



Kauno technologijos universitetas

Informatikos fakultetas

**Programavimo mokymosi aplinka, pritaikyta skirtingų
mokymosi pasiekimų lygių mokiniams**

Baigiamasis magistro projektas

Vita Strasevičienė

Projekto autorė

Doc. Renata Burbaitė

Vadovė

Kaunas, 2024



Kauno technologijos universitetas

Informatikos fakultetas

Programavimo mokymosi aplinka, pritaikyta skirtingų mokymosi pasiekimų lygių mokiniams

Baigiamasis magistro projektas

Nuotolinio mokymosi informacinės technologijos (6211BX010)

Vita Strasevičienė

Projekto autorė

Doc. Renata Burbaitė

Vadovė

Doc. Vytenis Punys

Recenzentas

Kaunas, 2024



Kauno technologijos universitetas

Informatikos fakultetas

Vita Strasevičienė

Programavimo mokymosi aplinka, pritaikyta skirtingų mokymosi pasiekimų lygių mokiniams

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;
2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąrašė;
3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);
4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Vita Strasevičienė

Patvirtinta elektroniniu būdu

Strasevičienė Vita. Programavimo mokymosi aplinka, pritaikyta skirtingų mokymosi pasiekimų lygių mokiniams. Baigiamasis magistro projektas / vadovė Doc. Renata Burbaitė; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Programų sistemos (B03), Informatikos mokslai.

Reikšminiai žodžiai: informacinės technologijos, tradicinis mokymasis, nuotolinis mokymasis.

Kaunas, 2024. 53 p.

Santrauka

Programavimas bendro lavinimo mokyklose yra ne tik svarbus įgūdis mokiniams, tačiau ir galingas įrankis plėtoti skaitmeninę visuomenę ir ruošiant mokinius ateičiai. Gyvename skaitmeninėje eroje, todėl svarbu suprasti, kaip veikia technologijos. Programavimas padeda mokiniams susipažinti su šiomis technologijomis ir jų veikimo principais. Taip pat mokiniai, kurie turi programavimo įgūdžių, labai vertinami darbo rinkoje. Mokiniai, turintys šiuos įgūdžius, turi didesnes galimybes įsidarbinti ateityje. Programavimo mokymas padeda užtikrinti skaitmeninį raštingumą, kuris yra būtinas šiuolaikiniame pasaulyje. Programavimas skatina kūrybiškumą, leidžia mokiniams kurti projektus ir juos įgyvendinti.

Baigiamojo projekto tikslas – pagerinti mokinių mokymosi motyvaciją ir ugdymo kokybę, sukuriant ir išdiriant tinkamiausią programavimo mokymosi aplinką, kuri būtų pritaikyta skirtingų mokymosi pasiekimų lygių mokiniams.

Sociologiniame tyrime dalyvavę III gimn. klasės programavimo besimokantys mokiniai pasigedo vienos programavimo mokymui(si) skirtos aplinkos, kurioje būtų galima rašyti programas, nagrinėti teorinę medžiagą, atlikti testus, dalintis savo sukurtomis programomis, jomis dalintis, analizuoti programų pavyzdžiu ir bendradarbiauti.

Atsižvelgus į sociologinio tyrimo rezultatus buvo nuspręsta integruoti į naudojamą mokymo valdymo sistemą *Moodle programavimo mokymui(si) skirtą aplinką*. Todėl buvo sukurtas aplinkos modelis ir integruotas į *VMA Moodle*. Programavimo mokymui(si) skirta aplinka buvo išbandyta ir naudojama mokymo(si) procese Vievio gimnazijoje.

Įdiegtą programavimo mokymui(si) skirtą aplinką vertino mokiniai atsakydami į jiems pateiktus anketos klausimus. Programavimo mokymui(si) skirta aplinka buvo įvertinta labai gerai, o atsižvelgiant į kai kurias pastabas siektina į jas atsižvelgti ir skirtą aplinką tobulinti: pateikti daugiau teorinės medžiagos ir įvairesnių užduočių. Numatytas ir mokymosi aplinkos tęstinumas: aplinka bus naudojama ir ateityje ir pristatyta savivaldybės informatikos mokytojams.

