti kompiuteriais, dirba su interaktyviais, mąstymą provokuojančiais žaidimais, naršo po internetą, susiranda reikalingos informacijos, integruoja įvairias informacijos rūšis, sintezuoja tai į kažką naujo ir įdomaus. Bendrauja vieni su kitais elektroninio pašto programa, dalyvauja įvairiuose konferencijose, pokalbiuose ir t.t. Tai visiškai keičia mokymo turinį, mokytojas ir vadovėlis jau nebėra vieninteliai žinių šaltiniai. Dabar, kai mus pasiekia didelis informacijos srautas, svarbiausiu mokymo tikslu tampa mokyti moksleivį mokyti, sudominti jį naujausiomis technologijomis, kurios padėtų jam atrasti žinomus reiškinius ir dėsningumus. Mokytojas turi tapti partneriu ar pagalbininku naujame mokymosi procese.

Informacijos ir komunikacijos technologijos integravimas į mokymą – švietimo reformos uždavinys. Mokytojo informacinė kompetencija – tai ne vien sugebėjimas dirbti su kompiuteriu ir programine įranga, reikia suvokti ir technologijos socialinę paskirtį. Mokytojo pasirengimas gyventi informacijos visuomenėje, jo informacinė kultūra – svarbus veiksnys diegiant technologijas švietime.

Mokytojai turi siekti, kad mokiniai „išmąnytų informatikos, informacinių ir komunikacinių technologijų žinių sistemą bei gebėjimą šias žinias taikyti pažinimo, kūrybos bei mokymosi procese, lavinti struktūrinį ir algoritminį mąstymą, skatinti kūrybinio improvizavimo gebėjimus tiek dirbant individualiai, tiek kolektyviai, ugdyti pasitikėjimą savo jėgomis; ugdyti smalsumą naujovėms, polinkį mokytis šiuolaikinių, efektyvesnių veiklos būdų, domėtis moderniomis technologijomis“ [1].

Bendrojo lavinimo mokyklos bendroji programa numato programavimo pradmenų kursą, kuris įrašytas į išplėstinio kurso modulių sąrašą (profilinis mokymas, 11-12 klasės). Išplėstinis kursas orientuotas į taikymą, mokėjimų bei įgūdžių ugdymą. Nuo bendrojo kurso jis skiriasi tuo, kad yra labiau integruotas su kitais profilio dalykais.

Programavimas — nelengvas intelektualinis darbas. Visų pirma reikia išnagrinėti kompiuterį ir programavimo kalbas, tam, kad turėti supratimą apie tikrąją informatiką.. Tačiau kur kas svarbiau suprasti bendrus programavimo principus, suvokti pagrindines programavimo kalbų konstrukcijas. Tai išmokti nėra lengva, tenka patiems daug pasipraktikuoti kuriant

algoritmus ir juos išbandant kompiuteriu. Geriausia mokytis tada, kai yra į ką panašaus pasižiūrėti

Programavimas — tai vienas iš sunkiausių informatikos temų ne tik mokykliniame informacinių technologijų kurse, bet ir studijuojant. Tai kūrybiška, daug intelekto pastangų, didelio loginio mąstymo reikalaujanti veikla.

Pasirinkto magistrinio darbo tema yra „Programavimo Pascal programavimo kalbos valdymo konstrukcijos“. Šia tema ryžausi pasirinkti dėl keleto priežasčių:

- Trūksta interaktyvių programų, kuriose mokinai vaizdžiai matytų programavimo konstrukcijų veikimą;
- Trūkumas programų, kuriose būtų laisvė rinktis temas ir kuriose gautų glaustą informaciją;
- Būtų pateikiamos ne tik formalios žinios, bet jos susietos ir su praktika, kad mokiniai galėtų teorines žinias patikrinti per savikontrolės pratimus.;

Darbe bus kuriama vizualiai patraukli, nesudėtinga naudoti mokymo priemonė „Pascal programavimo kalbos valdymo konstrukcijos“, kuri padėtų mokytis programavimo pradmenų. Bus remiamasi atliktu tyrimu apie moksleivių ir mokytojų požiūrį į esamų mokomųjų kompiuterinių priemonių panaudojimą mokantis programavimo, jų galimybes ir trūkumus.

1 Darbo tikslas ir uždaviniai

Darbo tikslas – sukurti kompiuterinę mokomąją priemonę, kuri padėtų moksleiviams kuo geriau įsisavinti Pascal valdymo konstrukcijas ir programavimą, padėtų ugdyti algoritminį mąstymą.

Pagrindiniai uždaviniai tikslui pasiekti:

1. Problemos apibrėžimas bei vartotojų poreikių analizė.
2. Mokomųjų programų vertinimo kriterijų nustatymas
3. Mokymo metodų ir mokomųjų priemonių klasifikavimas, programavimo dalyko mokomųjų veiksmų apibrėžimas pagal Bloomo kriterijus.
4. Mokyklose taikomų mokomųjų programų apžvalga ir jų įvertinimas pagal pasirinktus kriterijus
5. Mokomosios priemonės projektavimas ir jos įgyvendinimas taikant įvairius technologinius sprendimus
6. Sukurtos priemonės eksperimentinis tyrimas

2 Problemos analizė

Lietuvoje programavimo mokymas turi senas tradicijas, jo mokymo metodika išsamiai ištyrinėta ir moksliai pagrįsta. Bendrojo lavinimo mokykloje programavimo buvo mokoma nuo pat informatikos kurso pradžios. Tada algoritmavimui buvo skirta daugiau nei pusė informatikos pamokų. Vėliau programavimui imta skirti šiek tiek mažiau dėmesio.

Algotmavimo temų (pvz., algoritmo samprata, algoritmų vaizdavimo ir užrašymo būdai, duomenų rūšys, pagrindinės valdymo konstrukcijos, algoritminiai uždavinių sprendimo principai) yra ir dabartinėje bendrojo lavinimo informacinių technologijų kurso programoje. Algotmavimas sudaro maždaug ketvirtį privalomojo kurso. Moksleiviai mokosi tik pagrindinių programų sudarymo principų, beveik visiškai neakcentuojamas programavimo kalbos sintaksės mokymasis. Mokykloje paplitus kompiuteriams bei tobulesnei programinei įrangai, su kuria dirbti nereikia programavimo žinių ir profesionalaus informatiko išsilavinimo, pasikeitė ir bendrojo lavinimo mokyklos informatikos tikslai: siekiama supažindinti moksleivius su įvairiais informacinių technologijų taikymo aspektais, išugdyti reikiamus kompiuteriu naudojimosi įgūdžius. Pakito ir turinys: atsisakyta teorinių kai kurių programavimo temų, daugiau akcentuojamas kompiuterijos mokymas. Programavimas, kaip atskiras pasirenkamasis modulis, yra tik baigiamosiose klasėse.

Programavimą mokyti reiktų dėl daugelio svarbių priežasčių

- Programavimo mokoma ne dėl to turėtų būti, kad jo yra svarbus turinys, bet ir todėl kad jis lavina aukštesnio lygio mąstymo gebėjimus. Daug gyvenime pasitaikančių užduočių galima būtų išspręsti algoritmiškai. Sudarydamas algoritmus moksleivis suvokia, kad yra daug būdų tam pačiam uždaviniui spręsti: tai skatina ieškoti pačio optimaliausio ir tiesiausio sprendimo kelio. Yra ugdomas gebėjimas analizuoti, loginis mąstymas, kūrybiškumas, savarankiškumas.
- Programavimas reikalauja tam tikrų gebėjimų, tai alternatyva matematikos mokslui. Mokant uždavinių sprendimo mokymo elementai turėtų būti aktyvus tyrinėjimas, motyvacija, praktinės mokymosi aplinkybės.
- Programavimas sudaro galimybes panaudoti daugiau kompiuterio resursų, pvz., naudoti makrokomandas įvairiose taikomuosiose programose.
- Programavimas padeda geriau suvokti kompiuterio logiką, jo ribotumus ir galimybes. Sudaro sąlygas plačiau taikyti informacines technologijas.

Beveik kiekviename mokykloje yra mokinių besidominčių programavimu, dalis iš jų lanko programavimo pamokas, kiti mokosi kompiuterininkų mokykloje arba savarankiškai. Lietuvos mokyklose yra mokoma programuoti Pascal kalba, tačiau jaunimo tarpe yra didelis

susidomėjimas ir naujomis programavimo kalbomis. Tai mokytojui tik padeda, bet trūksta metodinės literatūros, moksleiviams skirtų kompiuterinių mokomųjų programų. Interneto portale www.e-mokykla.lt, kuri kuruoja Švietimo informacinių technologijų centras, yra sukaupta nemažai mokomųjų programų įvairių dalykų mokymui, tačiau jų ypač trūksta programavimui mokytis. Yra keletas, bet jos jau yra pasenę tiek savo turiniu, tiek savo galimybėmis ir forma, nes dauguma jų pritaikytos MS-DOS operacinei sistemai. Visos jos neatitinka dabartinės informatikos mokymo programos.

Neįmanoma išmokyti programuoti tik skaitant teoriją, kas būdinga kitiems mokslams. Šitaip galima tik turėti supratimą arba susipažinti su programavimo elementais. Kad išmoktume programuoti, būtina patiems bandyti kurti programas. Programavimas tai itin kūrybinga veikla, reikalaujanti labai daug pastangų ir žinių. Tam pačiam uždaviniui galima sudaryti ne vieną algoritmą, o pačiai optimaliausiai programai sudaryti reikia ir loginio mąstymo, ir žinių. Todėl reikia daug praktikos, noro mokintis, mąstyti. Gal dėl to programavimo kursas vidurinėje mokykloje yra labai sunkiai suvokiamas. Mokomosios kompiuterinės programos programavimui mokytis, reikalingos dėl daugelio priežasčių:

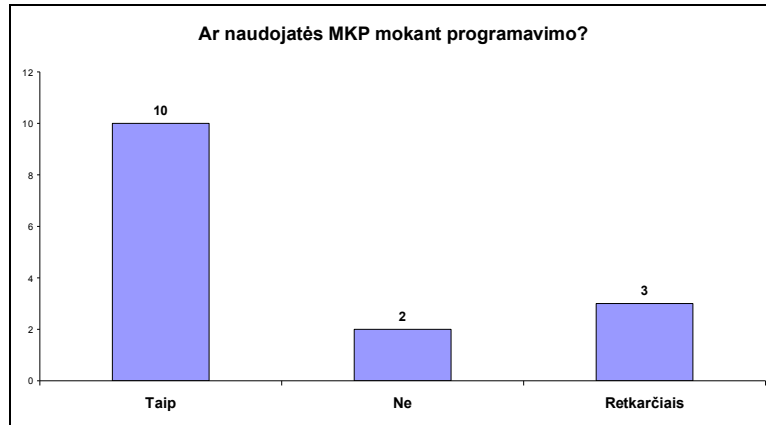
- Kad mokiniai turėtų galimybę mokytis individualiai ir jiems norimu tempu;
- Kad žinias būtų galima perteikti naujausių technologijų dėka;
- Kad skatintų mokinių kūrybingą veiklą, formuotų algoritminį mąstymą;
- Kad būtų pateikiamos ne tik žinios, bet jos būtų susietos su praktika, su interaktyviais pavyzdžiais;
- Būtų galimybė patikrinti žinias.

2.1 Mokytojų nuomonės tyrimas

Prieš pradėdama planuoti savo veiklą, norėjau sužinoti mokytojų ir moksleivių nuomonę. Apklauskos tikslas buvo sužinoti kaip integruojamos programavimui skirtos ir ar naudojamos mokomosios kompiuterinės priemonės. Norėjau išsiaiškinti ar mokytojai ir moksleiviai jaučia poreikį naudotis šiomis programomis ir su kokiomis problemomis susiduria. Nes viena svarbiausių sąlygų sėkmingai mokomosios priemonės integracijai į ugdymo procesą – teigiama mokytojo nuomonė apie jam siūlomą naujovę. Apklausoje dalyvavo 10 informacinių technologijų mokytojų iš įvairių Varėnos rajono vidurinių mokyklų. Jiems buvo pateikta anketa (žr. 60 psl. 2 priedas).

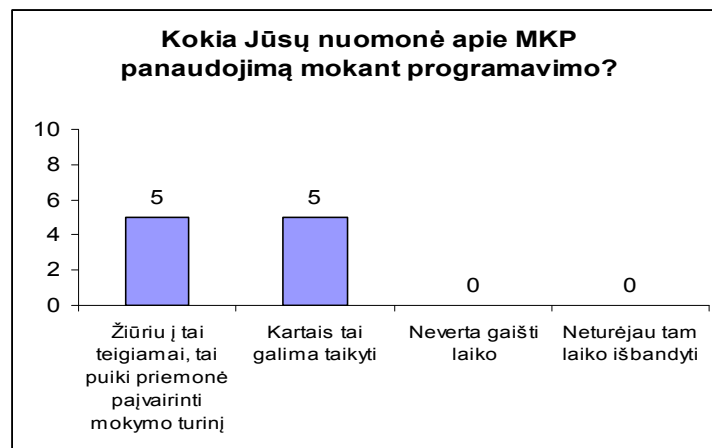
Iš gautų rezultatų matosi (žr. 2.1 pav.), kad dauguma mokytojų naudoja mokomasias kompiuterines priemones savo pamokose, tačiau ne visi. Ta nulemia įvairios priežastys: mažai valandų skirta programavimui mokytis, maža mokinių motyvacija, trūksta šiuolaikiškų MKP,

pačios mokomosios priemonės jau yra pasenę, neatitinka šių dienų techninių reikalavimų ar mokymo programų.



2.1 pav. Diagrama Nr.1

Tačiau dauguma mokytojų pasisako už tai (žr. 2.2 pav.), kad tai reikalingos priemonės, tam kad sudominti mokinius, pajvairinti mokymosi procesą. Galima pastebėti tai, kad visi mokytojai savo darbe naudoja ar bent pabandė naudoti MKP.



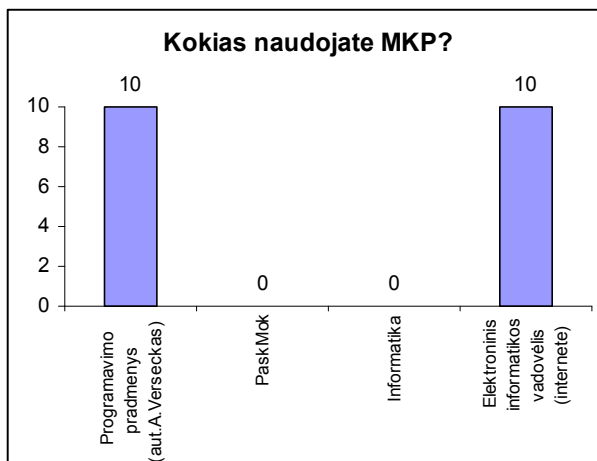
2.2 pav. Diagrama Nr.2

Tačiau kaip mokytojai žiūri į esamą programinę įrangą? Sprendžiant pagal jų pateiktus atsakymus yra naudojamos dažniau tik 2 priemonės (žr. 2.3 pav.) :

- Programavimo pradmenys (autorius V.Verseckas)
- Elektroninis programavimo vadovėlis

(http://62.80.232.136/informatika/kurybiniai_darbai/vad9_10/algos/algo_00.html).

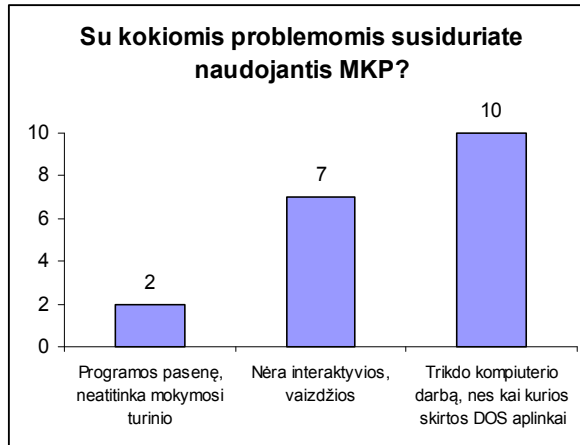
Kitomis priemonėmis nesinaudojama dėl jų nepritaikomumo darbai pamokoje.



2.3 pav. Diagrama Nr. 3

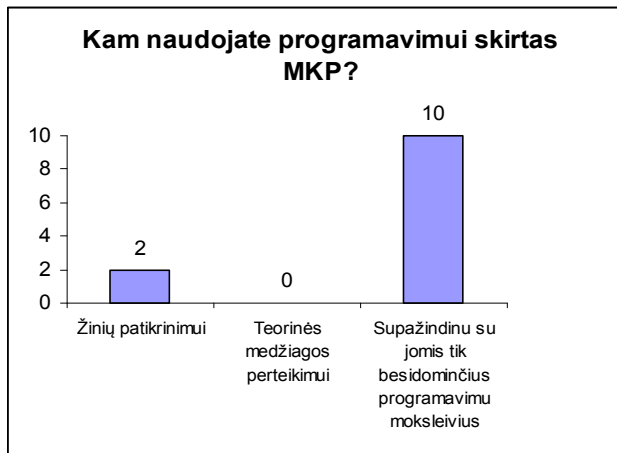
Dauguma mokytojų yra už tai, kad tokios priemonės yra reikalingos ir būtų naudingos. Jas ne tik būtų galima pasiūlyti savarankiškam mokymuisi, bet ir būtų pritaikytos darbui pamokoje, t.y. turėtų aiškiai aprašytą teorinę dalį, būtų įrankis mokinių žinių patikrinimui testų ir savarankiškų užduočių pagalba.

Mokytojų atsiliepimai apie šiuo metu esamas ir naudojamas kompiuterines priemones beveik sutampa (žr. 2.4 pav.). Visi pritaria tam, kad jos yra pasenę, neatitinka šių dienų techninių reikalavimų, nėra suderinamos su šiais dienai mokyklose naudojama programine įranga. Joms trūksta interaktyvumo, vaizdumo, patrauklumo.



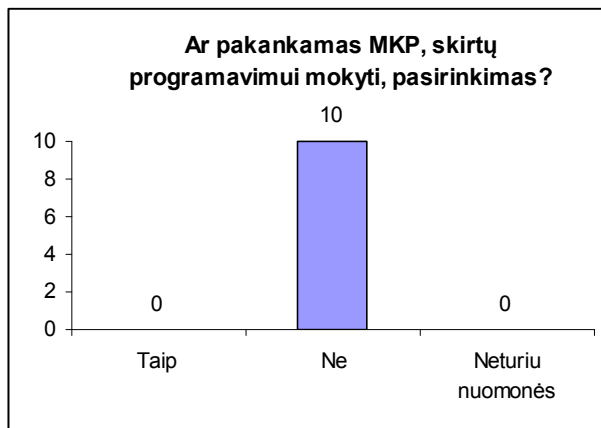
2.4 pav. Diagrama Nr.4

Iš apklausos rezultatų matosi, kad mokytojai su esamomis programomis tik supažindina besidominčius programavimu mokinius, jų beveik praktiškai nenaudodami pamokose. Mokytojai daugiau linkę patys kurti užduotis moksleivių žinių patikrinimui. (žr. 2.5 pav.)



2.5 pav. Diagrama Nr.5

Visi vienareikšmiškai pritaria tam, kad tokių kompiuterinių priemonių trūksta šiandieninėje mokykloje ir jų pasiūla yra labai maža. (žr. 2.6 pav.)



2.6 pav. Diagrama Nr.6

Galima daryti prielaidą, jog mokytojams aktualu pateikti savo ugdytiniais kuo naujesnės ir įdomesnės bei įvairesnės mokomosios medžiagos.

Mokytojų buvo klausama ar jie naudoja testus vertindami mokinių turimas žinias. Apie pusė mokytojų yra taikę kompiuterines testavimo priemones ar programas vertindami mokinių pasiekimus. Peržvelgus anketos rezultatus ir mokytojų nuomonę, galima daryti išvadą, kad mokytojai pageidauja daugiau ir įvairesnių testų dėstant programavimo kursą. Mokytojai vis plačiau taiko kompiuterinius testus ir mokymui ir mokinių mokymosi rezultatų vertinimui.

2.2 Moksleivių nuomonės tyrimas

Varėnos Ryto vidurinėje mokykloje atlikta žodinė apklausa tarp 11 klasės moksleivių, kurie 10 klasėje išklausė programavimo kursą. Apklausa parodė, kad programavimas jiems buvo nelabai suprantamas ir mėgiamas. Moksleiviams buvo sudėtinga perprasti tam tikras programavimo temas, pavyzdžiui temą „Ciklas“. Sunkiai buvo suvokiamos pačios Pascal

konstrukcijos, pats mechanizmas ką kompiuteris „galvoja“ vykdydamas jų sukurtas programėles. Gal dėl šių priežasčių ypač mažai mokinių baigiamosiose klasėse pasirenka programavimo modulį ir valstybinį informacinių technologijų egzaminą.

Klasėje tik vienam kitam moksleiviui programuoti buvo įdomu ir tik maža dalis žada pasirinkti programavimo modulį 12-toje klasėje. Pagrindinis argumentas tokiu mokinių buvo tai, kad programavimas jiems yra naudingas tuo, kad jie žada savo ateitį susieti su informacinėmis technologijomis arba savarankiškai domisi kitomis programavimo kalbomis ir terpėmis, kaip C++, Delfi, PHP ir kitomis.

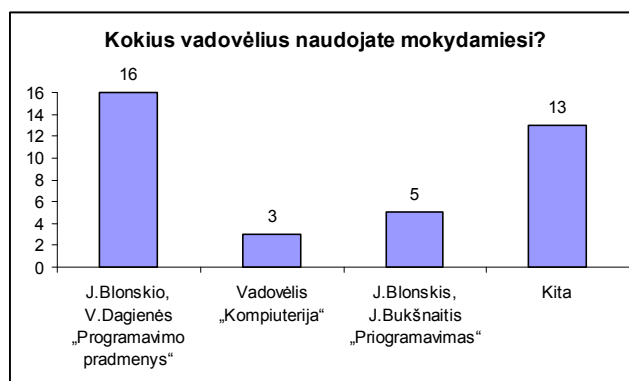
Nežiūrint į tai, programavimas yra svarbus, nes jo mokymasis pagrindinėje mokykloje leidžia mokiniams apsispręsti dėl tolimesnio informatikos mokymosi profiliuotose klasėse, o vėliau ir dėl su informatika ir informacinėmis technologijomis susijusių studijų.

Kitą apklausą atlikau su moksleiviais kurie jau yra pasirinkę programavimo pradmenų modulį. Apklausoje dalyvavo 17 mokinių iš Varėnos Ryto vidurinės mokyklos, Varėnos Ažuolo vidurinės mokyklos ir Merkinės V.Krėvės-Mickevičiaus gimnazijos. Jiems buvo pateikta anketa (žr. 59 psl. 1 Priedas).