Strasevičienė Vita. A Programming Learning Environment Adapted to Students at Different Levels of Learning Achievement. Master's Final Degree Project / supervisor assoc. prof Renata Burbaitė; Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): Software Engineering (B03), Computing.

Keywords: information technologies, traditional learning, distance learning.

Kaunas, 2024. 53 p.

Summary

Programming in general education schools is not essential skill for students, but also a powerful tool for developing a digital society and preparing students for the future. We live in a digital era, so it is beneficial to understand how technology works. Programming helps for students learn about how these technologies work. Students with coding skills are also highly valued in the field of the employment. Students with these skills have an opportunity to find a better skilled job in the future. Teaching coding helps ensure digital literacy, which is important in today's society. Programming motivates creativity, allowing students to designed and implement projects.

The main objective of the final project- to improve students` learning motivation, and the quality of education by designing and testing the most appropriate coding learning environment for students with different need.

The sociological survey has been participated by respondents of III grades students and they have been lacked of a single environment programming, where they could be written programs, study theory material, take tests, share and collaborate on the programs they had developed and analyse programs.

As per the data of the sociological study, it was decided to integrate the Moodle environment for teaching programming into the learning management system used. Therefore, an environment model was developed and integrate into VMA Moodle. This programme has been tested and used in Vievis Gymnasium as teaching/ learning tool.

The evaluation of installed environment for learning programming was questionnaire by students. The evaluation of the environment for teaching programming was very well, and some comments should be taken into account and improved by providing more theoretical material and more varied tasks. Continuity of the learning environment is also expected: the environment will be used in the future and presented to the municipality`s computer science teachers.

Turinys

Paveikslų sąrašas	3
Lentelių sąrašas	4
Įvadas.....	5
1. Programavimo nauda mokiniams.....	8
1.1. Programavimo samprata	8
1.2. Programavimo mokymui(si) skirtos aplinkos	13
1.3. Programavimo aplinkos bendrojo lavinimo mokyklose	18
1.4. Programavimo mokymosi aplinkos, pritaikytos skirtingų pasiekimų lygių mokiniams, kūrimo galimybės Vievio gimnazijoje.....	18
1.5. Kiekybinis tyrimas siekiant išsiaiškinti mokinių ir mokytojų nuomonę apie programavimo mokymui(si) skirtas aplinkas	20
2. Virtualiosios mokymosi aplinkos aprašas	23
2.1. Virtualiosios mokymosi aplinkos dalyviai ir jų poreikiai	24
2.2. VMA sistemos ir posistemiai	24
2.3. Virtualios mokymo aplinkos funkciniai ir nefunkciniai reikalavimai	25
2.4. Panaudojimo atvejų sąrašas pagal posistemas, dalyvių sąrašas ir jų funkcijos	27
2.5. Programinės įrangos parinkimas VMA realizuoti	32
3. Programavimo mokymui(si) skirtos aplinkos modeliavimas	35
3.1. Aplinkos modeliavimas	35
3.2. Programavimo mokymui(si) skirtos aplinkos funkcionalumas	39
3.3. Skyriaus išvados	41
4. Kurso kūrimas VMA Moodle ir jos dalyviai	42
4.1. Užduotys pagal mokinių pasiekimų lygius	44
4.2. Integruojamos programavimo mokymui(si) skirtos aplinkos įrankio „Shellter“ aprašas	44
4.3. Interaktyvus mokymosi turinio pateikimas SCORM.....	46
4.4. Sukurtos mokymosi aplinkos vertinimas	47
4.5. Mokymosi aplinkos tobulinimas ir tęstinumas	48
4.6. Skyriaus išvados	49
Išvados	50
Literatūros sąrašas	51
Priedai.....	54
1 priedas. Tyrimas „Programavimo aplinkos kūrimas, skirtingų poreikių mokiniams“	54
2 priedas. Tyrimas „Programavimo aplinkų poreikis, kuris reikalingas skirtingų poreikių mokiniams“	56
3 priedas. „Programavimo mokymui(si) skirtos aplinkos naudojimas“	58
4 priedas. Programavimo mokymosi aplinkos, pritaikytos skirtingų mokymosi lygių pasiekimų mokiniams, diegimo aktas Vievio gimnazijoje	61