Apibendrinus anketos rezultatus mokiniai atsakė, kad jiems programavimas bus naudingas, nesvarbu ar jie pasirenks laikyti valstybinį informacinių technologijų egzaminą ar ne, nes dauguma žada studijuoti aukštojoje mokykloje disciplinas artimas informatikai. Visi mokiniai supranta apie technologijų svarbą šiandieniniame gyvenime ir visi daugiau ar mažiau jau yra susidūrę su informacijos apdorojimu.

Apie pusę moksleivių dar domisi ir kitomis programavimo kalbomis ir terpėmis: C++, žiniatinklio kūrimo kalbą PHP ir Jawa (domisi 1 mokiny). Vienas mokiny mokosi neakivaizdinėje Jaunųjų programuotojų mokykloje.

Pateikus klausimą kokia mokomąją medžiaga moksleiviai naudojami, dauguma atsakiusiųjų (žr. 2.7 pav.) naudojami tik J.Blonskio ir V.Dagienės „Programavimo pradmenys“ vadovėliu, visa kitą medžiagą susijusią su programavimo mokymuisi atranda internete.

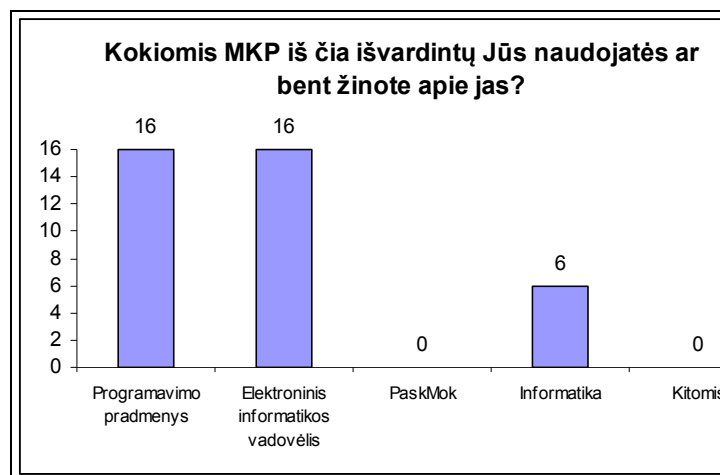


2.7 pav. Diagrama Nr. 7

Visi mokiniai žinojo tik 2 kompiuterines mokomąsias priemones (žr. 2.8 pav.):

- A.Versecko sukurtą „Programavimo pradmenys“
- Elektroninį informatikos vadovėlį patalpintą internete.

Šios priemonės buvo žinomos dėl to, kad dauguma moksleivių su jomis buvo supažindinti 10 klasėje mokydami programavimo įvadą. Tai natūralu, nes lyginant su mokytojų atsakymais, šios priemonės buvo pasiūlytos gabesniems ir besidomintiems programavimu moksleiviams.



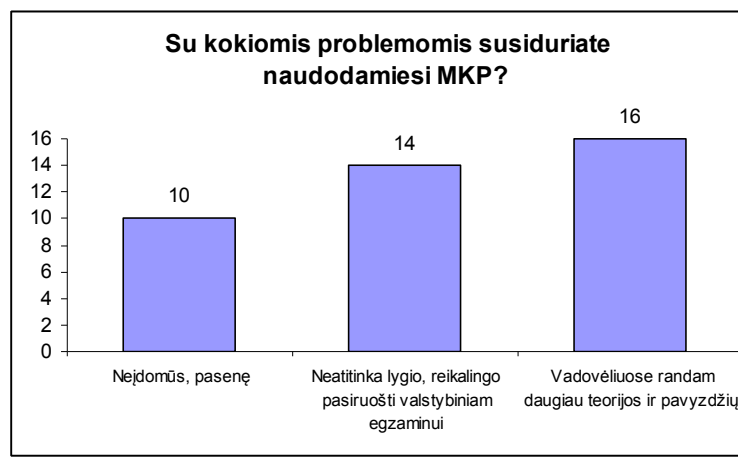
2.8 pav. Diagrama Nr. 8

Į klausimą, su kokiomis problemomis susiduriate jei naudojate MKP (žr. 2.9 pav.) moksleivių atsakymai praktiškai sutapo su mokytojų:

- pasenusios,
- trikdomas kompiuterio darbas,
- daugiau yra skirtos tik pažinčiai su programavimu, nes neatitinka valstybinio informacinių technologijų egzaminų programoje keliamų reikalavimų.

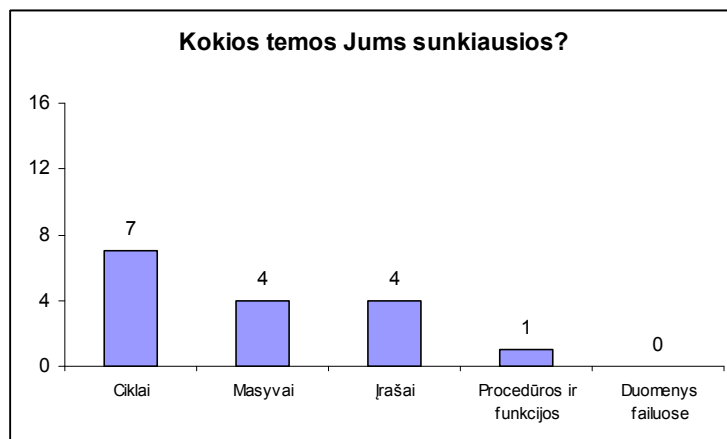
Dėl šių priežasčių mokymuisi ir žinių įgijimui yra daugiau naudojamas vadovėlis.

Mokiniai pageidauja tokios mokymosi priemonės, kurioje būtų aiškiai pateikta teorinė medžiaga, būtų priemonė žinių patikrinimui. Aišku, priemonė turi būti nenuobodi, patraukli ir lengvai valdoma. Ji turėtų pajvairinti ir informacinių technologijų pamokas, kuriose dažnokai trūksta naujų mokymo metodų taikymo.



2.9 pav. Diagrama Nr.9

Į klausimą kokios temos yra sunkiausios, atsakymai buvo gan įvairūs (žr. 2.10 pav.). Daugumai problemų sukelia ciklų pritaikymas programose, kitiems masyvai, įrašai.



2.10 pav. Diagrama Nr.10

Studentų požiūris į programavimą:

„Buvo apklausta 70 VPU – IV kurso informatikos bakalaurų, I ir II kurso informatikos magistrantų. 60 proc. studentų, į klausimą kokie dalykai iš informatikos jiems buvo naudingiausi, 50 proc. studentų atsakė, kad programavimo dalykai. Taigi studentams programavimas yra labai svarbus. Dauguma atsakiusių (91 proc.) jau dirba, taigi jų požiūris yra susijęs su darbine veikla. Įdomu ir tai, kad dauguma studentų (70 proc.) laiko Paskalio kalbos mokymą būtinu, nes jis dėstomas vidurinėje mokykloje. Kai kurie išreiškė pageidavimą, kad po Paskalio būtų dėstoma Delphi. Klausiami, kodėl daugumos tokios prastos algoritmovimo žinios, apie 80 proc. magistrantų atsakė, kad per mažai buvo skirta dėmesio algoritmovimo svarbai.“ [5].

Apibendrinant tyrimus galima teigti, kad pagrindiniai moksleivių ir mokytojų poreikiai yra šie:

- Valdymo paprastumas
- Dizaino patrauklumas
- Teorijos ir praktikos santykis

- Žinių pasitikrinimas
- Informacijos aktualumas
- Informacijos atnaujinimas
- Techninis suderinamumas su naudojama programine įranga.

3 Mokomųjų programų ir metodų analizė

3.1 Mokomųjų programų vertinimo kriterijų sudarymas

Mokinių ir mokytojų apklausos metu gauta informacija padėjo sudaryti kriterijų sąrašą, kurie padės įvertinti ir palyginti esamas programas programavimui mokyti. Moksleivių ir mokytojų paminėti kriterijai yra svarbūs, tačiau jų sąrašą papildysiu ir kitomis priemonių kokybę apibūdinančiomis savybėmis, kurios atitinka ne tik techninius reikalavimus, bet ir programos struktūrą, programos pritaikomumą darbui pamokoje ir savarankiškam darbui, informacijos pateikimą.

Savybių sąrašas pagal kurias bus vertinamos mokomosios kompiuterinės priemonės:

- **Valdymo paprastumas** – tai svarbus kriterijus, kadangi sudėtingas programos valdymas sumažina besimokančiųjų motyvaciją mokintis. Programos struktūra turi būti aiški, logiška. Turi būti aiškus joje navigacijos būdas, kad mokinys bet kuriuo momentu turėtų galimybę nutraukti darbą ir žino kaip tai padaryti
- **Vartotojo sąsajos patrauklumas** – patrauklus programos dizainas gali sukelti besimokančiojo susidomėjimą. Mokomajai programai svarbu ne tik turinys, bet ir jo pateikimas kompiuteriu: išdėstymas, grafika, šriftai.
- **Programos pritaikomumas darbui pamokoje** – programa turėtų būti tokia, kad ją būtų galima pritaikyti per pamokas: keisti sudėtingumo lygį, medžiagos turinį, laisvai naviguoti tarp temų. Programos turinys neturėtų prieštarauti kitų mokomųjų priemonių turiniui (vadovėliui), sutapti terminai. Pasirinkti testus pagal nagrinėjamą temą.
- **Teorinė medžiaga** – mokojoji aplinka turi pateikti ir tam tikrą teorijos dalį. Teorinė medžiaga turi būti glausta, aiški. Ji turi atitikti programą. Parengta profesionaliai, atitikti besimokančiųjų amžių. Ji neturėtų būti įprastinis vadovėlių atitikmuo.
- **Testavimo galimybė** – šis kriterijus yra svarbus, nes mokymosi procese yra svarbu ne tik žinių įgijimas, supratimas, bet ir galimybė tas žinias patikrinti ir praktiškai pritaikyti. Tai vienas iš būdų moksleiviams patikrinti įgytas žinias.
- **Užduočių pateikimas** – svarbu, kad mokinys galėtų įgytas teorines žinias praktiškai pritaikytų atlikdamas užduotis.

- **Moksleivių įvertinimas** – automatizuotas moksleivių įvertinimas aktualus ne tik mokiniams, bet ir mokytojams. Mokytojui tai patogu, nes nereikia darbų tikrinti asmeniškai, o moksleiviai turi galimybę įsivertinti savo žinias.

- **Turinio redagavimas** – tai su informacijos naujumu susijęs kriterijus. Turinio redagavimo galimybė suteikia mokytojui pakeisti pateiktą informaciją arba pačiam sukurti testus mokinių žinių patikrinimui.

- **Daugialypės terpės panaudojimas** – moksleiviui sudominti ir pritraukti jo dėmesį gali įvairūs garso bei interaktyvūs vaizdo intarpai.

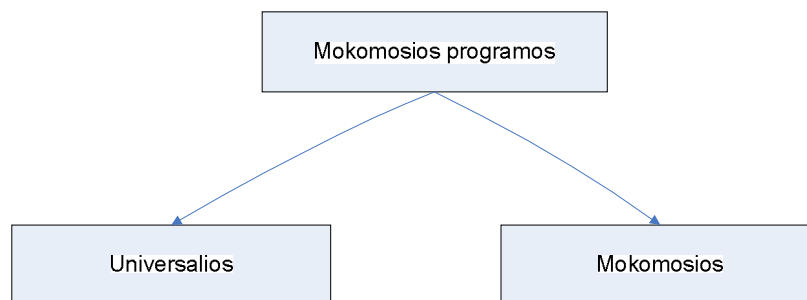
- **Darbas kompiuterių tinkle** – tai galimybė, kuri leidžia įdiegti programą viename kompiuteryje ir pasiekti iš bet kurio kito į tinklą įjungto kompiuterio.

- **Techninis suderinamumas su naudojama programine įranga** – svarbu, kad kompiuterinė priemonė būtų suderinta su mokyklose naudojama programine įranga, netrikdytų kompiuterių darbo. Programos įdiegimas turi būti paprastas, nereikalauti specialių žinių. Programą turi sugebėti pasileisti kiekvienas mokinys.

3.2 Mokomųjų programų klasifikacija

Mokomųjų programų yra įvairių: nuo paprastų programų skirtų vieno dalyko kuriai nors temai mokyti iki sudėtingų skirtų modeliuoti. Mokomosios programos yra klasifikuojamos įvairiai. Bet dažniausiai klasifikuojama pagal turinį ir paskirtį [6]

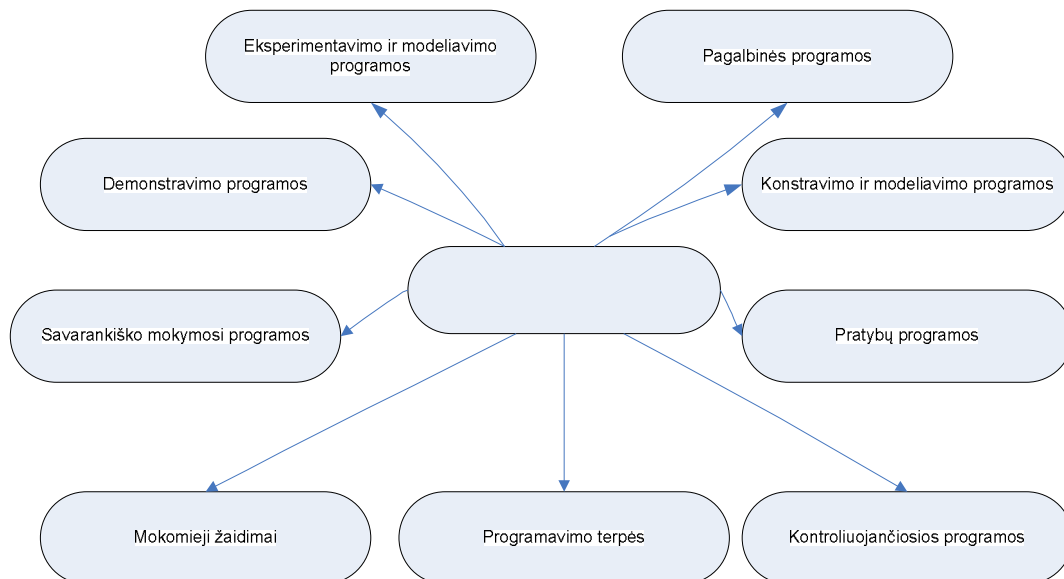
Pagal turinį mokomosios programos skirstomos:



3.1 pav. Mokomųjų programų skirstymas pagal turinį

Universalios programos nėra skirtos kokiam nors dalykui mokyti. Jos skirtos taikyti įvairiems dalykams, siekiant įvairių mokymo tikslų.

Visas mokomąsias programas galima suklasifikuoti, atsižvelgiant į tai ką šios programos leidžia besimokančiajam, kitaip sakant, kiek tas besimokantysis gali jas valdyti. Todėl jas galima suklasifikuoti į tokias kategorijas (žr. 3.2 pav.).



3.2 pav. Mokomųjų programų skirstymas

Demonstravimo programos dažnai atlieka įprastų demonstravimo priemonių funkciją. Tačiau kompiuterinės demonstravimo priemonės paprastai būna pranašesnės už įprastines, kadangi kompiuterio ekrane gali būti pademonstruoti sudėtingi eksperimentai, pateikti sudėtingų reiškinių modeliai. Demonstruojant derinami keli informacijos pateikimo būdai (tekstas, garsas, vaizdas); demonstravimo priemonės dažnai pasižymi interaktyvumu (stebimą vyksmą galima sustabdyti, pakartoti, pakeisti parametrus ir pan.).

Eksperimentavimo ir modeliavimo programos imituoja įvairių reiškinių vyksmą, savybes, mechanizmų veikimą. Jos dažnai naudojamos ir kaip demonstravimo priemonės. Pateikiami modeliai paprastai priklauso nuo įvairių parametrų, kuriuos galima keisti, pasirinkti. Nuo demonstravimo programų skiriasi tuo, kad pasižymi didesniu interaktyvumu.

Konstravimo ir modeliavimo programose galima ne tik keisti modeliuojamojo reiškinio parametrus, bet ir pačiam konstruoti mechanizmus, stebėti jų veikimą, tirti reiškinių dėsningumus, kurti hipotezes ir jas tikrinti. Nuo eksperimentavimo ir modeliavimo programų skiriasi galimybe kurti naujus eksperimentus.

Pratybų programos skirtos įvairioms teorinėms žinioms įtvirtinti ir praktiniams įgūdžiams ugdyti. Jos apibūdinamos kaip mokomojo pobūdžio t.y. šios programos aiškiai nurodo mokiniui ką reikia daryti. Jos daugiau skirtos teorinėms žinioms įtvirtinti ir praktiniams įgūdžiams formuoti. Pirmiausia besimokantysis turi įgyti reikalingų žinių ir išmokti taisykles, po to pasitelkus šias programas atliekamos nesudėtingos užduotys, susidaro reikiami įgūdžiai ir įtvirtinamos žinios.

Kontroliuojančios programos skirtos besimokančiųjų žinioms tikrinti. Nuo pratybų programų skiriasi tuo, kad nėra skirtos mokyti, bet tik patikrinti žinias. Tai gali būti testai, užduotys ir kt.

Savarankiško mokymosi programos pasižymi visų aukščiau minėtų tipų savybėmis. Jos skirtos padėti pačiam mokiniui mokytis vienos ar kitos temos, formuoti reikiamus įgūdžius, patikrinti savo žinias ir t. t.

Mokomieji žaidimai – tai kompiuteriniai žaidimai, kuriuose vyrauja mokomieji elementai. Tai gali būti įvairūs galvosūkių, kryžiažodžiai, modeliavimo žaidimai loginei ir strateginei mąstysenai lavinti.

Pagalbinės priemonės padeda atlikti įvairius nuobodžius ir varginančius veiksmus, leidžia daugiau dėmesio skirti esminiams dalykams. Tai gali būti elektroniniai žodynai, enciklopedijos, žinynai, katalogai.

3.3 Mokymo metodai ir jų klasifikacija

Mokymo metodai klasifikuojami pagal šiuos kriterijus: mokymo informacijos šaltinių pobūdį, mokymo žinių šaltinį, mokinio veiklos pobūdį, bendrą interaktyvią pedagogo ir ugdytinio veiklą.

Pedagogai praktikai, remdamiesi tradiciniu mokymo modeliu, kėlė ir tebekelia pamatinius ugdymo tikslus:

- perteikti informaciją;
- skatinti kūrybiškumą;
- formuoti praktinius gebėjimus.

Metodai, naudojant informacines technologijas ugdymo procese, klasifikuojami pagal įvairius požymius. Siūlomos 6 klasifikacijos.

1. klasifikacija pagal programinės įrangos tipus (aprašyta 3.1 skyriuje);
2. klasifikacija pagal programinės įrangos paskirtį. Išskiriami 3 kompiuterinio mokymo metodai:
 - kompiuteris – mokytojas;
 - kompiuteris – priemonė;
 - kompiuteris – mokymo terpė
3. klasifikacija pagal moksleivio aktyvumo lygmenį:
 - mokymas – sistemos žinios, sistemos tikslai
 - tyrinėjimas - sistemos žinios, moksleivio tikslai
 - konstravimas – moksleivio žinios, moksleivio tikslai
4. klasifikacija pagal moksleivio santykį su mokymosi medžiaga:
 - patirties įgijimas;
 - žinių suteikimas;

- žinių sutvirtinimas;
 - integravimas;
 - pritaikymas.
5. klasifikacijoje pagal kognityvinį poveikį išskiriami tokie metodai:
- kompiuteris – naši priemonė tam, kad moksleivis galėtų daugiau dėmesio skirti aukštesnio lygio veiklai, rutiniams darbams atlikti panaudojamos kompiuterių programos;
 - hipermedia – taikomos asociatyvųjų mąstymą skatinančios informacijos priemonės, kuriose įvairių rūšių informacija vaizduojama ne tiesiniu būdu, o per asociacijas;
 - mikropasauliai.
6. klasifikacijoje pagal mokymosi būdą metodai skirstomi remiantis D.Diujo išskirtais mokymo būdais:
- tyrinėjimas;
 - komunikavimas;
 - konstravimas;
 - išreiškimas.[10]

3.4 Programavimo dalyko mokomųjų veiksmų apibrėžimas pagal Bloomo kriterijus

Pažinimas – tai procesas, per kurį besimokantysis sužino faktus, sąvokas, taisykles, dėsnius. Pažinimo sričiai iškeliami mokymo tikslai susiję su: žinojimu, supratimu, atpažinimu, mąstymu, įvertinimu.

Veiklą su turiniu patogiu susieti remiantis Bloomo taksonomiją, tai priemonė kuri padeda klasifikuoti. 3.1 lentelėje pateikta Pascal programavimo kalbos turinio ir veiklos susiejimas pagal Bloomo taksonomiją.

3.1 Lentelė Pažinimo sritys

Veikla	Turinys
<p>Žinios</p> <p>Žinoti, prisiminti, atpažinti, atkurti faktus, sąvokas, terminus, apibrėžti, apibūdinti klasifikacijas, kriterijus, metodus, taisykles, dėsnius, dėsningumus.</p>	<p>Tai lygmuo, kuris remiasi pagrindinėmis programavimo Pascal kalbos žiniomis. Tam reikia žinoti kintamųjų, tipų, sakinių, konstrukcijų esmę, apibrėžimus ir pavyzdžius.</p> <p>Žino programos struktūra ir programavimo kultūros elementus, programų sudarymo etapus.</p> <p>Supranta tekstinių bylų paskirtį.</p>

<p style="text-align: center;">Supratimas</p> <p>reikšmių supratimas, vertimas, interpretavimas</p>	<p>Sugeba pateikti programos pavyzdžių.</p> <p>Suvokia kintamojo sąvoką.</p> <p>Skiria žinomo ir nežinomojo kartojimo skaičiaus ciklus, jų taikymo ypatybes.</p> <p>Supranta pagrindinius programavimo algoritmus: sumos, sandaugos, kiekio skaičiavimo, rikiavimą, maksimalios (minimalios) reikšmės paiešką.</p> <p>Apibūdina masyvo ir įrašo sąvoka.</p>
<p style="text-align: center;">Taikymas</p> <p>išminktų žinių (sąvokų) teorijų, dėsnių) taikymas naujose situacijose.</p> <p>Gebėjimai pritaikyti, parinkti, modifikuoti, apskaičiuoti, parengti, sukonkretinti</p>	<p>Šis lygmuo – tai sugebėjimas pritaikyti, perkelti įgytas žinias, taisykles įvairiuose programų pavyzdžiuose.</p> <p>Pritaiko realiųjų, sveikųjų, loginį ir simbolinį duomenų tipus.</p> <p>Sugeba sudaryti programas, kuriose pritaiko klasikinius algoritmus: sumos, sandaugos, kiekio skaičiavimo ir kt.</p> <p>Sugeba įvesti (išvesti) duomenis iš klaviatūros (bylos), išvesti į ekraną (į bylą).</p> <p>Taiko uždaviniuose pagrindinius valdymo sakinius bei struktūras: priskyrimo ir sąlygos sakinį, ciklus.</p> <p>Moka aprašyti masyvo tipo kintamuosius, sukuria įrašo duomenų tipą.</p> <p>Geba atlikti veiksmus su masyvo elementais: įvesti ir išvesti duomenis, atlikti paiešką, vartoti reiškiniuose.</p>
<p style="text-align: center;">Analizė</p> <p>Visumos suskaldymas į struktūrinės dalis.</p> <p>Gebėjimai atskirti, išskirti, nustatyti elementus, organizavimo principus, ryšius; išskirti, identifikuoti daiktus, reiškinius apibūdinančius požymius, ypatumus; klasifikuoti, priskirti kategorijai; palyginti elementus, ryšius, priklausomybes;</p>	<p>Tai gebėjimas visumą skaidyti į dalis, elementus. Nustato kokius kintamųjų tipus, programavimo konstrukcijas naudoti konkretaus uždavinio programos sudaryme.</p> <p>Sugeba analizuoti algoritmus, nustato programos struktūrinės dalis: duomenų įvedimą, programos veiksmus apibūdinančius sakinius, duomenų išvedimą.</p> <p>Gebą algoritmą (programą) suskaidyti į atskiras struktūrinės dalis: procedūras ir funkcijas.</p>

sukonkretinti, detalizuoti, išvesti, palyginti	
<p style="text-align: center;">Sintezė</p> atskirų dalių sujungimas į visumą. Gebėjimai susieti, sujungti dalis, susisteminti; apibendrinti medžiagą, praktinę patirtį; atrasti, sukurti ką nors naujo, modeliuoti.	<p>Tai gebėjimas atskiras programos dalis, blokus sujungti į visumą.</p> <p>Sumodeliuoja programą iš atskirų dalių. Sugeba susieti matematinę abstrakciją su technine realizacija: naudoja logines ir matematinės funkcijas programose.</p> <p>Sukuria struktūrinius duomenų tipus, kuriuose panaudoja masyvus ir įrašus.</p> <p>Moka rezultatus išvesti į tekstines bylas.</p> <p>Tinkamai parenka sprendžiamam uždaviniui duomenų struktūras.</p>
<p style="text-align: center;">Vertinimas</p> sprendimų apie medžiagos, idėjų vertingumą priėmimas. Gebėjimai susidaryti nuomonę argumentuoti, pateikti išvadas, logiškai pagrįsti.	<p>Supranta uždavinio skaidymo į dalis privalumus, užrašo programą procedūrų ir funkcijų pagalba.</p> <p>Geba sudaryti aiškia, teisingą programą efektyviausiu būdu.</p> <p>Sugeba suprasti, paaiškinti ir pagrįsti klaidų reikšmes, jas ištaisyti. Moka parinkti kontrolinius duomenis programos teisingumui patikrinti.</p>

Projektuojamoje mokomoje priemonėje bus siekiama įgyvendinti Bloomo kriterijus.

3.5 Mokomųjų programų apžvalga

Programavimo mokymui yra sukurta labai mažai mokomųjų programų. Beveik visos iš jų yra pasenę, neatitinka bendrųjų programų, kitos sukurtos DOS aplinkai. Nuspręsta pasirinkti šias programas, kurias bandysiu paanalizuoti:

- Programavimo pradmenys
- Automatizuotas apmokymo kursas Turbo Pascal 6.0
- Informatika
- PaskMok
- Elektroninis informatikos vadovėlis
- „Komenskio Logo“

Šios programos bus peržiūrimos ir įvertinamos pagal anksčiau aprašytus kriterijus.

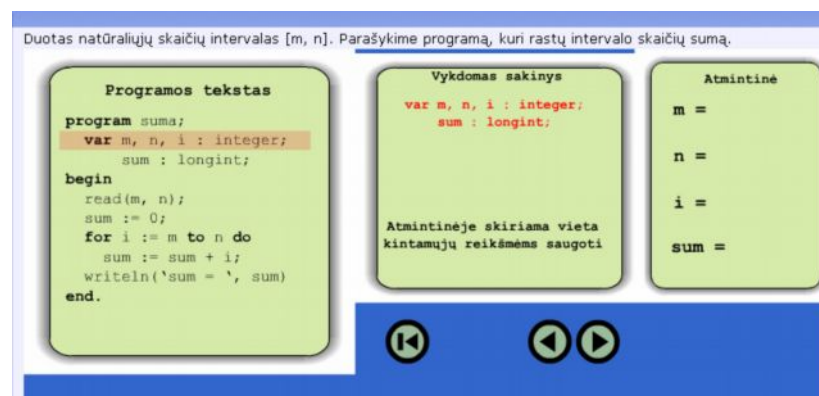
3.5.1 PROGRAMAVIMO PRADMENYS

Viena iš sėkmingiausiai naudojamų mokant programavimo pagrindų jaunesnėse klasėse yra Šiaulių Ivinsko gimnazijos mokytojo Arvydo Versecko kompiuterinė programa „Programavimo pradmenys“.



3.3 pav. Programos „Programavimo pradmenys“ pagrindinis langas

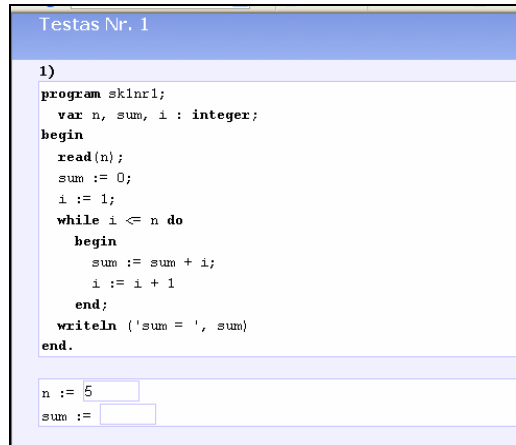
Kompiuteriniame vadovėlyje pateikiamos 14 pagrindinių programavimo pradžiamokslio temų, pratimų ir testų rinkinių. Kiekviename skyriuje yra tekstinė ir animacinė medžiaga. Vartotojas pasirinkęs temą, randa nemažai sudarytų programų. Atvedę pelės žymeklį ant bet kokios programos eilutės virš programos esančioje juostelėje pamatysite tos eilutės komentarą. Daugumai pateiktu programų pavyzdžių galima įsijungti režimą, kurio metu vaizdžiai demonstruojama programos eiga, vaizdinės medžiagos intarpai, galimybė interaktyviai stebėti ir valdyti nagrinėjamą programą (žr. 3.4 pav.).



3.4 pav. Programos „Programavimo pradmenys“ interaktyvios užduoties langas

Programos teorinė medžiaga išdėstyta hiperteksto principu. Programos naudojimas yra paprastas, nes užtenka turėti tik interneto naršyklę.

Žinių pasitikrinimui galima atlikti testus, kuriuose yra pateikta programa ir pradiniai duomenys (žr. 3.5 pav.).



```
1)
program sklnr1;
var n, sum, i : integer;
begin
read(n);
sum := 0;
i := 1;
while i <= n do
begin
sum := sum + i;
i := i + 1
end;
writeln ('sum = ', sum)
end.

n := 5
sum :=
```

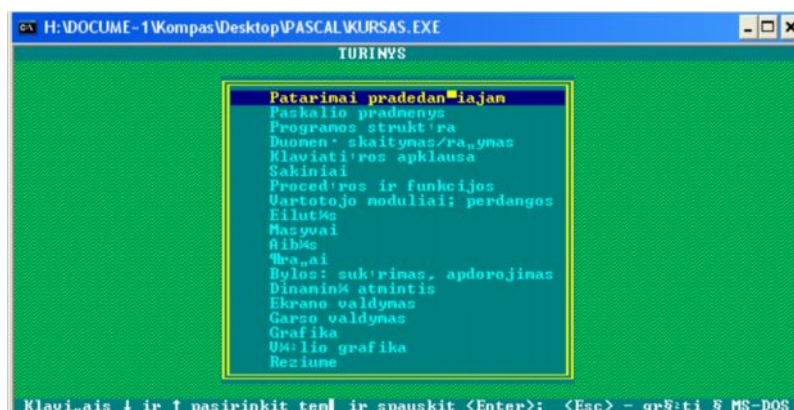
3.5 pav. Programos „Programavimo pradmenys“ testo langas

Yra nemažai originalių užduočių, palengvinančių mokytojui darbą, kai reikia mokiniams pasiūlyti individualias užduotis. Deja, ši kompiuterinė priemonė naudojama tik jaunesnėse klasėse, dėstant programavimo modulį čia pateiktų žinių jau nebepakanka.

Šia programa galima priskirti prie demonstravimo, pratybių, savarankiško mokymosi programų.

3.5.2 Automatizuotas apmokymo kursas Turbo Pascal 6.0

Ši programa yra skirta mokyti pagrindinių Pascal programavimo kalbos konstrukcijų. Teorija pateikta nuosekliai peržiūrint turinį (žr. 3.6 pav.). Kiekvienai temai yra numatytos praktinės užduotys, kurios tinka tiek darbui pamokoje, tiek savarankiškam žinių pasitikrinimui.



3.6 pav. Automatizuoto apmokymo kurso Turbo Pascal 6.0 turinys

Kadangi ši programa sukurta 1993 m. ji yra pritaikyta MS-DOS, todėl yra valdoma tik klaviatūros pagalba. Kompiuteriuose su šiuolaikinėmis operacinėmis sistemomis nekorektiškai vaizduojamos lietuviškos raidės, kai kurie simboliai. Todėl, ją labai sunku yra skaityti ir nagrinėti.

Programa būtų priskiriama demonstruojančių programų tipui.

3.5.3 INFORMATIKA

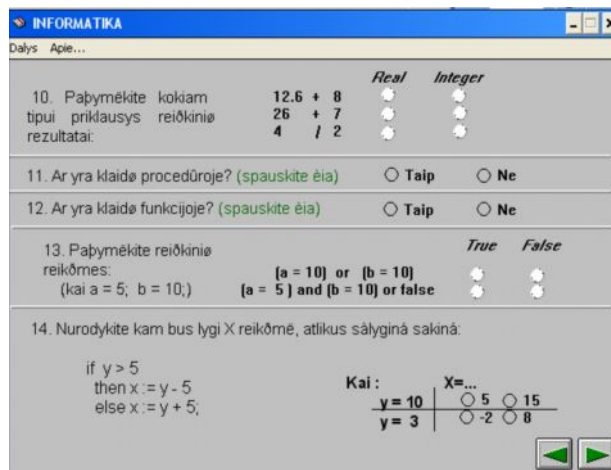
Programa skirta bendrojo lavinimo mokyklų 10-12 klasių mokiniams. Tai kompiuterizuotas V.Dagienės ir G.Grigo vadovėlio “Informatika” variantas - elektroninė knyga su testu žinioms patikrinti.

Interfeisas primena įprastos knygos turinį - čia išvardintos visos programoje siūlomos temos, kurios atitinka pagrindinės mokyklos programavimo programą (žr. 3.7 pav.). Programa patogi tuo, kad teorinė dalis yra išdėstyta suprantamai ir kompaktiškai, mokiniams sudaryta galimybė atžymėti keletą temų ir jas studijuoti nuosekliai. Pateikta daug programų pavyzdžių su paaiškinimais.



3.7 pav. Programos „Informatika“ temų ir užduoties langai

Kiekvieną teorinę temos dalį atitinka praktinė dalis (žr. 3.8 pav.). Tai keleto į dalis skaidomų klausimų paketas, skirtas savarankiškai patikrinti, kaip įsisavinta teorinė medžiaga, medžiagos technologijų analizė ir išpildymas.



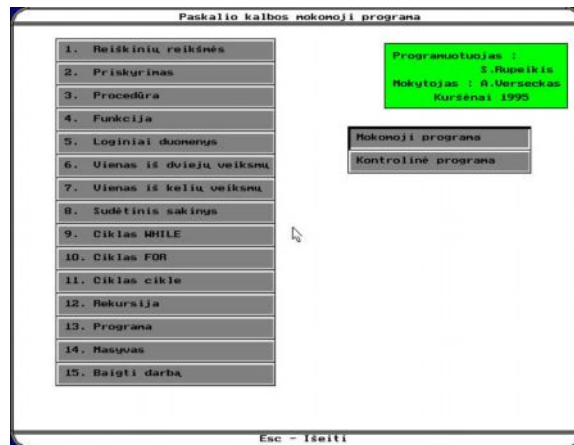
3.8 pav. Programos „Informatika“ testo langas

Programa nors ir sukurta 1997 m., jos valdymas yra gan paprastas, patogiai ir aiškiai pateikiama informacija vartotojui. Tačiau joje nėra interaktyvių elementų. Ji veikia be didelių trukdžių.

Programa galima priskirti prie demonstruojančių, savarankiško mokymosi ir kontroliuojančiųjų programų.

3.5.4 PaskMok

Ši mokomoji kompiuterinė priemonė yra sukurta 1995 metais, mokytojo A.Versecko ir programuotojo S.Rupeikio. Ją sudaro 15 pagrindinių programavimo temų (žr. 3.9 pav.), kurios buvo įdėtos į anksčiau naudotą vadovėlį „Informatika 11-12 klasei“.



3.9 pav. Programos PasMok temų langas

Programa skirta tik žinių patikrinimui testų pagalba, joje nėra teorinės medžiagos. Pasirinkus temą, programa pasiūlo pasirinkti praktinės užduoties tipą: kontrolinė arba bandomoji užduotis. Moksleiviui kiekvienu atveju pateikiama nagrinėti 5 pratimai, kuriuose reikia atlikti algoritmus arba pagal duotą programos fragmentą nustatyti rezultatus, turint pradinį duomenis.

Algoritmo atlikimas yra pakankamai efektyvi moksleivio kontrolės forma, padedanti išsiaiškinti moksleiviui ar suprastas buvo algoritmas ar jo konstrukcija. Atlikus užduotį pateikiama suvestinė lentelės pavidalu, kurioje matoma teisingi atsakymai ir moksleivio gauti rezultatai. Paspaudus klaidingo atsakymo numerį tiek moksleivis, tiek mokytojas gali atlikti klaidos analizę. Gautą įvertinimą programa saugo tik jos darbo metu. Vėliau galimybės pasižiūrėti kaip sekėsi moksleiviui nėra.

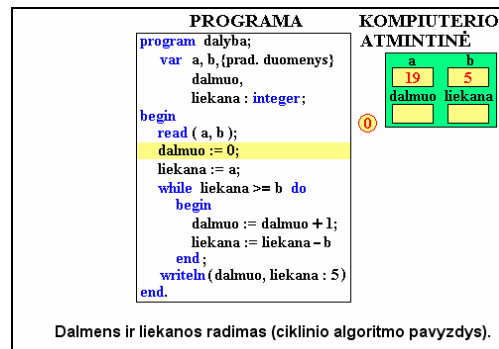
Ši mokomoji priemonė yra sukurta DOS operacinei aplinkai, todėl ją naudoti yra gan sunku, nes kyla problema su operacine sistema, nekorektiškai vaizduojamos lietuviškos raidės. Programa galima valdyti tiek pelės, tiek klaviatūros pagalba. Vartotojo sąsaja nėra patraukli vartotojui, nėra grafikos elementų, kurie pritrauktų dėmesį.

Programa priskiriama kontroliuojančiųjų programų tipui.

3.5.5 ELEKTRONINIS INFORMATIKOS VADOVĖLIS

http://62.80.232.136/informatika/kurybiniai_darbai/vad9_10/algor/algo_00.html

Šis informatikos vadovėlis, patalpintas internete, yra puiki mokomoji priemonė dėstant programavimo įvadą. Jis sudarytas pagal V.Dagienės knygą. Jame yra sudėtos pagrindinės programavimo temos. Plačiai išnagrinėta teorinė medžiaga, yra vaizdinės medžiagos intarpų, iliustruotų pavyzdžių.



3.10 pav. Elektroninio informatikos vadovėlio interaktyvios užduoties langas

Moksleivių žinių patikrinimui galima parsisiųsti testus. Vartotojas prieš atlikdamas testą turi užsiregistruoti įvesdamas vardą arba pavardę. Moksleivis turi galimybę pasirinkti mokomąjį arba kontrolinį testą ir jį atlikęs gauti įvertinamą.

Elektroninis vadovėlis priskiriamas prie pratybų, demonstravimo, savarankiško mokymosi programų.

3.5.6 „Komensio Logo“

Logo – ideali kalba pradedančiam mokytis programavimo bei darbo kompiuteriu pagrindų. Logo – lengva kalba mokymuisi:

- lengva įsidėmėti komandas, nes jos asocijuojasi su įprastais kasdieniniais veiksmiais ir žodžiais;
- draugiška aplinka;
- greita grįžtamoji reakcija.

Logo formuoja gerus programavimo įgūdžius:

- skatinama kurti lakoniškas programas;
- procedūros vartojamos kitose procedūrose (modulinis principas);
- laikomasi struktūrinio programavimo principų. [8]

„Komensio Logo“ programa yra daugiau skirta jaunesnio amžiaus moksleiviams. Joje galima ne tik programuoti, bet ir projektuoti. Ji turtinga šiuolaikinių programavimo kalbų idėjomis bei konstrukcijomis, bei patogiomis priemonėmis programinei įrangai kurti ir projektuoti.

Dirbant su šia programa galima vartoti procedūras, sąryšius. Yra apie 300 įvairių komandų.

Sistema nėra sudėtinga, greitai perprantami pagrindiniai veiksmai, vaizdžiai išreiškiami rezultatai. Yra žinynas, kuris palengvina darbą pradedantiems ir norintiems kurti sudėtingesnius projektus.

Kompiuterio valdomo Vėžliuko įvaldymas supažindina su pagrindinėmis programavimo sąvokomis. Mokinys, valdydamas Vėžliuką, programuoja kompiuterį, jį valdo. Todėl nesunkiai galima sukurti grafinius vaizdus, judesį. Programavimo kursas tampa žaisminga veikla, kurio procese lengviau įsisavinami programavimo įgūdžiai.

Logo aplinka nukreipta ne į taisyklių, teorijos įsisavinimą, o į mąstymo proceso formavimą. Logo formuoja gerus programavimo įgūdžius, skatina kurti lakoniškas programas, laikantis struktūrinio programavimo principų.

Ši programa priskiriama programavimo terpėms.

3.6 Programų palyginimas

Išnagrinėjus pasirinktas programas, matosi, kad mokytojai naudojami neatitinkančiomis jų poreikių programomis. Pagrindinė priežastis tai būtų mažas kompiuterinių programų pasirinkimas mokant programavimo pradmenų. Mokytojai priversti patys kurti programėles, kūrybiškas užduotis, vykdyti apklausas raštu. Bet to neužtenka pamokų pajavairinimui, moksleivių sudominimu programavimu. Šios programos nesudaro galimybės mokytojams patiems pasirinkti ir modifikuoti testų ar praktinių užduočių. Negalima pasirinkti patiems vertinimo skalės. Nėra galimybės analizuoti kiekvieno mokinio įvertinimus. Neigiamą įspūdį mokiniams sudaro ir tai, kad dauguma programų sukurtos DOS aplinkai, dažnai sutrikdomas kompiuterio darbas.

Išnagrinėjus mokytojų ir mokinių atsakymus į pateiktas anketas, jų nuomonę, aprašytus vertinimo kriterijus, išanalizavus pačias programas, jas galima būtų palyginti lentelėje. Programos vertinamos pagal tokią pasirinktą skalę:

- 1 – Neatitinka kriterijaus
- 2 – Nepilnai atitinka kriterijų
- 3 - Atitinka kriterijų
- („—“ – kriterijus nebūdingas programai)

3.2 lentelė Mokomųjų programų palyginimas

	Programavimo pradmenys	Automatizuotas apmokymo kursas Turbo	PaskMok	Informatika	Elektroninis informatikos vadovėlis	„Komenskio Logo“
Valdymo paprastumas	3	1	1	3	3	3
Vartotojo sąsajos patrauklumas	3	1	1	2	3	3
Programos pritaikomumas darbui pamokoje	3	2	1	2	3	3
Teorinė medžiaga	2	3	1	2	3	-
Testavimo galimybė	2	1	2	2	2	-
Užduočių pateikimas	3	2	2	1	2	-
Moksleivio vertinimas	2	2	1	1	2	-
Turinio redagavimas	1	1	1	1	1	-
Daugialypės terpės panaudojimas	3	1	1	1	3	-
Techninis suderinamumas su naudojama programine įranga	3	1	1	2	3	3

Kadangi programos skiriasi savo paskirtimi: vienos orientuotos teorijos perteikimui (Automatizuotas apmokymo kursas Turbo Pascal 6.0), kitos daugiau skirtos žinių pasitikrinimui („PasMok“, „Informatika“). Skiriasi ir sukūrimo metai, tuo metu buvusios ir panaudotos technologijos. Todėl negalima spręsti apie jų kokybę tik iš lentelėje pateiktų vertinimų.

Sprendžiant iš apklausos rezultatų yra žinomos ir naudojamos tik dvi: „Programavimo pradmenys“, „Elektroninis informatikos vadovėlis“. Kitas, tiek mokytojai tiek mokiniai, naudojami pritaikydami savo poreikiams, bei pagal savo galimybes.

4 Elektroninės priemonės projektavimas

Darbe yra siekiama sukurti vizualiai patrauklią, nesudėtinga naudoti mokymo priemonę „Pascal programavimo kalbos valdymo konstrukcijos“, kuri padėtų mokyti programavimo pradmenų. Bus remiamasi atliktu tyrimu apie moksleivių ir mokytojų požiūrį į esamų mokomųjų kompiuterinių priemonių panaudojimą mokantis programavimo.

Priemonė skirta pagrindinėm Paskalio konstrukcijom: priskyrimo sakiniui, sąlygos sakiniui, ciklams. Apžvelgiama šių konstrukcijų sintaksė, savybės. Išskirti ciklą panaudojimo atvejai, jų sudedamosios dalys, nustatytos taisyklės kas būdinga vienai ar kitai ciklo sudedamajai daliai. Daugelyje skyrių pateikta po keletą jau parašytų programų, ar programos fragmentų, tam kad teorinis dėstymas būtų vaizdesnis ir suprantamesnis. Panagrinėti dažniau naudojami algoritmai, kaip sumos, sandaugos, kiekio, aritmetinio vidurkio skaičiavimas.