Paveikslų sąrašas

1 pav. Problemų medis.....	14
2 pav. Tikslų medis	15
3 pav. Programavimo dėstymas pagal klases	20
4 pav. Respondentų pasiskirstymas pagal klases	21
5 pav. Mokinių naudojama papildoma medžiaga	22
6 pav. Administratoriaus PA diagrama administravimo posistemyje.....	30
7 pav. Mokytojo PA diagrama kurso kūrimo posistemyje.....	31
8 pav. Mokymo turinio rengimo ir teikimo posistemės panaudojimo atvejų diagrama.....	31
9 pav. Bendravimo ir bendradarbiavimo posistemės panaudojimo atvejų diagrama.....	32
10 pav. Kuriamos aplinkos ontologija	36
11 pav. Programavimo aplinkos pasirinkimo požymių diagrama (naudotas įrankis: http://splot-research.org/)	37
12 pav. Informatikos kurso III gimn. kl. mokymo turinys.....	38
13 pav. Programavimo mokymui(si) skirtos aplinkos struktūra	39
14 pav. Sukurtas kursas „Informatika III gimnazinei klasei“	42
15 pav. Vienos temos struktūra VMA Moodle kurse	43
16 pav. Integruota programavimo mokymo aplinka	43
17 pav. Vieno pasiekimų lygio užduotis VMA Moodle kurse	44
18 pav. Vykdomoji ir kūrimo aplinkos	45
19 pav. Uždavinių publikavimas.....	46
20 pav. Užduotis įkelta naudojant SCORM paketą	46
21 pav. Užduočių kūrimas naudojant SCORM paketo kūrimo aplinką.....	46

Lentelių sąrašas

1 lentelė. Informatikos srities „Algoritmai ir programavimas“ mokinių pasiekimų raida.....	10
2 lentelė. Informatikos pagrindinio ir vidurinio ugdymo pasiekimų lygiai	12
3 lentelė. VMA funkciniai ir nefunkciniai reikalavimai	26
4 lentelė. VMA dalyvių posistemės	28
5 lentelė. VMA palyginimas	33
6 lentelė. III gimn. klasės mokymo turinys ir gebėjimai	38

Įvadas

Programavimo mokymas, ypač mokymasis – aktuali problema. Pagal atnaujintas bendrąsias ugdymo programas, programavimo pradeda mokytis nuo pirmos klasės. Mokiniai supažindinami su programų kūrimu, pavyzdžiui pasitelkdami Logo arba Scratch aplinką. Vėliau, nuo septintos klasės įvedamas tekstinis programavimas. Programavimas – sudėtingas intelektinis procesas, tad norint juo sudominti mokinius reikia pateikti jį paprastai, aiškiai, patraukliai. Tačiau kad ir kaip patraukliai būtų pateikiama, vien žinių ar gerų pavyzdžių neužtenka: būtina pačiam įsitraukti į procesą, lavinti įgūdžius, logiškai mąstyti. Daugybė pažinimo teorijų šalininkų bandė atsakyti į klausimą, kodėl tiek daug besimokančiųjų nesugeba išmokyti programavimo? Štai keletas pastabų:[1].

- sunku suprasti programų tikslus ir jų ryšį su kompiuteriu;
- sunku suprasti specifinės programavimo kalbos sintaksę ir semantiką.

Atlikta nemažai tyrimų, kurie pabrėžia galimą didelę įtraukimo programuoti naudą į pradinių ir vidurinių mokyklų mokymo programą. Programavimo mokymas padeda ugdyti keletą įgūdžių:

- problemų sprendimo;
- kūrybinio mąstymo;
- algoritminio mąstymo;
- reflektivaus mąstymo;
- kritinio mąstymo;
- skaičiavimo mąstymo.

Šiuolaikiniam informatikos mokymui yra skiriamas itin didelis dėmesys. Pagal atnaujintas mokymo programas daug dėmesio skiriama temai „Algoritmai ir programavimas“. Ši tema turi plačias panaudojimo galimybes. Mokydamiesi šios temos, mokiniai susipažįsta su algoritmų ir programų nauda, atpažįsta ir vartoja pagrindines sąvokas, naudojami programavimo aplinkomis, testuoja programas. Mokiniai, mokydamiesi programavimo mokykloje ne tik įgaus programavimo žinių, tačiau taps gerais specialistais, taip pat įgytas žinias taikys įvairiose gyvenimo srityse.

Programavimo didinamasis poveikis neapsiriboja vien tik minėtais įgūdžiais, nes jis taip pat skatina matematikos, gamtos mokslų mokymąsi, socialinių mokslų, taip pat kalbų. Dėl šių priežasčių matyti, kad intensyviai stengiamasi mokytis programavimo mokyklose, ypač pastaraisiais metais kai dirbama pagal atnaujintas programas.

Programavimas - tai problemų perkėlimo į kompiuterinę aplinką procesas pasitelkiant programinę įrangą ir aplinkas tada, kai jos sumodeliuojamos. Programuojant svarbu ne tik veiksmų, bet ir duomenų, aprašų, pasirinktos kalbos formalizavimas, produkto užbaigimas ir pateikimas.

Pasak Lye ir Koh, programavimas reikalauja ne tik analizės įgūdžių, bet ir programavimo įgūdžių. Tai, kad programavimas kaip problemų sprendimų generavimo procesas reikalauja loginio mąstymo, mokiniams kyla sunkumų mokytis programavimo ir suvokti programavimą kaip pernelyg sudėtingą procesą, todėl pagal atnaujintas programas, programavimo mokoma nuo pirmos klasės.

Naudojamos grafikos pagrindu sukurtos vaizdinės programavimo aplinkos: Alice, Scratch, Microsoft SmallBasic, Toontalk, Stagecast Creator ir Code GameLab. Šios aplinkos yra skirtos vizualiniam programavimui, kurios palengvina programavimo mokymą [5].

Tekstiniam programavimui bendro lavinimo mokyklose naudojamos programavimo kalbos tokios kaip C++, kuri yra plačiai naudojama bendrosios paskirties į objektus orientuota programavimo kalba ir Python, kuri interpretuojama ir interaktyvi programavimo kalba [6].

Remiantis išvardintais teiginiais galima teigti, kad mokiniai informatikos pamokų metu turi tapti ne tik kompiuterio naudotoju, bet mokėtų programuoti ir testuoti programas.

Šiame darbe siekiama išanalizuoti kaip skirtingos programavimo aplinkos, taikomos skirtingų gebėjimų Vievio gimnazijos mokiniams, taip pat apžvelgiama kokios programavimo aplinkos galėtų būti potencialiai kuriamos skirtingų pasiekimų lygių 11-12 klasės mokiniams.

Darbo problema: pasirinkti tinkamą programavimo aplinką ir ją įdiegti į naudojamą mokymosi aplinką.

Darbo tikslas – pagerinti mokinių mokymosi motyvaciją ir ugdymo kokybę, sukuriant ir ištiriant tinkamiausią programavimo mokymosi aplinką, kuri būtų pritaikyta skirtingų mokymosi pasiekimų lygių mokiniams.

Darbo objektas: darbe tiriama aplinka, pritaikyta skirtingų mokymosi pasiekimų lygių mokiniams.

Darbo uždaviniai:

1. išanalizuoti mokslinę literatūrą apie programavimo mokymuisi skirtas aplinkas;
2. apžvelgti virtualiųjų mokymosi aplinkų ir priemonių panaudojimo programavimui mokytis galimybes;
3. detalizuoti programavimo mokymosi aplinkos, pritaikytos skirtingų gebėjimų mokiniams, kūrimo galimybes Vievio gimnazijoje, 11-12 klasės;
4. suprojektuoti ir įdiegti programavimo mokymuisi skirtą aplinką;
5. iširti sukurtą aplinką, pateikti rekomendacijas jos tobulinimui.

Darbo rezultatas: pagerėjusi mokinių motyvacija ir sukurta skirtingų poreikių mokiniams pritaikyta programavimo mokymui(si) skirta aplinka.

Darbo produktas: įdiegta programavimo mokymuisi skirta aplinka, pritaikyta skirtingų mokymosi pasiekimų lygių mokiniams.

Darbo struktūra:

Darbą sudaro įvadas, keturi skyriai, išvados, literatūros sąrašas ir priedai. Darbo apimtis yra 53 puslapiai, 21 paveikslas, 6 lentelės. Literatūros sąrašė 29 šaltiniai. Darbo pabaigoje pateikti 4 priedai.