Ši mokymo priemonė – tarsi elektroninis vadovėlis su interaktyviais pavyzdžiais ir savikontrolės pratimais.

4.1 Mokomosios programos sudarymo modelis

Programavimo mokymosi esmė yra geras pasirinktos programavimo kalbos konstrukcijų išmanymas, jų veikimo principai, įvaldymas veiksmų su įvairių struktūrų duomenimis. Programavimo kalba – tai konkrečios priemonės, skirtos uždaviniui sudaryto algoritmo užrašymui. Gerai įsisavinti pagrindus galima tik tada kai bandoma pačiam rašyti, analizuoti kitų padarytus programų pavyzdžius, yra galimybė žinias patikrinti testų, savikontrolės užduočių pagalba.

Kuriant mokomąją priemonę siekiama įgyvendinti ankstesniuose skyriuose iškeltus uždavinius. Kompiuterinei mokomajai priemonei keliami šie reikalavimai:

- Mokojoji priemonė turi pateikti tam tikrą kiekį teorinės medžiagos žinių įgijimui ir taisyklių išmokimui.
- Pascal programavimo kalbos pradžiamokslio teorija turi būti atrinkta ir suskirstyta į skyrius; Kiekvienai temai pateikiama teorinė medžiaga apie Pascal konstrukcijas ir algoritmų fragmentus; Teorija iliustruojama programų pavyzdžiais, demonstraciniais intarpais, interaktyviais pavyzdžiais apie vieną ar kitą Pascal konstrukciją; Mokinys turi galimybę žinias ir taisykles pritaikyti pavyzdžiuose, kurias siūlo mokojoji programa;
- Mokojoji priemonė turi gebėti patikrinti mokinių žinias ir palaikytų dialogą su mokiniu, o mokiniui leistų patikrinti žinias įvairių savikontrolės pratimų pagalba;
- Mokiniui turi būti sudarytos sąlygos tyrinėti: jis turi matyti pradinis duomenis bei rezultatus ir turėtų nuspėti kas vyksta. Svarbu, kad programoje būtų pereinama nuo atskirų atvejų iki bendresnių, daromos išvados; Sudaroma galimybė pačiam sumodeliuoti programėlę.
- Reikia sukurti patogią ir aiškia vartotojo sąsają, numatyti vizualų teorijos pateikimo būdą, galimybę laisvai pasirinkti mokymosi temas;
- Numatyti galimas mokomosios priemonės tobulinimo galimybes.

4.2 Mokymo proceso organizavimas

Mokymasis naudojantis šia programa gali vykti įvairiai:

- Nagrinėjantis programų pavyzdžius
- Savarankiškai atlikinėti užduotis ir tokiu būdu patikrinti žinias.

Informacinių technologijų mokytojai galės taikyti šią priemonę aiškindami naują temą arba teorijos kartojimui, nes bus patalpinta ir vaizdinės medžiagos. Šios priemonės pagalba mokinys galės savarankiškai mokintis ruošiantis pamokai, savarankiškai pakartoti, užpildyti likusias žinių spragas arba pasiruošti valstybiniam informacinių technologijų egzaminui.

Teorinė tema sudaryta iš atskirų temų, kurios yra iliustruojamos programų pavyzdžiais.

Susipažinęs su pateikta teorinę medžiagą, mokinys galės atlikti savikontrolės testus. Savarankiškai atlikus testus, iškart bus sužinoma į kiek klausimų atsakyta teisingai. Mokiniai savo įvertinimą matys taškais, juo mokytojas galės konvertuoti į pažymius. Savikontrolės dalyje bus galima atlikti ne tik testus, bet ir praktines užduotis.

Mokytis su šia kompiuterine priemone bus galima ne tik klasėje pamokų metu, bet ir laisvu laiku mokyklos bibliotekoje, kompiuterių klasėje, bet ir namuose.

Mokiniai turi turėti bendruosius darbo su kompiuteriu įgūdžius. Bet kadangi ši programa skirta vyresnių klasių mokiniams, tai naudojimosi šia programa problemų neturėtų būti, nes reikalinga bus tik priemonė – Internet Explorer.

4.3 Programos paskirtis

Mokomoji kompiuterinė programa „Pascal programavimo kalbos valdymo konstrukcijos“ pagal savo turinį bus dalykinė mokomoji programa skirta mokyti konkrečiam dalykui t.y. programavimo mokymui Pascal programavimo kalba.

Pagal paskirtį tai bus mokymo ir mokymosi programa. Ji turėtų palengvinto mokytojo darbą, mokiniui sudarytų sąlygas lengviau įsisavinti Pascal valdymo konstrukcijas. Pagal tai kokiai veiklai ji skirta, ją galima būtų priskirti prie:

- Demonstravimo programų, nes yra programų pavyzdžių, bus derinama keletas informacijos pateikimo būdų (vaizdas ir tekstas).
- Kontrolinė programa, nes sudaryta galimybė tikrinti mokinių žinias.
- Pratybų programa, nes sudaryta galimybė teorinėms žinioms įtvirtinti ir praktiniams įgūdžiams ugdyti.
- Savarankiško mokymosi programa – gali būti naudojama mokinių savarankiškam mokymuisi.

Programą pagal moksleivio santykį su mokymosi medžiaga galima būtų priskirti prie:

- Žinių suteikimo – programa naudojama žinioms perteikti.
- Žinių įtvirtinimo – programa taikoma žinioms įtvirtinti. Tai galima atlikti per testus ir interaktyvių pratimų pagalba.
- Žinių pritaikymo – įgytas žinias pritaikyti konstruojant programas testų kūrimo įrankio TestTool pagalba.

Mokymosi veikla turi būti ne slopinanti, gniuždanti, o skatinanti mokymąsi, pažinimą, ugdyti norą save išreikšti ir save tobulinti. Mokymosi veikla turi būti individualizuojama t.y. mokomasi skirtingu tempu, ritmu, skirtingais būdais, pasiekiami ir skirtingi rezultatai. Vieni mokiniai geriau išmoksta dirbdami individualiai su mokytoju, kiti dirbdami savarankiškai. Kuriam mokomoji kompiuterinė programa turėtų sudaryti sąlygas savarankiškai mokytis, ji turi perteikti mokiniams šiuo atveju fundamentalias programavimo Pascal kalbos žinias, pačią sistemą, išmokyti juos taikyti tas žinias praktiniame darbe, modeliuojant reiškinius Suformuoti gebėjimą nagrinėjamuose pavyzdžiuose išvelgti taisykles, kurias vėliau galima būtų pritaikyti panašių uždavinių sprendime. Kartu būtų ugdomas gebėjimas nuosekliai, struktūriškai, algoritmiškai mąstyti, kūrybingai improvizuoti. Ši kompiuterinė mokomoji programa ne tik būtų teorinės dalies perteikimui skirta, bet daugiau skirta programų modeliavimui, užduočių atlikimui ir įgytų žinių pasitikrinimui. Pagrindinis šios programos tikslas būtų – žinių suteikimas ir jų taikymas, bei spragų užpildymas.

4.4 Programos struktūra

Programa sudaryta iš dviejų dalių:

- Teorinės medžiagos pateikimo
- Savikontrolės

Teorinės medžiagos dalyje pateikiama teorinė medžiaga apie Pascal programavimo kalbos valdymo elementus: priskyrimo sakiny, sąlygos sakiny ir ciklai.

Šioje dalyje mokinys gali ne tik susipažinti su teorija, bet ras ir interaktyvių pratimų paaiškinančių vieną ar kitą konstrukcijos veikimą gyvenimišku pavyzdžiu. Kiekvienoje temoje yra programų pavyzdžių, iliustruojančių pagrindinius algoritmus.

Savikontrolės dalyje pateikiami 2 tipų testai: savikontrolės ir kontroliniai. Mokinys turi galimybę pasirinkti norimą testą. Kiekvieną testą sudaro 5-10 klausimų. Už kiekvieną teisingą atsakymą yra gaunama nuo 1 iki 2 balų. Atlikus testus yra pateikiama lentelė su įvertinimo rezultatais.

4.5 Mokymo medžiagos pagrindinės temos

Pascal programavimo kalbos kompiuterizuotoj mokymo priemonėj nagrinėjamos šios pagrindinės temos:

- Priskyrimo sakiny
- Sąlygos sakiny
- Ciklai

Sudėtingesnės temos yra dar smulkiau padalintos į potemes. Pavyzdžiui, temą „Ciklai“ sudaro dar smulkesnės potemės:

- Ciklo samprata
- Ciklas While
- Ciklas For
- Ciklų panaudojimo atvejai
- Ciklų konstrukcijų ypatumai ir taisyklės

4.6 Mokomosios programos reikalavimų specifikacija

Analizuojant moksleivių ir mokytojų poreikius bei pageidavimus, sudarytas reikalavimų sąrašas. Reikalavimai suskirstyti į:

- Vartotojų reikalavimai
- Funkciniai reikalavimai – tai programos kūrimą įtakojantys veiksmai ir nurodymai;
- Nefunkciniai reikalavimai – tai reikalavimai, kurie aktualūs vartotojui, bet neturi įtakos programos veikimui.

4.6.1 Vartotojų reikalavimai

- Programa turi būti nedidelės apimties
- Nesudėtingas valdymas, galimybė nuo vieno mokomojo objekto pereiti prie kito
- Patogi ir aiški vartotojo sąsaja. Valdymas vyksta pasirinkimų mygtukų ir meniu pagalba.
- Mokymo priemonė turi atitikti savo turiniu ir struktūra vidurinės mokyklos programavimo modulio reikalavimus
- Programa turi būt pritaikoma darbui pamokoje, nereikalauti papildomos techninės ir programinės įrangos
- Aiškiai, trumpai ir nesudėtingai pateikta teorinė dalis, kuri būtų suskaidyta atskiromis temomis ir pamokomis

- Testų pateikimas
- Praktinių užduočių pateikimas
- Interaktyvūs pavyzdžiai
- Pagal poreikius redaguoti teorinę medžiagą.

4.6.2 Nefunkciniai reikalavimai

- Mokomosios programos elgsena turi atitikti šiuolaikinius vartotojo grafinės sąsajos reikalavimus, nesunkiai suprantama
 - Lengvai įdiegiama
 - Programa valdoma pelės pagalba
 - Vartotojo sąsajoje turi būti naudojami standartiniai Windows objektai: meniu, mygtukai, pranešimų langai
- Mokomosios priemonės vartotojo sąsaja turi būti skirta besimokančiajam jau mokančiam dirbti kompiuteriu Microsoft Windows operacinėse sistemose ir su Microsoft Office taikomųjų programų paketu
 - Mokojoji programa turi veikti saugiai, nekelti grėsmės kitai programinei įrangai, nesukelti programų „pakibimų“
 - Mokomosios programos kūrimui naudoti hiperteksto technologijas.
 - Programa turi veikti įvairiose interneto naršyklėse

4.6.3 Funkciniai reikalavimai

Panaudojant esamų kompiuterinių priemonių analizę sukurta priemonė turi tenkinti funkcinis reikalavimus. Programa turi būti efektyvi ir suprantama besimokančiajam, kad besimokantysis matytų jos naudą, galėtų plėsti žinių ratą.

- Pateikti medžiaga apie Pascal programavimo kalbą, jos valdymo elementus;
- Pateikti teorinę medžiagą iliustruojančius interaktyvius pavyzdžius, būdingus vienai ar kitai temai;
- Pateikti praktines užduotis;
- Pateikti priemonės žinių pasitikrinimui;
- Pateikti galimybes besimokančiajam pačiam sumodeliuoti programą.

4.7 Programos panaudojimo atvejai

Kuriant programą buvo apgalvoti veiksmai, kuriuos leistų atlikti mokomoji kompiuterinė priemonė. Sistemoje aprašyti 3 vartotojai:

- Administratorius
- Mokytojas
- Mokinys

Mokytojas šią priemonę gali naudoti:

- Naujos medžiagos nagrinėjimui X ir XII klasėje, teorinių žinių įtvirtinimui;
- Teorijos pakartojimui, įtvirtinimui: priemonėje esanti trumpa teorinė medžiaga leidžia mokiniams pakartoti ir įtvirtinti jau įgytas žinias;
- Atlikti apklausą, palengvinti vertinimą.

Kiekvienas iš aprašytų vartotojų galės atlikti vienus ar kitus jam skirtus veiksmus. Šių vartotojų preliminarūs atliekami veiksmai yra pateikiami paveiksle (žr. 4.1 pav.):



4.1 pav. Programos vartotojai ir panaudojimo atvejai

Kiekvienas iš programos panaudojimo atvejų, kelia tam tikrus reikalavimus pačiai programai, tiek vartotojų, funkcinis, tiek ir nefunkcinis. Pateiktose lentelėse aprašyti svarbesni vartotojų veiksmai ir jų veikla (žr. 4.1 lentelė., 4.2 lentelė., 4.3 lentelė., 4.4 lentelė., 4.5 lentelė., 4.6 lentelė, 4.7 lentelė).

4.1 Lentelė Veiklos „Programos įdiegimas“ vartotojų veiksmai ir veikla

Veikla	„Programos įdiegimas“
Dalyvis	Mokytojas, administratorius
Reikalavimai	Lengvai įdiegiama. Mokomoji programa turi veikti saugiai, nekelti grėsmės kitai programinei įrangai, nesukelti programų „pakibimų“.
Veiksmas	Programos įdiegimas

4.2 Lentelė Veiklos „Kurso kūrimas“ vartotojų veiksmai ir veikla

Veikla	„Kurso kūrimas“
Dalyvis	Mokytojas
Reikalavimai	Pagal poreikius redaguoti teorinę medžiagą. Mokymo priemonė turi atitikti savo turiniu ir struktūra vidurinės mokyklos programavimo modulio reikalavimus
Veiksmas	Galimybė įtraukti naujas temas į jau paruoštų temų sąrašą.
Alternatyva	Galimybė pagal poreikius koreguoti teorinę medžiagą.

4.3 Lentelė Veiklos „Temos pasirinkimas“ vartotojų veiksmai ir veikla

Veikla	„Temos pasirinkimas“
Dalyvis	Moksleivis
Reikalavimai	Patogus ir paprastas temos pasirinkimas Aiškiai, trumpai ir nesudėtingai pateikta teorinė dalis, kuri būtų suskaidyta atskiromis temomis ir pamokomis.
Veiksmas	Pasirinkimas dominančios temos iš meniu sąrašo mygtukų pagalba
Alternatyva	Bet kuriuo momentu besimokantysis gali suaktyvinti kitą pamokos temą ar mygtuką ir pereiti prie kitos teorinės pamokos, užduoties, ar testo.

4.4 Lentelė Veiklos „Pamokos pasirinkimas“ vartotojų veiksmai ir veikla

Veikla	„Pamokos pasirinkimas“
Dalyvis	Mokinys
Reikalavimai	Pamokų pasirinkimai turi būti išdėstyti aiškiai ir suprantamai. Jų valdymas neturėtų būti sudėtingas.
Veiksmas	Pasirenkama reikalinga tema Pateikiamas pamokų sąrašas Pasirenkama norima pamoka
Alternatyva	Mokinys turi galimybę nutraukti teorijos skaitymą ir pereiti prie sekančios pamokos ar temos. Tai pat jis turi galimybę peržiūrėti vaizdinę medžiagą iliustruojančią skaitomą teoriją.

4.5 Lentelė Veiklos „Teorijos skaitymas“ vartotojų veiksmai ir veikla

Veikla	„Teorijos skaitymas“
Dalyvis	Mokinys
Reikalavimas	Teorija turi būti išdėstyta aiškiai, glaustai, suprantamai
Veiksmas	Pasirenkama tema Pasirenkama pamoka Pateikiama teorinė medžiaga
Alternatyva	Mokinys gali pereiti į kitą temą, pamoką arba pasirinkti teorinę medžiagą iliustruojančius interaktyvius pavyzdžius.

4.6 Lentelė Veiklos „Vaizdinės medžiagos peržiūra“ vartotojų veiksmai ir veikla

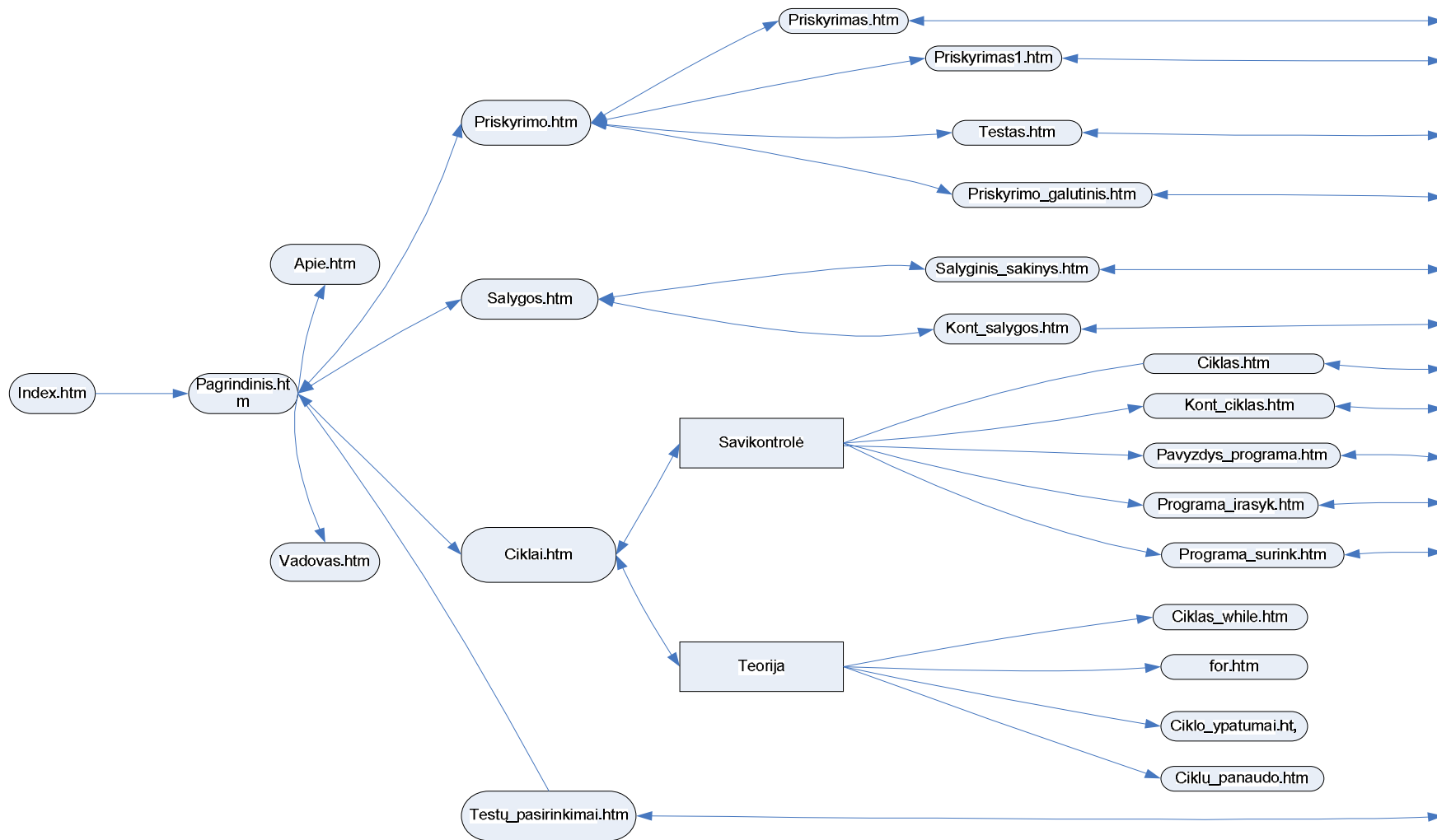
Veikla	„Vaizdinės medžiagos peržiūra“
Dalyvis	Mokinys
Reikalavimas	Pateikti teorinę medžiagą iliustruojančius interaktyvius pavyzdžius, būdingus vienai ar kitai temai
Veiksmas	Pasirenkamas interaktyvus pavyzdys ar užduotis, atitinkanti pasirinktą temą.
Alternatyva	Nutraukiama vaizdinės medžiagos peržiūra arba ji peržiūrima dar kartą, sustabdant reikiamus momentus.

4.7 Lentelė Veiklos „Testų atlikimas“ vartotojų veiksmai ir veikla

Veikla	„Testų atlikimas“
Dalyvis	Mokinys
Reikalavimai	Testų pateikimas Pateikti priemonės žinių patikrinimui.
Veiksmas	Pasirenkamas norimas atlikti testas žinių patikrinimui. Pateikiamas atlikto testo rezultatas.
Alternatyva	Nutraukti testo atlikimą, pereiti prie teorinės medžiagos.

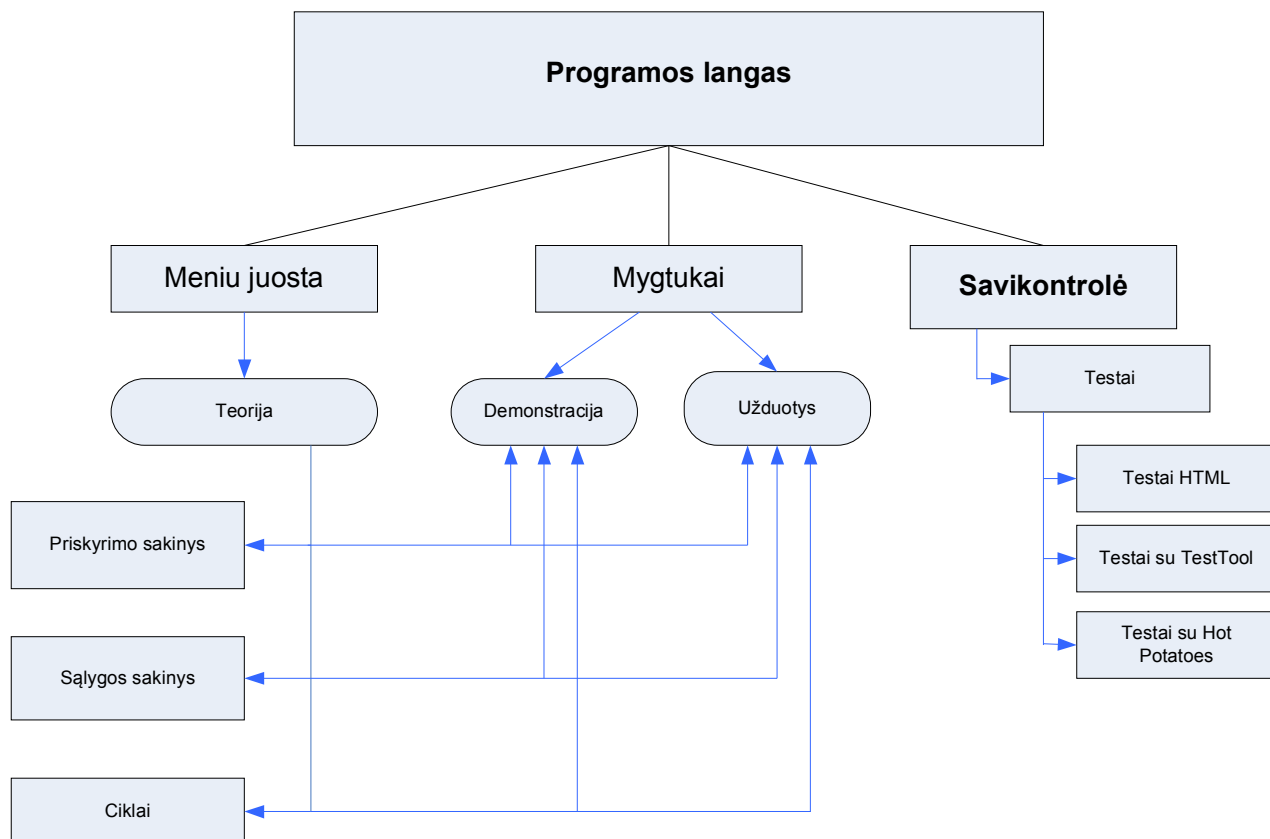
4.8 Programos struktūra

Mokomoji programa realizuojama atskiromis temomis – kurios iškviečiamos pasirinkus pagrindinės formos atitinkamus valdymo elementus - meniu punktus. Pati programos vartotojo sąsaja realizuojama per Internet naršyklę.



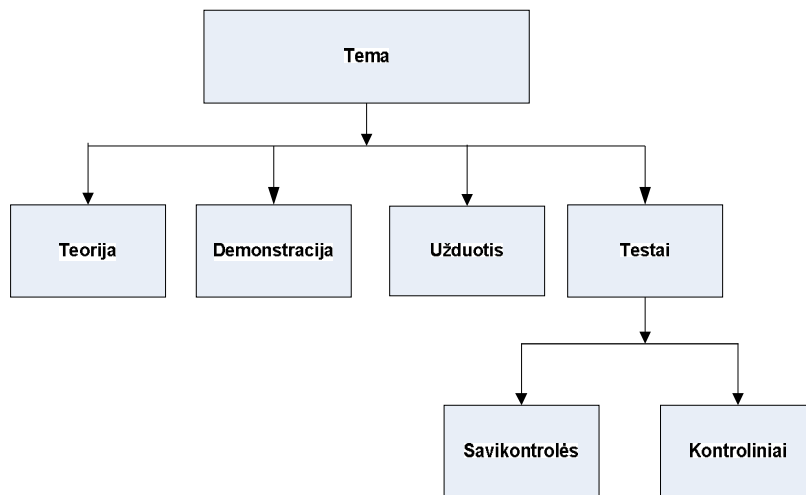
4.2 pav. Programos struktūra

Pasirinkus vieną iš pateiktų meniu punktų atveriamas langas, kuris sudarytas iš atitinkamų punktų, sudarančių menių pasirinkimus, kurių pagalbą galima rinktis priemonės komponentus: teoriją, testo tipą ar užduotį.



4.3 pav. Programos lango tarpusavio ryšiai

Kiekviena tema turi tokią loginę struktūrą:



4.4 pav. Programos temų loginis išdėstymas

Šios mokomosios priemonės temas sudaro:

- Teorija – trumpai ir glaustai pateikta teorinė medžiaga;
- Demonstracija – trumpa animuota medžiaga aprašyti teorinei daliai;
- Užduotis – užduotis, pritaikyti teorijai;
- Savikontrolės testai – testai patikrinti įgytoms žinioms, pateikiant ir įvertinimus;
- Kontrolinis testas – kontrolinis žinių patikrinimas įvedant asmens duomenis.

Kai kuriuos teorinius skyrius sudaro dar atskiros potemės. Pateikiama skyriaus „Ciklai“ langas:



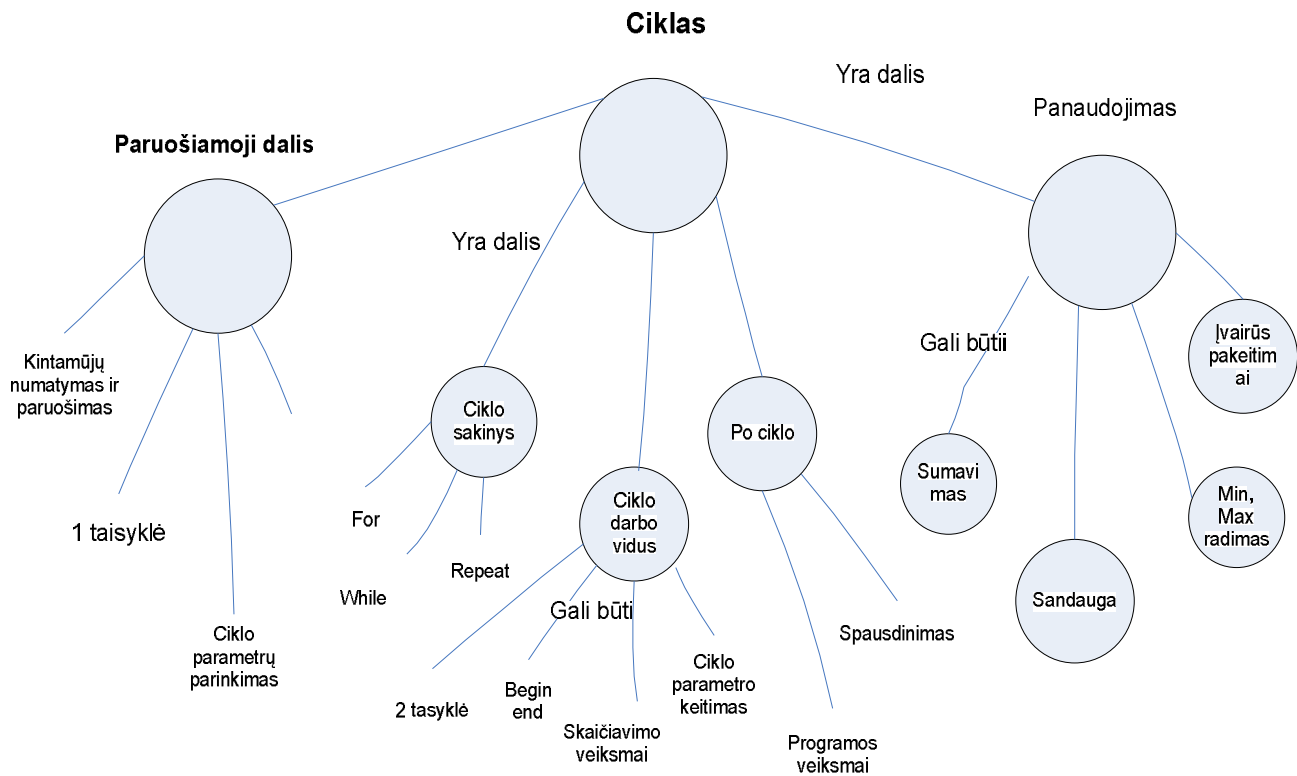
4.5 pav. Programos temos „Ciklai“ langas

Kompiuterinės priemonės kitų teorinių temų langus galima pamatyti prieduose (žr. 77-80 psl., 13 Priedas).

4.9 Temos „Ciklai“ ontologija

Ontologija – tam tikros srities sąvokų visumos specifikuojamas išreikštu pavidalu. Ontologijos yra tam tikros srities terminų ir sąryšio žodynas. Jos kuriamos srities sąvokų ir sąryšių detalizavimui ir tikslinimui, kad susietų ir analizuotų tam tikrų sričių žinias, kad vienas ontologijos būtų galima panaudoti kuriant kitas[9].

Ontologija (žr. 4.6 pav.) apima ciklo struktūros sandarą, nes ciklai yra viena iš sunkesnių programavimo temų moksleiviams. Jis apima ciklo naudojimosi, aiškumo, priklausomybės aspektus. Ši ontologija atsako į keletą klausimų: iš kokių sudedamųjų dalių sudarytas ciklas, kokios sudedamosios dalys jį sudaro, ciklų charakteristikos.



4.6 pav. Ciklo antologija

4.10 Ciklo konstravimo būsenų grafas

Ciklas tai sistema, sudaryta iš sakinių. Į ciklo konstravimą žiūrime kaip į sakinių sekos rašymo procesą, kuriame galima išskirti veiksmus prieš ciklą, veiksmus cikle ir veiksmus sekančius po ciklo. Konstravimas turi vykti pagal taisykles, kurios grafiškai pavaizduotos būsenų kaitos grafe. Šis grafas (žr. 4.8 lentelė.) apima ciklo konstrukcijas bei pagrindinius ciklinius algoritmus: sumavimą, sandaugą, kiekio skaičiavimą ir maksimalios (minimalios) reikšmės radimą.

4.8 Lentelė Ciklo konstravimo eiga ir taisyklės

Užduotis	S0	
Sakiniai prieš ciklą	S1	Jei sumavimas tai $S = 0$, kitaip Jei sandauga tai $S = 1$, kitaip Jei kiekio skaičiavimas tai $S = 1$, kitaip
Ciklo Antraštė	S2	For i:= a to b do begin
Ciklo darbo dalis	S3	Jei sumavimas tai $S = S + i$, kitaip Jei sandauga tai $S = S * i$, kitaip Jei lyginio skaičių suma tai „jei $i \bmod 2 = 0$ “, tai tada $S = S + i$ End;
tolesni veiksmai po ciklo	S4	Jei vidurkio skaičiavimas tai $S = S / (a - b + 1)$
spausdinimas	S5	Jei sumavimas tai spausdinti „Suma“, S, kitaip Jei sandauga tai spausdinti „Sandauga“, S, kitaip Jei vidurkio skaičiavimas tai „Vidurkis“, S, kitaip
pabaiga	S6	

5 Mokymo priemonės įgyvendinimo technologiniai sprendimai

Kuriant mokomąją programą, buvo siekiama įgyvendinti užsibrėžtus tikslus ir uždavinius bei efektyviai panaudoti informacinių technologijų teikiamas galimybes.

Išanalizavus funkcinis ir nefunkcinis reikalavimus, tai pat įvairių realizacijos priemonių galimybes, kuriamai mokomajai kompiuterinei priemonei buvo pasirinktos tokios technologijos:

- Žiniatinklio kūrimo programa Microsoft Office FrontPage
- Interaktyvių užduočių kūrimo programa „Hot Potatoes“
- Nuotolinio grafinio testavimo sistema TestTool
- Darbalaukio filmavimo programa Wink
- Grafikos programa Adobe Photoshop CS2

Panagrinėsiu keletą technologijų pasirinktų užduoties įgyvendinimui.

5.1 Žiniatinklio kūrimo programa Microsoft Office FrontPage

Hipertekstas – teksto pateigimo būdas, kai informacijos vienetai (kompiuterio ekrano puslapiai) išdėstomi ne nuosekliai, bet pagal pasirinktą struktūrą. Hipertekstinis dokumentas leidžia besimokančiam naudotis informacija taip, kaip jam atrodo reikalinga ir būtina. Ši aplinka turi daug privalumų lyginant su kitomis:

- Lengvai galima kurti svetainės navigacijos medį, nuorodas;
- Lengvai modifikuojasi svetainės puslapiai panaudojant šablonus;
- Galimybė peržiūrėti būsimą svetainę prie įvairių ekranų skiriamųjų gebų, bei per skirtingas naršykles;
- Nesudėtingai papildoma svetainė naujais puslapiais, nesudėtinga modifikuoti jau esamus.

Dėl šių priežasčių ir buvo pasirinkta ši programa kuriamai vartotojo sąsajai. Vartotojui suteikiama galimybė pasinaudoti šiomis paslaugomis:

- Teorinė medžiaga išdėstoma į atskirus *.htm failus ir sujungiama į visumą.
- Puiki navigacija, nes vartotojas gali laisvai rinktis temas, pereiti su mygtukais nuo vienos temos prie kitos;



5.1 pav. Temos „Priskyrimo sakiny's“ pagrindinis langas

Microsoft FrontPage byla – svetainė. Mano svetainėje nuorodų mygtukai atitinka temų pavadinimus. Kiekvienoje temoje glausta teorinė medžiaga ir galimybė ją išstudijavus pasitikrinti savo žinias testų pagalba arba atliekant užduotis. Svetainės temų tinklalapių langai pateikti prieduose (žr. 77 psl. 13 Priedas)

Pagrindiniai reikalavimai norintiems dirbti su programa yra šie:

- Kompiuteris turintis naršyklės programą;
- Vartotojas turi turėti elementarius darbo su kompiuteriu ir naršyklės programa įgūdžius.

5.2 Testų ir užduočių kūrimo priemonės

Žinių testavimas yra vienas iš mokymosi proceso grįžtamojo ryšio realizavimo metodų. Testai padeda užtikrinti pakankamai objektyvų žinių, mokėjimų ir įgūdžių kontrolės galimybes. Žinių ir įgūdžių testavimas ugdymo procese užima išskirtinę vietą. Dauguma šiuo metu naudojamų testų yra skirti tik primityviai apklausai, bet nėra skirti praktinių įgūdžių vertinimui bei įtvirtinimui. Darbe yra numatyta atsakyti į keletą esminių žinių ir įgūdžių testavimo klausimų, apimančių praktinį programavimo mokymą t.y. sukurti interaktyvius testus, kurių pagalba mokiniai sumodeliuotu programų fragmentus.

Mokymo priemonėje pateikiami savikontrolės ir kontroliniai testai. Savikontrolės ir kontroliniai testai skirti moksleiviams pasitikrinti savo žinias ir įgūdžius. Testų ir užduočių kūrimui panaudotos 3 priemonės:

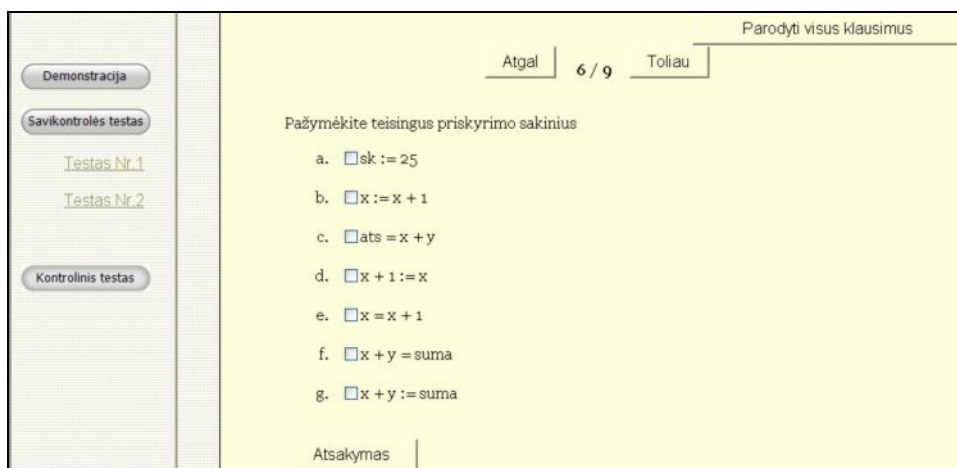
- Interaktyvių užduočių kūrimo programa „Hot Potatoes“;
- HTML kalba;
- Grafinio testavimo sistema TestTool.

Perskaitęs teoriją, mokinys turi galimybę pasitikrinti savo žinias. Kontroliniuose testuose yra sudaryta galimybė mokiniui įvesti savo asmens duomenis. Atlikę savikontrolės ar kontrolinius testus mokiniai mato savo įvertinimus pažymiais arba procentine išraiška (testuose, kurie padaryti su „Hot Potatoes“ įrankiu). Jiems parodoma į kuriuos klausimus atsakė teisingai ir to klausimo svorį balų sistemoj, tai pat pateikiamas ir galutinis įvertinimas pažymiu. Klausimų kiekis testuose svyruoja nuo 5 iki 10. Dalis testų ir užduočių yra sudaryta su nuotolinio testavimo programa TestTool.

5.2.1 Interaktyvių užduočių kūrimo programa „Hot Potatoes“

Testams kurti naudoju interaktyvią užduočių kūrimo programą „Hot Potatoes“. Ši programa tai įrankių programa, suteikianti mokytojams nežinantiems programavimo kalbų, galimybę savarankiškai kurti interaktyvias užduotis. Ypatumas programos tas, kad sukurtas užduotis leidžiama išsaugoti standartinio internetinio puslapio formate, todėl vartotojams nereikia pačios programos, o tik turėti interneto naršyklę. Šios programos pagalba galima sukurti užduotis panaudojant tekstinę, grafinę, audio ir video medžiagą. Užduotys sudaromos su 5 programų blokais.

Bloką JQuiz naudoju testų kūrimui. Galima naudoti 4 rūšių užduotis: su 1 teisingu atsakymu, keletu teisingu atsakymu, atviro tipo atsakymas ir hibridinis variantas, kuris apjungia klausimus su atviro tipo atsakymais ir klausimus su keletu atsakymu.



The screenshot shows a web-based test interface. On the left side, there is a sidebar with buttons for 'Demonstracija', 'Savikontrolės testas', 'Testas Nr.1', 'Testas Nr.2', and 'Kontrolinis testas'. The main area has a yellow background and contains the following text: 'Parodyti visus klausimus' at the top right, 'Atgal 6 / 9 Toliau' in the middle, and 'Pažymėkite teisingus priskyrimo sakinius' followed by seven multiple-choice options (a-g) with checkboxes. At the bottom, there is an 'Atsakymas' input field.

5.2 pav. Testo, sukurto su Hot Potatoes, pavyzdys

Šie testai atliekami tik treniruotės režime, užduoties atlikimo rezultatas apskaičiuojamas procentais. Nesėkmingi bandymai sumažina atliktos užduoties įvertinimą. Daugiau testų klausimų yra pateikta prieduose (žr. 65 psl. 6 Priedas).

Blokas JMatch naudotas priemonės interaktyvumui ir vaizdumui palaikyti. Šis formatas pritaikytas praktinio pobūdžio užduočiai, tam, kad vartotojas galėtų praktiškai patikrinti savo žinias modeliuodamas programą pagal pateiktą sąlygą. Šio tipo užduotys atliekama pelės pagalba nešiojant grafinius objektus į jiems skirtas vietas. Užduoties maksimalus įvertinimas yra 100%. Suklydus procentinis įvertinimas sumažėja, bet yra galimybė ištaisyti klaidas (žr. 76 psl. 12 Priedas).

Blokas JCloze skirtas praleistų žodžių įrašymui. Tam buvo suvesta programa, kurioje buvo padaryti žodžių praleidimai. Šiai užduočiai atlikti reikia klaviatūros pagalba suvesti teisingus žodžius arba raides (žr. 75 psl. 11 Priedas).

Šios programos pagalba sukurtos užduotys turėtų skatinti mokinių motyvaciją, bei padėtų mokytojams ieškantiems naujovių savo darbe.

5.2.2 HTML kalba ir JavaScript

Dialogas žiniatinklyje naudojamas organizuojant formas. Šia priemonė panaudota interaktyvumui padidinti, būtent savikontrolės testų kūrimui. Formos kūrimas pradėtas kurti naujame html tinklalapyje, įterpiančias formas laukelius bei tekstą. Testo klausimams kurti buvo panaudoti šie laukai (žr. 68 psl. 7 Priedas „Testo, sukurtos formų pagalba, HTML kodas“):

- Radio button – įterpia į formą mygtuką – jungiklį.
- Check Box – įterpia į formą žymos langelį. Šio lauko pagalba galima pažymėti keletą langelių ir pasirinkti nurodytus variantus.
- Drop – down – tai galimybė pasirinkti iš alternatyvų sąrašo, naudojant iškrentantį meniu.

Testo pavyzdys naršyklėje:



1. Kokio tipo gali būti ciklo FOR kintamasis?

tik realaus (Real)
 tik sveikąjo (Integer)
 Realaus (Real) ir sveikąjo (Integer)

2. Kiek kartų bus atliekamas šis ciklas

```
for i := 0 to 15 do
```

15
 16

3. Kokia a reikšmė įvykdžius ciklą?

```
n := 5;  
a := 2;  
for i := 3 to n do  
a := a + 1;
```

[-pasirinkite=-]

4. Kiek kartų bus atliekamas sakiny `a := a + k`?

```
a := 0;  
k := 10;
```

5.3 pav. Testo pavyzdys

JavaScript tai objektiškai grįsta programavimo kalba. JavaScript kodai yra įterpiami į HTML dokumentą tiesiogiai. Skriptų rašymo įrankių pagalba galima organizuoti interaktyvų bendravimą su besimokančiu:

- Duomenų įvedimą;
- Testavimą;
- Rezultatų pateikimą.

Prieš pradėdant spręsti testą mokinys yra registruojamas, paprašant įvesti vardą ir pavardę (žr. 5.4 pav.). Mokinio registracijai buvo panaudotas šis įterpiamas elementas:

```
58
59 </SCRIPT><FORM name=testas>
60 <DIV align=center>
61   <p align="left">
62
63     <b><i>
64
65       <font size="3">Pavardė#279;: </font>
66
67     <font size="3" color="#000080"> <input type="text" name="T1" size="30"></font></i></b>
68   <p align="left">
69
70     <b><i>
71
72     <font size="3">Vardas: </font>
```

5.4 pav. JavaScript panaudojimas mokinio registracijai

Kuriant testus formos pagalba, privalu nurodyti ir teisingus atsakymus, priešingu atveju bus rodomi klaidingi atsakymai. Tam yra panaudota funkcija (žr. 5.5 pav.).

```
13 <p align="center"></p>
14 <SCRIPT lang="JavaScript">
15 function testuoti(){
16 var f = document.testas;
17 var balas1 = 0;
18 var balas2 = 0;
19 var balas3 = 0;
20 var balas4 = 0;
21 var balas5 = 0;
22 var balas7 = 0;
23 var balas9 = 0;
24 var suma = 0;
25 var pazym = 0;
26 var pav = "";
27 var vard = "";
28 pav = f.T1.value;
29 vard = f.T2.value;
30
31 if(f.ats_1[0].checked==true) balas1=1;
32 if(f.ats_2[0].checked==true) balas2=1;
33 if(f.ats_3[1].checked==true) balas3=1;
34 if(f.ats_4[2].checked==true) balas4=1;
35 if(f.ats_5[3].checked==true) balas5=2;
36 if(f.ats_7[2].checked==true) balas7=2;
37 if(f.ats_9[2].checked==true) balas9=2;
38 suma =balas1+balas2+balas3+balas4+balas5+balas7+balas9;
39 pazym = suma;
40
```

5.5 pav. Testo teisingumui nustatyti funkcijos panaudojimas HTML kalboje

Alert() yra lango objekto metodas. Jis naudojamas iššokantiems pranešimams su mygtuku OK. Šis metodas buvo panaudotas ekrane pateikti sprendžiamo testo rezultatams. (žr. 68 psl., 7 Priedas „Alert() metodo aprašymas HTML kalboje“)

Testo atlikimas yra efektyvi kontrolės priemonė mokiniui, patikrinant savo žinias, tiek mokytojui, gaunant grįžtamąjį ryšį mokymo procese. Atlikus testo užduotis pateikiama suvestinė lango pavidale, kurioje matomi už kiekvieną klausimą gauti balai (žr. 68 psl., 7 Priedas „Rezultatų langas atlikus testą“). Balų sumavimui ir jų konvertavimui į pažymį naudota funkcija.