Įvade pateikiamas temos aktualumas, įvardijame problema, darbo objektas, tikslas ir uždaviniai, darbo rezultatas ir produktas.

Pirmame skyriuje pristatyta literatūros šaltinių apžvalga. Išanalizuota programavimo nauda, informatinio mąstymo įgūdžiai. Aprašomi vidurinio ugdymo uždaviniai, mokinių pasiekimų lygiai. Išanalizuotos programavimo aplinkos ir aprašytas kiekybinis tyrimas, pateikiamos išvados.

Antrame skyriuje aprašyta virtuali mokymo aplinka, jos dalyviai ir jų poreikiai. Išanalizuoti VMA sistemos ir posistemiai, funkciniai ir nefunkciniai reikalavimai. Taip pat pateiktos panaudojimo atvejų diagramos bei priemonė VMA realizuoti. Įvertinus įvairius kriterijus nustatyta, kad programavimo aplinka bus įdiegta *Moodle* VMA.

Trečiame skyriuje aprašytas programavimo mokymui(si) skirtos aplinkos modeliavimas. Sudaryta ontologija, kuri atvaizduoja informatikos III-IV gimnazijos klasių informatikos mokymo(si) turinį, programavimo mokymui(si) skirtas platformas, programavimo redaktorius, mokomąją medžiagą, mokymosi resursus bei požymių diagrama, kuri pateikia išsamų programavimo mokymui(si) skirtų aplinkų vaizdavimą. Sudaryta programavimo mokymui(si) skirtos aplinkos struktūra.

Ketvirtame skyriuje pristatytas sukurto kurso ir įdiegtos programavimo mokymui(si) skirtos aplinkos aprašas. Aprašytas programavimo aplinkai įdiegti pasirinktas įrankis *Schellter*. Klausimyno pagalba atliktas įdiegtos programavimo mokymui(si) skirtos aplinkos, pritaikytos skirtingų pasiekimų lygių mokiniams vertinimas ir pasiūlytos rekomendacijos jos tobulinimui.

1. Programavimo nauda mokiniams

Programavimo mokymasis yra reikšminga žmogaus raidos dalis, galinti pagerinti kognityvinius procesus, pasiekimus mokymosi procese ir kt. Šiuos teiginius įrodo mokslinės literatūros šaltinių analizė, kurioje akcentuojama programavimo nauda besimokantiejiems. Pavyzdžiui, Louka[3] nurodo, kad programavimas skatina mokinių gebėjimą ir pasitikėjimą būti kūrybingais problemų sprendėjais. Ateityje tai gali padėti spręsti iššūkius, su kuriais susiduriama kasdieniame gyvenime, pasitelkiant lankstesnę mąstymą. Be to, programavimas padeda pagerinti ir akademinis pasiekimus ugdymo įstaigoje. Mokymasis programuoti yra vienas iš būdų, kaip juos geriausiai paruošti ateičiai, padėti pasiekti aukštesnįjį mokymosi lygį mokykloje arba palaikyti ir pagerinti dabartinį mokymąsi. Be to, yra mokslinių tyrimų išvadų, įrodančių, kad programavimas stiprina:

- organizuotumą;
- atkaklumą;
- problemų sprendimo įgūdžius;
- pasitikėjimą savimi.

1.1. Programavimo samprata

Informatinį mąstymą (IM) pirmą kartą 1990 m. pristatė S. Papertas (žymiausias iš šiuolaikinių švietimo teoretikų). Nuo to laiko jo apibrėžimą, mokymą ir vertinimą aptarė įvairūs mokslininkai tokie kaip Grover ir Pea (2013) ir Wing (2006) [23]. Jie pabrėžė, kad informatinis mąstymas yra vienas iš kasdienio gyvenimo įgūdžių, kuris reikalingas kiekvienam, o ne tik programavimo įgūdis, kuriuo naudojami tik kompiuterių specialistai.