4 iš 7 testų klausimams yra skiriama po 1 balą, kitiems 3- 2 balus, atsižvelgiant į tai, kad šie testo klausimai yra sunkesni.

Viso testinės užduotys sukurtos panaudojant HTML kalbos formas, bei JavaScript pateikiamos prieduose (žr. 4, 5, 8, 9 ir 10 Priedus).

5.2.3 TestTool

Tai nuotolinio testavimo programa. Ji leidžia kurti tekstinius ir grafinius testus bei pateikti juos internete. Sistemą sudaro 4 dalys:

- TestTool serveris – saugo duomenis ir suteikia priėjimą prie jų.
- Autoriaus programa – naudojama kurti klausimų variantus, kurie vėliau įkeliami į TestTool serverį.

- Administratoriaus programa – naudojama administruoti TestTool sistemą.

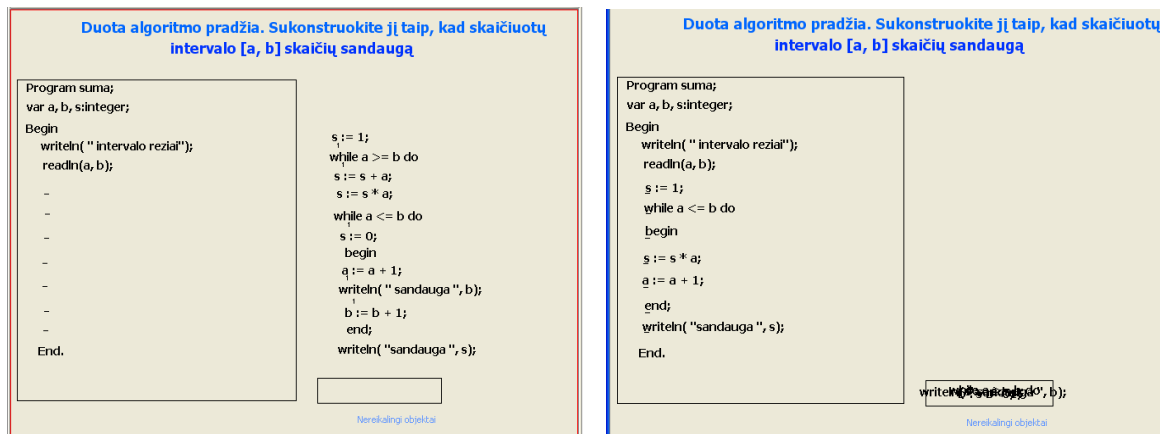
Administratoriui leidžiama valdyti sistemos vartotojus ir grupes, įkelti klausimų variantus, kurti testus bei valdyti testavimo rezultatus bei statistiką.

- Studento programa – naudojama atlikti testus bei studentui pateikia egzamino, testo klausimus.

Ši testų kūrimo programa leidžia kurti šių tipų klausimus:

- Vieno atsakymo pasirinkimas iš pateikto atsakymų variantų
- Kelių atsakymų pasirinkimas iš pateiktų atsakymo variantų
- Grafiniai objektai turi būti išdėstomi tam tikra tvarka
- Keičiama grafinio objekto vieta

Remiantis apklausos rezultatais, mokytojai pageidauja naudoti žinių testavimą. Svarbus elementas teste yra panaudojimas ir vaizdinės informacijos. Mokant programavimo kurso susiduriama su sunkumais: sunku išmokyti naudoti ir taikyti įvairias programavimo struktūras, mokiniai sunkiai suvokia programos sakinių veikimo principus. Pats pirmas sunkumas, kurį tenka įveikti tai sąlygos sakinyje ir ciklai. Todėl ir sukūriau testų pavyzdžių su šiomis struktūromis. Šiems testams panaudojau grafinių objektų išdėstymą tam tikra tvarka. Atsakinėjant į klausimą reikia surašytus sakinius sudėlioti į vietas programos fragmente, o nereikalingus objektus sudėti į jiems skirtą stačiakampį su užrašu „Nereikalingi objektai“ (žr. 5.6 pav.).

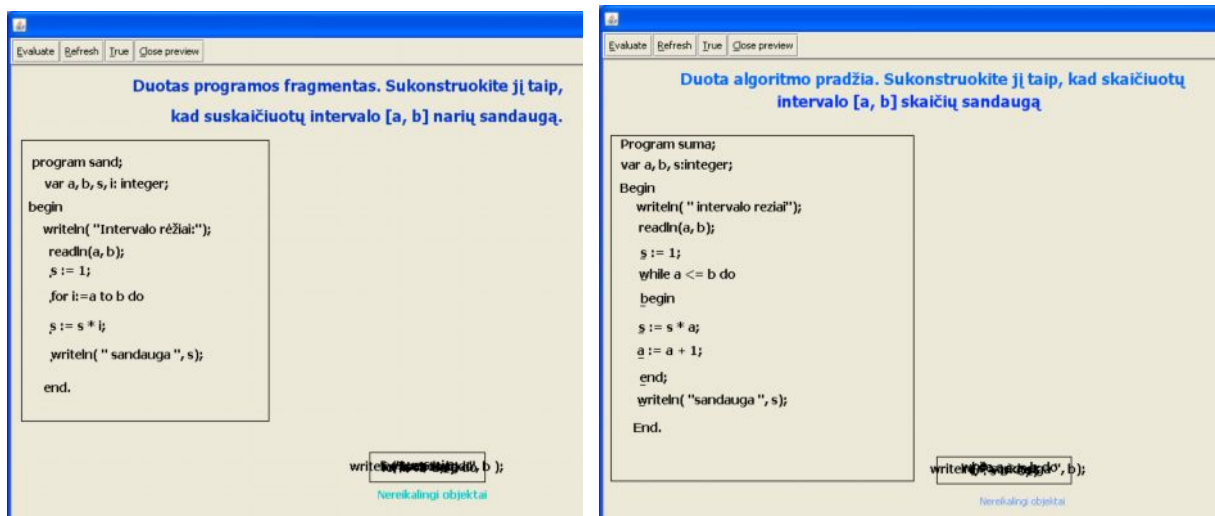


5.6 pav. Testo sukurto su TestTool pavyzdys Nr.1

Kad testai būtų naudingi reikėtų laikytis bendrų jų sudarymo principų. Keletas iš tokių principų galėtų būti :

- Testo turinys

Suprasti kaip veikia viena ar kita konstrukcija, padeda daug panašaus tipo testų, keičiant tik pradines sąlygas (pvz. skaičiuoti sumą, sandaugą) ir ciklo rūšį (ciklas For arba ciklas While) (žr. 5.7 pav.).



5.7 pav. Testo sukurto su TestTool pavyzdys Nr.2

Yra ir daugiau panašių klausimų į pateiktąjį, kuriuose reikia sudėlioti elementus į jiems skirtas vietas (žr. 84 psl.17 priedas). Testuojama turi būti tai, ką svarbu mokėti ir žinoti, vengti atskirų formalių klausimų ir uždavinių.

- Mokymo ir mokymosi

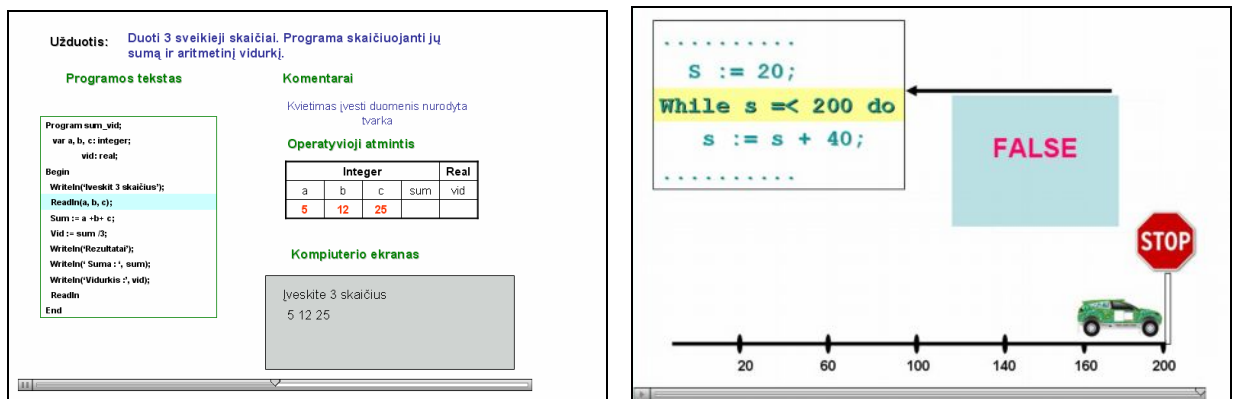
Testavimas turėtų padėti mokymui ir mokymuisi: tirti tai, ką mokinys moka ir geba, o ne „gaudyt“ ko jis nemoka. Testas turi būti viso mokymo proceso sudedamoji dalis. Testui

nereikėtų turėti specialaus pasirengimo, o užtekų pamokos mokiniui išmokti tai, ko reikalauja programa.

Atsakinėjant į tokio tipo testus lavinamas programos struktūros supratimas, suteikiama programavimo žinių.

5.3 Demonstracinių objektų kūrimas

Tik skaitant teorinę medžiagą kartais būna sunku įsidėmėti ar suprasti svarbiausius teiginius. Todėl kompiuterinėje priemonėje yra galimybė peržiūrėti demonstracinius pavyzdžius sukurtus su darbalaukio filmavimo programa Wink. Ši programa - tai priemonė kurios dėka galima realizuoti animuotą vaizdo pateikimą. Naudojantis šią programą parengti demonstraciniai siužetai aiškinantys vieną ar kitą Pascal valdymo konstrukciją (žr. 5.8 pav.). Daugiau su programa Wink sukurtų demonstracinių intarpų pateikiama prieduose (14, 15 ir 16 priedai).



5.8 pav. Programos animuotų užduočių pavyzdžiai

6 Mokomosios priemonės mokymo objektų įgyvendinimas ir testavimas

6.1 Technologinis mokymo objektų įgyvendinimas

Anksčiau aprašytų programų pagalba galėjau įgyvendinti kuriamos priemonės komponentus. Vienos programos buvo skirtos įgyvendinti savikontrolės ir testavimo galimybei, kitos teorinės medžiagos pateikimui ar valdymo paprastumui programoje įdiegti (navigacijai). Lentelėje 6.1 pateiktos technologijos ir, kaip jų pagalba buvo įgyvendinti analitinėje dalyje aprašyti mokomųjų programų kriterijai ir išskelti uždaviniai, kurių pagrindu buvo kuriama elektroninė priemonė programavimui mokyti.

6.1 Lentelė Mokymo objektų technologinis įgyvendinimas

	Naudotos technologijos				
	HTML	TestTool	Java Script	Hot Potatoes	Wink2000
Interaktyvios užduotys, animacija	-	+	±	+	+
Grafinės priemonės	+	+	+	+	+
Valdymo paprastumas	+	+	+	+	+
Testavimo priemonių, užduočių pateikimo galimybė	-	+	+	+	-
Vertinimas	-	+	+	+	-
Rašymo priemonės, teorinės medžiagos pateikimo galimybė	+	+	+	+	±
„+“ - taip, „-“ - ne, „±“ - iš dalies					

6.2 Mokomosios priemonės testavimas

Mokomoji kompiuterinė priemonė buvo testuojama Varėnos „Ryto“ ir „Ažuolo“ vidurinėje mokykloje. Tyrime dalyvavo 10 klasių mokiniai ir 5 mokytojai. 10 klasės buvo pasirinktos dėl to, kad šiuo metu juose yra dėstomos programavimo temos. Kiekvienoje mokykloje yra po 4 dešimtokų klases. Todėl tyrime dalyvavo tik po 2 klases iš kiekvienos mokyklos, po 2 klases nedalyvavo. Tai buvo pasielgta dėl to, kad būtų galima sulygtinti žinių lygį mokinių, kurie naudojami kompiuterine priemone ir mokinių nedalyvavusiu tyrime.

Tyrime tai pat dalyvavo 10 moksleivių pasirinkusių programavimo modulį ir žadančių laikyti informacinių technologijų valstybinį egzaminą. Jiems buvo pasiūlyta savarankiškai susipažinti su mokomąją priemone ir išreikšti savo nuomonę.

6.2.1 Eksperimentinio tyrimo tikslas

Pagrindinis šio eksperimento tikslas – supažindinti mokinius su kompiuterinę programa „Pascal valdymo struktūros“, išsiaiškinti ar ji reikalinga ir naudinga mokymo procese. Tuo pačiu sužinoti mokinių bei mokytojų nuomonę apie ją.

6.2.2 Tyrimo uždaviniai

Norint atlikti eksperimentą reikėjo įgyvendinti tokius tikslus:

- Supažindinti su kompiuterine mokomąją priemone

- Pamokose arba savarankiškam mokymuisi naudoti priemonę
- Po susipažinimo su teorinę medžiagą pateikti testus, kurie turi parodyti kaip, naudojant skirtingus mokymosi būdus mokiniai įsisavino teorinę medžiagą
- Atlikti mokinių ir mokytojų apklausą ir sužinoti jų nuomonę apie programą ir jos naudingumą bei reikalingumą.

6.2.3 Tyrimo eiga

Mokomoji programa buvo pateikta išbandyti mokiniams ir mokytojams. Ją buvo naudojama vidutiniškai po 6 pamokas kiekvienoje klasėje. Atliekant šį eksperimentą buvo panaudoti šie metodai:

- Mokomosios priemonės paruošimas, supažindinimas su ja
- Temos aiškinimas
- Testavimas
- Apklausa
- Rezultatų analizė ir apibendrinimas.

Išėjus kiekvieno skyriaus teorinę medžiagą, buvo atlikti kontroliniai testai. Testą atliko visų 10 klasių mokiniai: tiek tyrime dalyvavę, tiek nedalyvavę. Surinkus savikontrolės ir testų rezultatus, tyrime dalyvavusių mokinių (52 mokiniai), buvo gauta tokia suvestinė (Lentelė 6.2).

6.2 Lentelė Tyrime dalyvavusių testų rezultatų lentelė

Pažymys	Mokinių įvertinimai (52 mok.)	Bendras pažymio vidurkis
	Kiekis	
4	1	7,09
5	3	
6	17	
7	16	
8	5	
9	4	
10	6	

Lentelėje 6.3 pateikta testų rezultatai tų 10 klasių mokinių kurie nedalyvavo tyrime. Tokių mokinių buvo 49.

6.3 Lentelė Tyrime nedalyvavusių testų rezultatu lentelė

Pažymys	Mokinių įvertinimai (49 mok.)	Bendras pažymio vidurkis
	Kiekis	
4	1	6,71
5	4	
6	19	
7	18	
8	1	
9	3	
10	3	

Apibendrinant testų rezultatus tiek dalyvavusių, tiek nedalyvavusių tyrime, galime teikti, kad mokiniai kurie naudojo ir vadovėlį ir kompiuterinę priemonę gavo didesnius įvertinimus, nei tie kurie nesinaudojo priemonę. Taigi galima daryti išvadą, kad kompiuterinės priemonės naudojimas mokymo procese yra naudingas ir reikalingas. Tai skatina mokinių susidomėjimą, kelia jų motyvaciją, tuo pačiu pajvairina mokymosi procesą.

6.2.4 Tyrimo rezultatai

Pabaigus tyrimą mokytojams ir mokiniams buvo pateikta anketos (61 psl. 3 Priedas).

Mokinių ir mokytojų atsiliepimai apie kompiuterinės priemonės reikalingumą ir naudingumą pateikti lentelėje (žr. 6.4 Lentelė):

6.4 Lentelė Atsiliepimų apie programą lentelė

Kriterijus	Reikalingumas	Naudingumas
Teigiamas	110	108
Neigiamas	-	-
Neturi nuomonės	6	8
Įvertinimas proc.	94,8%	93.1%

Visi užpildę anketas pabrėžė, kad tokia kompiuterinė priemonė yra reikalinga ir naudinga.

Išnagrinėjus kitus mokiniams ir mokytojams pateiktus anketos klausimus galima padaryti tokias išvadas:

- Mokiniai pageidautų, kad dalis mokomosios medžiagos ir paaiškinimų būtų pateikta garso ir vaizdo intarpais. Tai pat buvo išreikštas noras matyti kitokią mokomosios priemonės aplinką.

- Įtraukti daugiau teorinių temų.

- Beveik visiems patiko tai, kad rado užduočių, kurios nebuvo jiems įprastos ir matytos. Teko išbandyti patiems sudėlioti programėlę iš „nešiojamų elementų“, programėlėje įrašyti teisingus pasirinkimus ar teiginius. Todėl visi pageidavo, kad naujoje kompiuterinėje priemonėje būtų daugiau tokio tipo užduočių ir testų, nei įprastų, kurių metu tik pasirenkami teisingi atsakymai.

- Vyresnių klasių moksleiviai pritarė, kad tokia priemonė yra reikalinga. Nes tai būdas vienoje vietoje surasti ir teorinę medžiagą ir tuo pačių įsivertinti savo žinias testų pagalba. Buvo išreikštas pageidavimas papildyti naujomis teorinėmis temomis, kurios įeina į valstybinio egzamino programą: funkcijomis ir procedūromis, masyviais, įrašais, tekstinėmis bylomis. Pateikti daugiau interaktyvių užduočių, testų.

- Informacinių technologijų mokytojai pageidautų daugiau mokomosios medžiagos, daugiau akcentuoti tam tikras temas. Pateikti daugiau praktinio pobūdžio pavyzdžių ir užduočių, sudaryti atskirą aplanką joms kaupti ir nuolat pildyti.

Nurodyti pageidavimai įtraukti į mokomosios priemonės tobulinimo planą.

6.3 Galimi priemonės patobulinimai

Mokomosios priemonės tiek kūrimo, tiek testavimo metu kilo daug idėjų. Todėl yra numatyti šie programos tobulinimo etapai:

- Papildyti mokomąją kompiuterinę priemonę testais, su vartotojo registracija ir rezultatų išsaugojimu
- Papildyti mokomąją priemonę naujomis temomis
- Papildyti kiekvieną temą savikontrolės klausimais, uždaviniais savarankiškam darbui
- Mokomąją medžiagą ir užduotis įkelti ir tekstiniu formatu, kad būtų galimybę ją atsispausdinti.

7 Išvados

- Mokyklose atliktas tyrimas parodantis mokytojų ir moksleivių nuomonę apie programavimui skirtų mokomųjų kompiuterinių priemonių panaudojimą. Tyrimas rodo, kad egzistuojančios priemonės yra pernelyg pasenusios, daugiau skirtos tik pažinčiai su programavimu, neatitinka ugdymo programos reikalavimų.
- Atlikus moksleivių ir mokytojų apklausą sudarytos elektroninių mokymo priemonių kriterijų sąrašas. Jo pagrindu atliktas programų, skirtų programavimui mokyti, tyrimas.
- Įvertinus šias ypatybes buvo numatyta sukurti priemonę skirtą programavimui mokyti, kuri apimtų pagrindinės mokyklos ugdymo planą, tai pat būtų skirta moksleiviams pasirinkusiems programavimo modulį. Šia priemonę galės naudotis visi besimokantieji programavimo pradmenų.
- Parengta priemonė apima Pascal kalbos pagrindines valdymo struktūras: priskyrimo ir sąlygos sakinius, ciklus.
- Mokymo medžiaga pateikta HTML formate, animuoti pavyzdžiai sukurti panaudojant programą Wink2000. Savikontrolės ir kontroliniai testai sukurti su Jawa skriptais, interaktyvių užduočių kūrimo programa „Hot Potatoes“. Pratimams panaudota testavimo programa TestTool.
- Mokomoji medžiaga programoje pateikta aiškiai ir suprantamai, programa patogiu naudotis.
- Žinių patikrinimo sistema įgyvendinta tik dalinai. Reikalinga universali, nuolat besikeičianti žinių testavimo programa, kurioje būtų galima lengvai keisti ir apdoroti duomenis. Testai turi būti kuriami atsižvelgiant į moksleivių motyvacijos ir žinių lygį. Šie aspektai įtraukti į programos tobulinimo planą.

8 Literatūros sąrašas

1. Bendrosios programos ir išsilavinimo standartai. – V., 2003
2. Blonskis J, Baniulis K, Jusas V, Marcinkevičius R, Smolenskas J. Programavimas. –K.: Technologija, 1999. – 338 p.
3. Blonskis J, Dagienė V. Programavimo pradmenys: vadovėlis XI-XII kl. – V.: TEV, 2001. – 270 p.
4. Informatikos laboratoriniai darbai.[žiūrėta 2007 12. 15]. Prieiga per internetą: http://proin.ktu.lt/I2_papild/Laboratoriniai_darbai.htm
5. Lipeikienė J. Švietimo reforma ir mokytojų rengimas: aktualijos ir perspektyvos. 2004.[žiūrėta 2008 04 22]. Prieiga per internetą: <http://www.ceeol.com/aspx/getdocument.aspx?logid=5&id=e01052ad-d6e8-49df-b965-8201a4ff725c>
6. Kurasova O. Paskaitų ciklo „Mokomųjų programų kūrimas“ , skirto Pedagogų kvalifikacijos centro informatikos specialybės studentams konspektas. 2004.
7. TestTool4 sistemos vartotojo vadovas. 2006. Prieiga per internetą. http://pilis.pit.ktu.lt/tt/dokuwiki/doku.php?id=tutorial_lt
8. logo.lt. [žiūrėta 2007 10 16]. Prieiga per internetą: www.logo.lt
9. http://www.info.lt/intelektika/index.php?lng=lt&content=pages&page_id=22 [žiūrėta 2008 04 15]
10. www.mkc.lt/dokumentas/mokymosi_medziaga/pazinkime_virtualias_mokymosi_aplinkas.doc [žiūrėta 2008 04 26]
11. www.lt.wikipedia.org (žiūrėta. 2006-10-12).
12. <http://testtool.ktu.lt/> (žiūrėta 2008 04 15)

9 Priedai

1 PRIEDAS Anketa moksleiviams

Gerb. Moksleivi,

KTU Informatikos fakulteto kompiuterizuoto mokymo magistrantė Rima Čeplikienė atlieka tyrimą, kurio tikslas – išsiaiškinti kokios mokomosios kompiuterinės priemonės naudojamos mokant programavimo bei su kokiais problemomis susiduriate mokymo procese.

Perskaitykite kiekvieną klausimą ir pažymėkite tinkamą atsakymo variantą arba parašykite savo komentarą.

Jūsų pateikti duomenys bus naudojami tik apibendrinti, todėl anonimiškumas garantuojamas.

Anketa moksleiviams

1. Ar Jūsų mokykloje dėstomas „Programavimo pradmenų“ modulis
 - a) Taip
 - b) Ne
2. Ar domitės kitomis programavimo kalbomis be Pascal? (kokiomis?)
 - a) Taip
 - b) Ne

3. Ar mokotės neakivaizdinėje JPM?
 - a) Taip
 - b) Ne
4. Kokiomis priemonėmis naudojotės mokydami?
 - a) J.Blonskio ir V.Dagienės vadovėliu „Programavimo pradmenys“
 - b) Vadovėlis „Kompiuterija“
 - c) J.Blonskis, V.Bukšnaitis „Programavimas“
 - d) Kitomis _____
5. Kuriais kompiuterinėmis priemonėmis iš čia išvardintų Jūs naudojotės ar bent žinote jas?
 - a) Programavimo pradmenys
 - b) Elektroninis informatikos vadovėlis
 - c) PaskMok
 - d) Informatika
 - e) Kitomis _____
6. Su kokiais problemomis susiduriate naudodami MKP?
 - a) Neįdomios ir pasenę
 - b) Neatitinka to lygio, reikalingo pasiruošti valstybiniam egzaminui
 - c) Vadovėliuose randam daugiau teorijos ir pavyzdžių
7. Kokios temos Jums yra sunkiausios (išrinkite 1 sunkiausią)?
 - a) Ciklai
 - b) Masyvai
 - c) Įrašai
 - d) Procedūros ir funkcijos
 - e) Duomenys failuose
8. Ar programavimo mokymasis Jums bus naudingas pasirenkant studijas?
 - a) Taip
 - b) Ne
 - c) Nežinau

*Nuoširdžiai dėkoju už atsakymus
Parengė KTU kompiuterizuoto mokymo magistrantė Rima Čeplikienė.*

2 PRIEDAS Anketa mokytojams

Gerb. Mokytojai,

KTU Informatikos fakulteto kompiuterizuoto mokymo magistrantė Rima Čeplikienė atlieka tyrimą, kurio tikslas – išsiaiškinti kokios mokomosios kompiuterinės priemonės naudojamos mokant programavimo, jų efektyvumas. Perskaitykite kiekvieną klausimą ir pažymėkite tinkamą atsakymo variantą arba parašykite savo komentarą. Jūsų pateikti duomenys bus naudojami tik apibendrinti, todėl anonimiškumas garantuojamas.

Anketa mokytojams

1. Kokias kompiuterines programas žinote skirtas mokytis programavimo?
 - a)
 - b)
 - c)
 - d)
2. Ar naudojate MKP mokant programavimo?
 - a) Taip
 - b) Ne
 - c) Retkarčiais
3. Kokia Jūsų nuomonė apie MKP panaudojimą mokant programavimo
 - a) Žiūriu į tai teigiamai, tai puiki priemonė pajvairinti ugdymo procesą
 - b) Kartais tai galima taikyti
 - c) Neverta gaišti laiko
 - d) Neturėjau galimybės išbandyti tai su kokia priemone
4. Kokiomis priemonėmis iš čia paminėtų Jūs naudojate?
 - a) Elektroninis programavimo vadovėlis
 - b) PaskMok
 - c) Informatika
 - d) Programavimo pradmenys
5. Kam naudojate programavimui skirtas MKP?
 - a) Žinių patikrinimui
 - b) Teorinės medžiagos perteikimui
 - c) Pasiūlau susipažinti su jomis tik besidominčius programavimu moksleivius
6. Su kokiomis problemomis susiduriate naudojantis MKP?
 - a) Programos pasenę, neatitinka mokymosi turinio
 - b) Nėra interaktyvios, vaizdžios
 - c) Kai kurios jų skirtos DOS aplinkai, todėl trikdo kompiuterio darbą
7. Ar pakankamas MKP pasirinkimas?
 - a) Taip
 - b) Ne
 - c) Neturiu nuomonės
8. Ar mokinių apklausai vartojate testus?
 - a) Dažnai
 - b) Retai
 - c) Niekada
9. Kokius kriterijus turėtų atitikti testai?
 - a)
 - b)
 - c)
 - d)

*Nuoširdžiai dėkoju už atsakymus
Parengė KTU kompiuterizuoto mokymo magistrantė Rima Čeplikienė*

Anketa moksleiviams ir mokytojams

1. Ar reikalinga mokomoji programa apie Pascal programavimo kalbą?

- a. Taip
- b. Ne
- c. Neturiu nuomonės

2. Ar ji buvo Jums naudinga?

- a. Taip
- b. Ne
- c. Neturiu nuomonės

3. Kas Jums patiko pateiktoje mokomojoje programoje?

- a. _____
- b. _____
- c. _____
- d. _____

4. Kas Jums pateiktoje programoje nepriimtina?

- a. _____
- b. _____
- c. _____
- d. _____

5. Kokias klaidas radote?

- a. _____
- b. _____
- c. _____

6. Ką reikėtų pakeisti šioje programoje?

- a. _____
- b. _____
- c. _____
- d. _____

7. Kuo reikėtų papildyti šią mokomąją programą?

- a. _____
- b. _____
- c. _____

4 PRIEDAS Kontrolinis testas temai „Priskyrimo sakiny“

Kontrolinis testas

Pavardė:

Vardas:

1. Kokia bus kintamojo x reikšmė atlikus sakinių seką

```
x := a;  
x := x - b;  
x := c * x - x;
```

Jei $a = 12$, $b = 16$, $c = 7$?

- 24
- 12
- 4
- 15

2. Kokia bus kintamojo x reikšmė atlikus sakinių seką

```
x := a;  
x := x mod c;  
x := c * x div b + x;
```

Jei $a = 33$, $b = 7$, $c = 6$?

- 5
- 35
- 12
- 24

3. Kokia bus kintamojo reikšmė atlikus sakinių seką

```
x := a mod c + b div 3;  
b := x + 3 * b div a;  
x := 3 * b mod (a + c);  
x := (x + a + b) mod 10;
```

Jei $a = 7$, $b = 8$, $c = 5$?

- 5
- 3
- 9
- 4

4. Kam lygi kintamojų x ir y reikšmė atlikus sakinių seką

```
x := a;  
y := x + b;  
x := y mod x;  
y := x * y mod 10;
```

Jei $a = 4$, $b = 6$?

- $x = 0, y = 4$
- $x = 0, y = 2$
- $x = 2, y = 0$
- $x = 2, y = 10$

5. Kam lygios kintamųjų x ir y reikšmės atlikus sakinių seką

```
y := a mod b;  
x := (b + 13) mod a;  
y := x - y;  
x := x + y - x * y
```

Jei $a = 8$, $b = 11$?

- $x = 8, y = 8$
- $x = -4, y = 4$
- $x = -8, y = 8$
- $x = -8, y = -8$

6. Kam lygios kintamųjų x ir y reikšmės atlikus sakinių seką

```
y := b;  
x := y + a;  
y := x mod (y + 1) + y + 1;  
x := x div y - x mod y;  
y := x * y - 10;
```

Jei $a = 2$, $b = 9$?

- $x = 2, y = 1$
- $x = 8, y = 5$
- $x = 1, y = 1$
- $x = -1, y = -1$

7. Kam lygios kintamųjų x ir y reikšmės atlikus sakinių seką

```
x := (a + b) mod 10;  
y := x + a;  
a := x * y div 5;  
b := (x + y) mod 5;  
y := x - a + b;
```

Jei $a = 2$, $b = 9$?

- $x = 1, y = 1$
- $x = 1, y = 3$
- $x = 3, y = 1$
- $x = 3, y = 3$

Patikrinti

5 PRIEDAS Testas temai „Priskyrimo sakiny“

Testas

1. Ar yra netaisyklingų priskyrimo sakinių? Nurodykite juos.

- $xx := x * x;$
- $x := 2 \bmod x;$
- $2 := x - 2;$
- $x \bmod 2 := x * x.$

2. Kokia bus kintamojo z reikšmė atlikus šių sakinių seką?

```
z := 3;  
z := z * z;  
z := z + z;  
z := z - 1;
```

- 17
- 18
- 16
- 15

3. Kokia bus kintamojo x reikšmė atlikus sakinių seką?

```
x := a;  
x := x mod c;  
x := c * x div b + x;
```

Jei $a = 33$, $b = 7$, $c = 6$?

- 7
- 5
- 10
- 6

4. Kokias reikšmes įgis kintamieji n ir s, kompiuteriui įvykdžius šiuos programos sakinius?

```
a := -1; b := 1; c := 3;  
s := 0; n := 0;  
if a > 0 then begin  
    s := s + a;  
    n := n + 1  
end;  
if b > 0 then begin  
    s := s + b;  
    n := n + 1  
end;  
if c > 0 then begin  
    s := s + c;  
    n := n + 1  
end;
```

- $s=2, n=2.$
- $s=2, n=4$
- $s=-1, n=1$
- $s=4, n=2.$

5. Atlikus sakinį

```
if a > b then a := a - b  
else b := b - a
```

gautos tokios kintamųjų reikšmės $a=10$ ir $b=4$. Kokios buvo pradinės kintamųjų a ir b reikšmės?

- $a = 14; b = -10$
- $a = 18; b = 8.$
- $a = 14; b = -6$
- $a = 14 ; b = 4 .$

Įvertinimas

Išvalyti

6 PRIEDAS Testas sukurtas su „Hot Potatoes“ programa

Mokomasis testas „Priskyrimo sakinyš“	
Testas	
14:49	
Šis testas padės Jums geriau suprasti priskyrimo sakini	
Parodyti visus klausimus	
1 / 9 Toliau	
3. Atlikus sakinių seką c:=c div 4 c:=c - 2 c:=c + 4 gauta kintamojo c reikšmė 8. Kokia buvo pradinė c reikšmė?	
a. <input type="checkbox"/> 24 b. <input type="checkbox"/> 26 c. <input type="checkbox"/> 23 d. <input type="checkbox"/> 25 e. <input type="checkbox"/> 27	
Parodyti visus klausimus	
Atgal 2 / 9 Toliau	
Duota programa.	
<pre>program tr; var tr, s1, s2, s3, s: integer; begin Readln(x); s1 := x div 100; s2 := x mod 10 div 10; s3 := x mod 10; s := s1 + s2 + s3; writeln(s1); writeln(s2); writeln(s3); writeln(s); end.</pre>	
Ką atspausdins ši programa, jei x = 156? (rašydami skaičius palikite tarp jų tarpus)	
<input type="text"/>	
Atsakymas	
Parodyti visus klausimus	
Atgal 3 / 9 Toliau	
Kokia bus kintamojo n reikšmė atlikus sakinių seką? n := 6; n := 1 + n * n + 2;	
A. <input type="text"/> 44 B. <input type="text"/> 56 C. <input type="text"/> 39 D. <input type="text"/> 38	

Parodyti visus klausimus

Atgal

4 / 9

Toliau

4. Ką gausite, atlikę šią sakinių seką:

$a := 28$

$a := a \text{ div } 5$

$a := a + 2$

$a := a * 7$

Atsakymas

Parodyti visus klausimus

Atgal

5 / 9

Toliau

Kokia yra priskyrimo sakinio struktūra?

A. kintamasis := kintamieji

B. reiškinyks := kintamasis

C. kintamasis := reiškinyks

Parodyti visus klausimus

Atgal

6 / 9

Toliau

Pažymėkite teisingus priskyrimo sakinius

a. $sk := 25$

b. $x := x + 1$

c. $ats = x + y$

d. $x + 1 := x$

e. $x = x + 1$

f. $x + y = suma$

g. $x + y := suma$

Atsakymas

Parodyti visus klausimus

Atgal

7 / 9

Toliau

Kas yra priskyrimo sakinyks?

A. Tai, ką atlieka kompiuteris

B. Žymenų ir taisyklių visuma

C. Tai tokia konstrukcija, nurodanti atlikti kokius nors veiksmus

Parodyti visus klausimus

Atgal

8 / 9

Toliau

Ką gausite atlikę šią sakinių seką:

`x:=19`

`x:=x div 5`

`x:=x *3`

`x:=x div 6`

A. 4

B. 1

C. 2

D. 3

Parodyti visus klausimus

Atgal

9 / 9

`x := a;`

`x := x - b;`

`x := c * x*x;`

Kokia kintamojo x reikšmė atlikus šiuos priskyrimo sakinius, jei:

a = 5; b = 6, c = 3.

Atsakymas

7 PRIEDAS Testo sukurto formų pagalba HTML kodas

```
62 <span lang="en-us">&nbsp;</span></font>1.  
63 <span lang="en-us">Ar </span>yra netaisyklingų priskyrimo sakinių? Nurodykite  
64 juos.</b></p>  
65 <p style="margin-top: 0; margin-bottom: 0; margin-left:75px">  
66 <font color="#000080">  
67 <input type="checkbox" name="atsl_1" value="ON"></font><span lang="en-us">xx := x * x;</span><br>  
68 <font color="#000080">  
69 <input type="checkbox" name="atsl_2" value="ON"></font><span lang="en-us">x := 2 mod x;</span><br>  
70 <font color="#000080">  
71 <input type="checkbox" name="atsl_3" value="ON"></font><span lang="en-us">2 := x&nbsp;  <br>  
72 - 2;</span>.<br>  
73 <font color="#000080">  
74 <input type="checkbox" name="atsl_4" value="ON"></font><span lang="en-us">x mod 2 := x  
75 * x</span>.</p>  
76 <hr>  
77 <p style="margin-top: 0; margin-bottom: 0; margin-left:75px">  
78 &nbsp;</p>  
79 <p style="margin-top: 0; margin-bottom: 0; margin-left:40px">  
80 <b>2.&br/>81 <span lang="en-us">Kokia bus kintamojo z reikšmė atlikus šią sakinių seka</span>?</b></p>  
82 <p style="margin-top: 0; margin-bottom: 0; margin-left:140px">  
83 <b><font color="#008080" face="Courier New">z := 3;</font></b></p>  
84 <p style="margin-top: 0; margin-bottom: 0; margin-left:140px">  
85 <span lang="en-us"><font face="Courier New" color="#008080"><b>z := z * z;</b></font></span></p>  
86 <p style="margin-top: 0; margin-bottom: 0; margin-left:140px">  
87 <span lang="en-us"><font face="Courier New" color="#008080"><b>z := z + z;</b></font></span></p>  
88 <p style="margin-top: 0; margin-bottom: 0; margin-left:140px">  
89 <span lang="en-us"><font face="Courier New" color="#008080"><b>z := z - 1;</b></font></span></p>  
90 <p style="margin-top: 0; margin-bottom: 0; margin-left:75px">  
91 <font color="#000080">  
92 <input type="radio" name="ats2"> <span lang="en-us">17</span><font color="#000080"><br>  
93 <input type="radio" name="ats2"><span lang="en-us">18</span><font color="#000080"><br>  
94 <input type="radio" name="ats2"><span lang="en-us">15</span><font color="#000080"><br>  
95 <input type="radio" name="ats2"><span lang="en-us">16</span><font color="#000080"><br>
```

Alert() metodo aprašymas Html kalboje

```
51 alert(pav+" "+vard+"\n1 užd. balas: "+balas1+"\n2 užd. balas: "+balas2+"\n3 užd. balas: "+balas3+"\n4 užd. bal  
52 +balas4+"\n5 užd. balas: "+balas5+  
53 "\n6 užd. balas: "+balas7+  
54 "\n7 užd. balas: "+balas9+  
55 "\n Pazymys: "+pazym);  
56 )
```

Rezultatų langas atlikus testą



8 PRIEDAS Kontrolinis testas temai „Sąlygos sakiny“

Kontrolinis testas

Pavardė:

Vardas:

1. Kokia bus kintamojo x reikšmė atlikus sakinių seką

```
if b mod 2 = 0 then x := a * b
                    else x := a + b;
x := a * x - b;
```

Jei $a = 2, b = 4$?

- 12
- 0
- 6
- 2

2. Kokios bus kintamųjų a ir b reikšmės, atlikus šių sakinių seką?

```
a := 10; b := 6;
if a > b then a := a - b
            else a := a + 3;
```

- $a = 4, b = 6$
- $a = 6, b = 4$
- $a = 4, b = 7$
- $a = 7, b = 6$

3. Atlikus sakinį

```
if a > b then a := a - b
            else b := b - a;
```

gautos šitokios a ir b reikšmės: $a = 5, b = 5$. Kokios galėjo būti šių kintamųjų reikšmės prieš atliekant priskyrimo sakinį?

- $a = 10, b = 10$;
- $a = 5, b = 10$
- $a = 10, b = 15$;

4. Nustatykite, kokia bus kintamojo x reikšmė atlikus sakinių seką

```
x := -7;
if x < 0 then begin
    x := -x;
    x := x mod 3;
end;
```

- 2
- 1
- 1
- 2

5. Nustatykite kintamojo x reikšmę atlikus sakinių seką?

```
x := 7;
if x < 0 then x := -x
            else begin
                if x < 10 then x := x * 10;
                if x < 100 then x := x * 100;
            end;
```

- $x = -7$
- $x = 70$
- $x = 7$
- $x = 700$

6. Kam lygios kintamųjų x ir y reikšmės atlikus sakinių seką

```
x := 20;  
y := 10;  
if x < y then y := y - x  
    else x := x - y;  
if x <= y then x := x - 5;  
if x >= y then y := y + 5;
```

- x = -15, y = 10
- x = 15, y = 5
- x = 5, y = 15
- x = 20, y = -10

7. Kokia bus kintamojo rez reikšmė atlikus šias sakinių sekas?

```
sk := 5;  
if sk mod 3 = 0 then rez := sk * 5  
    else if sk > 0 then begin  
        sk := sk * 5;  
        rez := sk * 3  
    end  
    else rez := sk * 2
```

- rez = 25
- rez = 10
- rez = 75

Patikrinti

9 PRIEDAS Savikontrolės testas temai „Ciklai“

Savikontrolės testas

1. Kokio tipo gali būti ciklo FOR kintamasis?

- tik realaus (Real)
- tik sveikojo (Integer)
- Realaus (Real) ir sveikojo (Integer)

2. Kiek kartų bus atliekamas šis ciklas

```
for i := 0 to 15 do
```

- 15
- 16

3. Kokia a reikšmė įvykdžius ciklą?

```
n := 5;  
a := 2;  
for i := 3 to n do  
a := a + 1;
```

4. Kiek kartų bus atliekamas sakiny $a := a + k$?

```
a := 0;  
k := 10;  
x := 27;  
for z := k to x do  
a := a + k;
```

- 17
- 18
- 19

5. Duota sakinių seka

```
p := 1;  
i := 0;  
while i < n do  
begin  
p := p * a;  
i := i + 1;  
end
```

Nustatykite, koks bus rezultatas p, jei pradiniai duomenys yra: $a = 7, n = 3$?

- 343
- 7
- 49

6. Duoti 2 ciklai. Ar sutaps jų rezultatai?

<pre>while a <= b do begin b := b - 2; a := a + 1 end;</pre>	<pre>for k := a to b do b := b - 2</pre>
---	--

- Ne
 Taip

7. Nustatykite, kokios bus kintamųjų x ir y reikšmės atlikus sakinių seką:

```
x := 0;
y := 1;
while x <= 13 do
x := x + 3;
y := y + 3;
```

- x= 15, y= 13
 x= 15, y= 4

8. Nurodykite, ką atspausdins kompiuteris atlikęs šią sakinių seką:

```
read(a, b);
if a < b then
for k := a to b do
writeln(a + k);
```

Pradiniai duomenys: 5, 7

- 6 7 8
 10 11 12

9. Kiek kartų bus atliekamas ciklas?

```
n := 10;
x := 56;
while n < x do
x := x - n;
```

- 6
 5
 7

10. Kiek kartų atliekamas ciklas?

```
n := 4;
x := 29;
while x >= n do
begin
  n := n + 2;
  x := x - 3;
end;
```

- 6
 5
 7

Tikrinti

Išvalyti

10 PRIEDAS Kontrolinis testas temai „Ciklai“

Kontrolinis testas

Pavardė:

Vardas:

I. Duota programa

```
program uzd1;
var n, sum, i : integer;
begin
  readln(n);
  sum := 0;
  i := 1;
  while i <= n do
  begin
    sum := sum + i;
    i := i + 1;
  end;
  writeln ('sum = ', sum);
  readln
end.
```

Kam lygi *sum* reikšmė:

1. jei $n = 2$

- 3
- 2
- 0
- 1

2. jei $n = 4$

- 10
- 11
- 7
- 5

II. DUOTA PROGRAMA

```
program sklnr4;
var m, s : integer;
begin
  readln(m);
  s := 0;
  while m > 0 do
  begin
    s := s + m;
    m := m - 1;
  end;
  writeln('s = ', s);
  readln
end.
```

Kam lygi s reikšmė

1. jei $m = 3$

- 8
- 6
- 4

2. jei $m = 5$

- 5
- 10
- 15
- 0

III. Duota programa

```
program sklnr5;  
  var n, a, i, d : integer;  
begin  
  readln(n, a);  
  i := 1; d := a;  
  while i <= n do  
  begin  
    a := a + d;  
    i := i + 1  
  end;  
  writeln('a = ', a);  
  readln  
end.
```

Kam lygi a reikšmė, kai $n = 9$, $a = 3$?

- $x = 35$
- $x = 20$
- $x = 15$
- $x = 30$

Patikrinti

11 PRIEDAS Interaktyvi užduotis temai „Ciklai“

Parašyta programa skaičiuojanti intervale [a, b] lyginių skaičių sumą ir nelyginių skaičių sandaugą. Įrašykite praleistas komandas.

	0	1	2	a	b	i	integer	sand	sum
<pre> Program sum_sand; var a, b, sum, sand, i: <input type="text"/>; begin WriteLn('Iveskite intervalo pradzia'); ReadLn(<input type="text"/>); WriteLn('Iveskite intervalo pabaiga'); ReadLn(<input type="text"/>); sum := <input type="text"/>; sand := <input type="text"/>; for i := <input type="text"/> to <input type="text"/> do if <input type="text"/> mod <input type="text"/> = 0 then sum := <input type="text"/> + i else sand := <input type="text"/> * <input type="text"/>; WriteLn(' Suma ', <input type="text"/>); Writeln(' Sandauga ', <input type="text"/>); End. </pre>									
<input type="text" value="Atsakymas"/>									

Saunu, Jūs atsakėte teisingai
Jūs atsakėte į 100%.

	0	1	2	a	b	i	integer	sand	sum
<pre> Program sum_sand; var a, b, sum, sand, i: integer; begin WriteLn('Iveskite intervalo pradzia'); ReadLn(a); WriteLn('Iveskite intervalo pabaiga'); ReadLn(b); sum := 0; sand := 1; for i := a to b do if i mod 2 = 0 then sum := sum + i else sand := sand * i; WriteLn(' Suma ', sum); Writeln(' Sandauga ', sand); End. </pre>									
<input type="text" value="Atsakymas"/>									

12 PRIEDAS Interaktyvi užduotis temai „Ciklai“

Duota seka, kurios pabaigos simbolis yra nulis. Sudėliokite programą, kuri suskaičiuotų sekos narių sumą ir suskaičiuotų iš kiek narių sudaryta seka.

Sudėliokite sakinius į vietas

Atsakymas		
.....	Program seka;	sum := 0;
Kintamųjų tipų aprašas		begin
.....		var n, sum, kiek: integer;
.....		kiek := kiek + 1;
Ciklo parengiamoji dalis		Writeln(' kiekis, kiek);
.....		Writeln('suma, sum);
Ciklo kartojimo sąlygos tikrinimas		while n <> 0 do
Darbinė ciklo dalis	begin	
.....		end;
.....		Readln(n);
Ciklo parametro reikšmės keitimas	Read(n);	sum := sum + n;
.....		
Sakiniai po ciklo		Writeln('Iveskite pirmą sekos narį')
.....		
.....	end.	kiek := 0;

Atsakymas		
.....	Program seka;	
Kintamųjų tipų aprašas	var n, sum, kiek: integer;	
.....	begin	
.....	Writeln('Iveskite pirmą sekos narį')	
.....	Readln(n);	
Ciklo parengiamoji dalis	sum := 0;	
.....	kiek := 0;	
Ciklo kartojimo sąlygos tikrinimas	while n <> 0 do	
Darbinė ciklo dalis	begin	
.....	sum := sum + n;	
.....	kiek := kiek + 1;	
Ciklo parametro reikšmės keitimas	Read(n);	
.....	end;	
Sakiniai po ciklo	Writeln('suma, sum);	
.....	Writeln(' kiekis, kiek);	
.....	end.	

13 PRIEDAS Programos langų pavyzdžiai

	
<p>Priemonė skirta mokytį programavimo Pascal programavimo kalbas. Ją galima panaudoti pamokose, tai pat programavimo mokymui išplėstiniam modulyje XI-XII klasėse, būrelių užsiėmimų metu ar savarankiškam mokymuisi.</p> <p>Mokomoji priemonė sudaryta iš 3 temų, kiekvienoje iš jų yra teorinės medžiagos pristatymas, pagrindinių akcentų demonstravimas, užduotys, savikontrolės ir kontroliniai testai.</p> <p>Sėkmės darbe</p> <p>ceplikiene@gmail.com</p>	<h3>Kaip mokytis naudojantis priemone?</h3> <p>Naudojant šią mokomąją priemonę pamokose siūlome temas nagrinėti ta pačia eilės tvarka kaip jos išdėstytos priemonėje. Teorinę medžiagą reiktų perskaityti, išmokyti arba pakartoti, je jau mokėtės šią temą.</p> <p>Užduotis nesistenkite atlikti visas per vieną pamoką. Geriau leiskite mokiniam išmėginti savo kūrybinius sugebėjimus.</p> <p>Kiekvienos temos teorinės medžiagos įsisavinimui patikrinti paruošti savikontrolės testai, savarankiško darbo užduotys. Testai skirti įtvirtinti ir patikrinti žinias po kiekvienos temos. Juos galite nagrinėti daug kartų.</p> <p>Sukurtoje mokymo priemonėje kiekvienos temos yra kontroliniai (atsiskaitymo) testai, kuriuos reiktų spręsti išmokus visą čia siūlomą medžiagą.</p> <p>Pateikta demonstracinė medžiaga geriau suprasti programavimo konstrukcijų veikimą.</p>

Testai ir užduotys

Priskyrimo sakiny's

Savikontrolės testai [Nr.1](#) ir [Nr.2](#)

[Kontrolinis testas](#)

Sąlygos sakiny's

[Testas](#)

Ciklai

[Savikontrolės testas](#)

[Kontrolinis testas](#)

Priskyrimo sakiny

Turinys

Demonstracija

Savikontrolės testas

Testas Nr.1

Testas Nr.2

Kontrolinis testas

Bendroji priskyrimo sakinio forma:

`Kintamasis := reiškinys;`

Juo "liepiama" kompiuteriui visu pirma apskaičiuoti išraiška, o po to gauta rezultatą priskirti kintamajam.

Priskyrimo sakiny - vienas dažniausių programos vykdomoje dalyje rašomų sakinių.

Jo paskirtis – suteikti reikšmę priskyrimo ženklo := kairėje pusėje esančiam kintamajam. Reiškiny gali būti kintamasis, konstanta, kintamųjų, konstantų, funkcijų, operacijų bei skliaustų junginys.

Pavyzdžiui:

```
a := x + b;  
b := 5;  
kampas := 15 + Sin(15*x);
```

Taisyklės

Priskyrimo sakinio dešinėsios ir kairėsios pusių kintamųjų tipai turi būti suderinti: tiek rezultatas tiek reiškinys turi būti to paties tipo;

Priskyrimo ženklo := viduryje negalima palikti tarpo;

Priskyrimo sakinio dešinėje pusėje esančios kintamųjų reikšmės turi būti žinomos.

Sąlygos sakiny

Turinys

Demonstracija

Testas

Sąlygos sakiny [galina keisti nuosekalia programos sakinių atlikimo tvarka, atsižvelgiant į tikrinamas sąlygas.

Sąlygos sakiny yra sudarytas iš sąlygos (kelių sąlygų), kuri gali būti tenkinama arba ne. Priklausomai nuo to bus vykdoma viena ar kita sakinių grupė. Sąlyga – tai loginis reiškinys: kai jo reikšmė yra **True**, tai sakoma, kad sąlyga tenkinama ir atliekamas pirmasis sakiny, kai **False**, sakome, kad sąlyga netenkinama ir tada atliekamas antrasis sakiny.

Paskalio kalba sąlyginis sakiny užrašomas taip:

```
if sąlyga then sakiny1  
else sakiny2;
```

Pagal paskalio kalbos taisykles po žodžių **then** ir **else** galima rašyti tik vieną sakiny. Jeigu prireikia kelių sakinių, tuomet juos sujungiame į vieną sudėtinį, pavartoję bazinius žodžius **begin** ir **end**.

```
if sąlyga then begin  
    sakinių grupė;  
end  
else begin  
    sakinių grupė;  
end;
```

Kartais reikia parinkti vieną iš kelių veiksmų. Pasirinkimas vyksta, atsižvelgiant į sąlygas. Tikrinti reikia viena

Ciklai

[Turinys](#)

[Ciklas FOR](#)

[Ciklas WHILE](#)

[Panaudojimo atvejai](#)

[Ypatumai ir taisyklės](#)

Demonstracija

Užduotis

[Užduotis Nr.1](#)

[Užduotis Nr.2](#)

Savikontrolės testas

Kontrolinis testas

Ciklas For

Ciklo **For** antraštė prasideda žodžiu **For**. Po jo rašomas kintamojo vardas, šis kintamasis vadinamas ciklo kintamuoju. Toliau nurodoma, su kuriomis kintamojo reikšmėmis turi būti atliekamas ciklas - užrašoma pradinė (cp) ir galutinė (cg) ciklo kintamojo reikšmė. Kitimo žingsnis gali būti tik **1** arba **-1**. Ciklo kintamojo reikšmė vienetu padidinama **automatiškai**. Ciklo viduje jos keisti nereikia ir negalima. Pradinė ir galutinė ciklo kintamojo reikšmė gali būti ne tik konstanta, bet ir kintamasis arba reiškinys (tipai turi būti vienodi).

```
For i :=cp to cg do  
sakiny;
```

i- ciklo kintamasis, kuriam priskiriama (**ne prilyginama!**) pirmoji reikšmė; be to ciklo kintamasis visada yra iš sveikųjų skaičių aibės skaičius, t.y. priklauso **Integer** tipui.

Jeigu $cg < cp$, tai ciklo sakiny visai nevykdomas ir i reikšmė lieka neapibrežta. Jei $cp < cg$, pirmą kartą vykdamas ciklo sakiny, i įgyja cp reikšmę ir atliekamas po žodelio **do** esantis sakiny (dažnai tai būna sudėtinis sakiny). Jei $cp < cg$, tai i reikšmė padidinama vienetu ir vėl atliekamas ciklo sakiny. Veiksmai kartojami tol, kol $i = cg$. Vadinasi, ciklas atliekamas $cg - cp + 1$ kartų.

Jei toliau **For** antraštėje pradinė reikšmė bus didesnė už galinę, ciklas **For** nebūna vykdomas nei

Ciklai

[Turinys](#)

[Ciklas FOR](#)

[Ciklas WHILE](#)

[Panaudojimo atvejai](#)

[Ypatumai ir taisyklės](#)

Demonstracija

Užduotis

[Užduotis Nr.1](#)

[Užduotis Nr.2](#)

Savikontrolės testas

Kontrolinis testas

Ciklas while

Kartojimo veiksmus taikliausiai aprašo ciklas **While**. Šio tipo ciklas vartojamas tada, kai nežinomas ciklo kartojimų skaičius.

Programoje ši konstrukcija būtų užrašoma taip:

```
While loginys reiškinys do  
sakiny;
```

Pirmoji eilutė – ciklo antraštė. Po jos (t.y. po žodžio **do**) einantys sakiny arba sakiniai kartojami tol, kol tenkinama ciklo antraštėje užrašyta sąlyga (kol loginio reiškinio reikšmė yra **True**).

Ar kiekvienas ciklas turi pabaigą? Tam, kad ciklas baigtųsi, reikia, kad jo antraštėje užrašyto loginio reiškinio reikšmė kada nors taptų **False**. Vadinasi, loginio reiškinio komponentus turi keisti sakiny. Priešingu atveju loginio reiškinio reikšmė išliks **True** ir ciklas niekada nesibaigs. Toks ciklas vadinamas amžintuoju.

Amžinojo ciklo pavyzdys:

Ciklai

[Turinys](#)

[Ciklas FOR](#) [Ciklas WHILE](#) [Panaudojimo atvejai](#) [Ypatumai ir taisyklės](#)

Demonstracija

Užduotis

[Užduotis Nr.1](#)

[Užduotis Nr.2](#)

Savikontrolės testas

Kontrolinis testas

Ciklų panaudojimo atvejai

- ▶ Ciklas **While** yra universalus, jis naudojamas kai nežinomas pasikartojimų skaičius, arba kai ciklo kintamojo reikšmė didėja ne kas 1.
- ▶ Ciklą **For** yra patogiau naudoti tada, kai ciklo kintamojo reikšmė didėja 1 arba -1 ir iš anksto žinoma kiek kartų bus atliekami pasikartojantys veiksmai
- ▶ Ciklas **For** netinka tada kai, ciklo kintamasis yra nesveikąjo tipo, t.y ne **integer** tipo.
- ▶ Jei naudojame ciklą ir ciklo kintamojo žingsnis ne 1 ar -1 tada tokiam uždaviniui reikia numatyti papildomus veiksmus cikle.

Ciklai

[Turinys](#)

[Ciklas FOR](#) [Ciklas WHILE](#) [Panaudojimo atvejai](#) [Ypatumai ir taisyklės](#)

Demonstracija

Užduotis

[Užduotis Nr.1](#)

[Užduotis Nr.2](#)

Savikontrolės testas

Kontrolinis testas

Ciklo konstrukcijos ypatumai ir taisyklės

- ▶ Ciklo sakiniai, kurie yra prieš ciklą vykdomi tik 1 kartą.
- ▶ Kiekvienas ciklas vykdo tik vieną sakinį, kuris yra po ciklo antraštę.
- ▶ Jei po ciklo antraštę yra keletas sakinių, kuriuos reikia vykdyti, tai jie apskliaudžiami tarp žodelių **Begin** ir **End**.
- ▶ Cikle kiekvienas sakinytis kartojamas tiek kartų, kiek vykdomas ciklas.
- ▶ Pasibaigus ciklui vykdomi sekantys sakiniai, tai gali būti tolimesni skaičiavimo veiksmai arba spausdinimas.

14 PRIEDAS Demonstracinis pavyzdys temai „Ciklai“

Ciklai

Turinys

[Ciklas FOR](#) [Ciklas WHILE](#) [Panaudojimo atvejai](#) [Ypatumai ir taisyklės](#)

Demonstracija

Užduotis

Užduotis Nr.1

Užduotis Nr.2


Savikontrolės testas

Kontrolinis testas

```

.....
S := 20;
While s =< 200 do
  s := s + 40;
.....
                    
```

Automobilis judės |
 priekį pakol ciklo
 sąlyga bus **TRUE**,
 priešingu atveju tapus
FALSE – automobilis
 sustos.



Ciklai

Turinys

[Ciklas FOR](#) [Ciklas WHILE](#) [Panaudojimo atvejai](#) [Ypatumai ir taisyklės](#)

Demonstracija

Užduotis

Užduotis Nr.1

Užduotis Nr.2

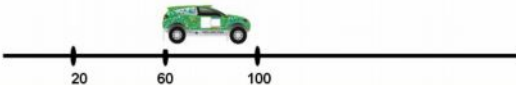
Savikontrolės testas

Kontrolinis testas

```

.....
S := 20;
While s =< 200 do
  s := s + 40;
.....
                    
```

TRUE



Ciklai

Turinys

[Ciklas FOR](#) [Ciklas WHILE](#) [Panaudojimo atvejai](#) [Ypatumai ir taisyklės](#)

Demonstracija

Užduotis

Užduotis Nr.1

Užduotis Nr.2

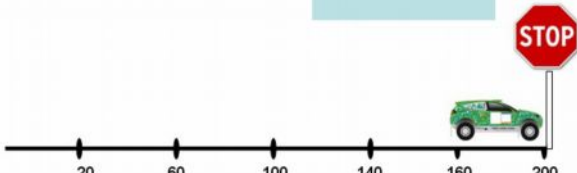
Savikontrolės testas

Kontrolinis testas

```

.....
S := 20;
While s =< 200 do
  s := s + 40;
.....
                    
```

FALSE



15 PRIEDAS Demonstracinis pavyzdys temai „Priskyrimo sakiny“

Priskyrimo sakiny

Turiny

Demonstracija

Savikontrolės testas

Testas Nr. 1

Testas Nr. 2

Kontrolinis testas

Užduotis: Duoti 3 sveikieji skaičiai. Programa skaičiuojanti jų sumą ir aritmetinį vidurkį.

Programos tekstas

```

Program sum_vid;
var a, b, c: integer;
    vid: real;

Begin
  Writeln('Įveskit 3 skaičius');
  Readln(a, b, c);
  Sum := a + b + c;
  Vid := sum / 3;
  Writeln('Rezultatai');
  Writeln(' Suma : ', sum);
  Writeln(' Vidurkis : ', vid);
  Readln
End
                    
```

Komentari

Apibrėžiama kad įvesti duomenys ir jų suma yra sveikojo tipo, o vidurkis **realiojo**

Operatyvioji atmintis

Integer				Real
a	b	c	sum	vid

Kompiuterio ekranas

Priskyrimo sakiny

Turiny

Demonstracija

Savikontrolės testas

Testas Nr. 1

Testas Nr. 2

Kontrolinis testas

Užduotis: Duoti 3 sveikieji skaičiai. Programa skaičiuojanti jų sumą ir aritmetinį vidurkį.

Programos tekstas

```

Program sum_vid;
var a, b, c: integer;
    vid: real;

Begin
  Writeln('Įveskit 3 skaičius');
  Readln(a, b, c);
  Sum := a + b + c;
  Vid := sum / 3;
  Writeln('Rezultatai');
  Writeln(' Suma : ', sum);
  Writeln(' Vidurkis : ', vid);
  Readln
End
                    
```

Komentari

Gautą sumą padalija iš 3 ir gautą rezultatą priskiriam kintamajam **Vid**

Operatyvioji atmintis

Integer				Real
a	b	c	sum	vid
5	12	25	42	14.0

Kompiuterio ekranas

Įveskite 3 skaičius
 5 12 25

Priskyrimo sakiny

Turiny

Demonstracija

Savikontrolės testas

Testas Nr. 1

Testas Nr. 2

Kontrolinis testas

Užduotis: Duoti 3 sveikieji skaičiai. Programa skaičiuojanti jų sumą ir aritmetinį vidurkį.

Programos tekstas

```

Program sum_vid;
var a, b, c: integer;
    vid: real;

Begin
  Writeln('Įveskit 3 skaičius');
  Readln(a, b, c);
  Sum := a + b + c;
  Vid := sum / 3;
  Writeln('Rezultatai');
  Writeln(' Suma : ', sum);
  Writeln(' Vidurkis : ', vid:1:1);
  Readln
End
                    
```

Komentari

Programos pabaiga

Operatyvioji atmintis

Integer				Real
a	b	c	sum	vid
5	12	25	42	14.0

Kompiuterio ekranas

Įveskite 3 skaičius
 5 12 25
 Rezultatai
 Suma: 42
 Vidurkis: 14,0

82

16 PRIEDAS Demonstracinis pavyzdys temai „Sąlygos sakiny“

Sąlygos sakiny

Tuiny

Demonstracija

Testas

```
A = 8;  
B = 5;  
If A > B then A := A - B  
    else B := A + B
```

Tikrinama sąlyga

Navigation arrows: left and right.

Sąlygos sakiny

Tuiny

Demonstracija

Testas

```
A = 15;  
B = 22;  
If A > B then A := A - B  
    else B := A + B
```

15 > 22
FALSE

Navigation arrows: left and right.

17 PRIEDAS Testų pavyzdžiai, sukurtų su TestTool

The screenshot shows the TestTool interface for a programming problem. At the top, there are buttons for 'Evaluate', 'Refresh', 'True', and 'Close preview'. The main heading reads: 'Duota algoritmo pradžia. Sukonstruokite jį taip, kad skaičiuotų intervalo [a, b] lyginių skaičių sandaugą. (Intervalo pradžia a - lyginis skaičius)'. Below this, there are two code editors. The left editor contains the following code:

```
program sand;  
var a, b, s:integer;  
  
Begin  
  writeln( " intervalo režiai ");  
  readln(a, b);  
  
  -  
  -  
  -  
  -  
  -  
  -  
  -  
End.
```

 The right editor contains the following code:

```
while a >= b do  
while a <= b do  
  s := 1;  
  s := 0;  
begin  
  s := s * a;  
  a := a + 2;  
  s := s + a;  
  a := a + 1;  
  writeln( " sandauga ", b);  
  writeln( "sandauga ", s);  
  b := b + 1;  
end;
```

 At the bottom right, there is an empty text input box and the text 'Nereikalingi objektai'.

The screenshot shows the TestTool interface for a programming problem. At the top, there are buttons for 'Evaluate', 'Refresh', 'True', and 'Close preview'. The main heading reads: 'Duota algoritmo pradžia. Sukonstruokite jį taip, kad skaičiuotų intervalo [a, b] lyginių skaičių sumą. (Intervalo pradžia a - lyginis skaičius)'. Below this, there are two code editors. The left editor contains the following code:

```
Program suma;  
var a, b, s:integer;  
  
Begin  
  writeln( " intervalo režiai");  
  readln(a, b);  
  
  -  
  -  
  -  
  -  
  -  
  -  
  -  
End.
```

 The right editor contains the following code:

```
while a >= b do  
while a <= b do  
  s := 1;  
  s := 0;  
begin  
  s := s * a;  
  a := a + 2;  
  s := s + a;  
  a := a + 1;  
  writeln( "suma ", b);  
  writeln( "suma ", s);  
  b := b + 1;  
end;
```

 At the bottom right, there is an empty text input box and the text 'Nereikalingi objektai'.

Evaluate Refresh True Close preview

Duota algoritmo pradžia. Sukonstruokite jį taip, kad skaičiuotų intervalo [a, b] nelyginių skaičių sumą.
(Intervalo pradžia a - nelyginis skaičius)

```

Program suma;
var a, b, s: integer;

Begin
  writeln( " intervalo režiai");
  readln(a, b);
  -
  -
  -
  -
  -
  -
  -
  End.

```

```

while a >= b do
while a <= b do
  s := 1;
  s := 0;
begin
  s := s * a;
  a := a + 2;
  s := s + a;
  a := a + 1;
  writeln( "suma ", b);
  writeln( "suma ", s);
  b := b + 1;
end;

```

Nereikalingi objektai

Evaluate Refresh True Close preview

Duotas programos fragmentas. Sukonstruokite jį taip, kad suskaičiuotų intervalo [a, b] narių sandaugą.

```

program sand;
var a, b, s, i: integer;
begin
  writeln( "Intervalo režiai:");
  readln(a, b);
  .
  .
  .
  .
  .
end.

```

```

i := i + 1;

s := 1;
s := s + i;

for i:=b do a do
for i:=1 to b do
for i:=a to b do

s := s + a;
s := s * i;
s := 0;
writeln(" sandauga ", b );
writeln( " sandauga ", s);

```

Nereikalingi objektai

Duota algoritmo pradžia. Sukonstruokite jį tai, kad skaičiuotų intervalo [a, b] skaičių sumą (s) ir sandaugą (ss).

<pre> program sum_sand; var a, b, i, s, ss: integer; begin writeln(" Intervalo režiai :); readln(a, b); - - - - - - - - - - end. </pre>	<pre> writeln(" suma ", s); writeln(" sandauga ", ss); begin for i := 1 to b do end; ss := 1; s := 0; s := s + i; ss := ss * i; s := s + a; ss := ss * a; for i := a to b do </pre> <div style="border: 1px solid black; width: 100px; height: 15px; margin: 5px auto;"></div> <p style="text-align: center; font-size: small;">Nereikalingi objektai</p>
---	--

Duota algoritmo pradžia. Sukonstruokite jį taip, kad skaičiuotų intervalo [a, b] skaičių sandaugą

<pre> Program suma; var a, b, s:integer; Begin writeln(" intervalo rezial"); readln(a, b); - - - - - - - - - - End. </pre>	<pre> s := 1; while a >= b do s := s + a; s := s * a; while a <= b do s := 0; begin a := a + 1; writeln(" sandauga ", b); b := b + 1; end; writeln("sandauga ", s); </pre> <div style="border: 1px solid black; width: 100px; height: 15px; margin: 5px auto;"></div> <p style="text-align: center; font-size: small;">Nereikalingi objektai</p>
---	--