Informatinis mąstymas laikomas svarbia kompetencija, kuri reikalinga norint prisitaikyti prie ateities. Tačiau pedagogai, ypač 12 klasių mokytojai ir mokslininkai, nėra aiškiai nustatę, kaip jo mokyti. 2016-2017 m. buvo atliktas tyrimas, akademiniuose žurnaluose paskelbtų tyrimų metų apžvalga, siekiant išanalizuoti IM mokymo taikomuosius kursus, pritaikytas mokymosi strategijas, dalyvius, mokymo priemones, programavimo kalbas ir kursų kategorijas. Iš apžvalgos rezultatų nustatyta, kad per pastarąjį dešimtmetį IM skatinimas švietime padarė didelę pažangą. Pastaraisiais metais įvairiose šalyse ne tik daugėja IM studijų, bet ir įvairėja dalykai, tyrimų problematika ir mokymo priemonės. Daugumoje tyrimų daugiausia dėmesio buvo skiriama programavimo įgūdžių lavinimui ir matematiniam skaičiavimui, o kai kuriuose buvo taikomas įvairių sričių mokymo būdas, kad mokiniai galėtų kompiuteriu valdyti ir analizuoti įvairių sričių medžiagą. Be to, kadangi skirtingo amžiaus mokinių pažintiniai gebėjimai skiriasi, atitinkamai turėtų skirtis ir kompiuterinio raštingumo gebėjimų ugdymo metodai bei turinio kriterijai [5].

Informatinis mąstymas – tai samprotavimo būdas ir problemų sprendimo procesas, kai ugdomi gebėjimai atpažinti, formuluoti ir spręsti uždavinius, analizuoti ir tvarkyti duomenis, taikyti schemas ir modelius, automatizuoti sprendimus naudojantis skaitmeninėmis technologijomis. Skaitmeninio raštingumo ir informatinio mąstymo ugdymas leidžia informatiką integruoti su kitais dalykais sudarant efektyvias šiuolaikiško mokymo(si) galimybes, ugdyti ir plėtoti pažinimo, skaitmeninę, komunikavimo, socialinę, emocinę ir sveikos gyvensenos, kūrybiškumo, pilietinę, kultūrinę kompetencijas.

Automatizavimas – vienas pagrindinių informatinio mąstymo ugdymo komponentų. Jau nuo jauno amžiaus, išvelgia algoritmų ir programų naudą. Pagal atnaujintas programas algoritmų ir programų kūrimo gebėjimai ugdomi nuosekliai. Nuo pirmos klasės pradedama paprastais kasdieniais algoritmais, algoritmo žingsnių nustatymu ir atlikimu, veiksmų valdymo komandomis (seka, pasirinkimas, kartojimas), mokiniai dirba su vizualinio mokymo aplinkomis. Vėlesnėse klasėse mokomasi kurti programas ir programavimo, naudojantis žaidybinėmis programavimo aplinkomis. Nuo 7-os klasės mokomas tekstinis programavimas. Algoritmavimas ir programavimas – problemos (uždavinio) sprendimo kelio įvaldymas, pradedant siekiamo tikslo, formuluotės tikslinimu, algoritmo sudarymu, užrašymu, programos parengimu kompiuteriui, įvykdymu, testavimu, tobulinimu, dokumentavimu. Sprendžiant mokiniams rūpimas problemas supažindinama su klasikiais algoritmais (pavyzdžiui, mažiausios ir didžiausios reikšmių nustatymo, duomenų rikiavimo, dvejetainės paieškos, trumpiausio kelio radimo, kombinatorikos, duomenų šifravimo). Suteikiamos galimybės mokiniams nagrinėti euristinius algoritmus, išbandyti konkrečius neuroninių tinklų ar dirbtinio intelekto algoritmus.

Informatikos dalyko tikslas – sudaryti galimybę kiekvienam mokiniui ugdytis informatinį mąstymą siekiant sumaniai spręsti realias gyvenimo problemas; ugdytis gebėjimus kūrybiškai, atsakingai ir saugiai taikyti skaitmenines technologijas mokantis ir kitoje asmeninėje veikloje; nuolatos tobulinti skaitmeninę ir kitas kompetencijas, būtinas visaverčiam, sėkmingam gyvenimui šiuolaikinėje visuomenėje; pasirengti tolesnėms studijoms ir darbui (profesijai).

Programavimas – sudėtingas procesas, todėl norint jo išmokti daugelis turi įdėti daug darbo ir nugalėti sunkumus. Sudominti besimokančiuosius programavimu galima pateikiant jį labai paprastai, aiškiai, parenkant tinkamas aplinkas bei užduotis. Taip pat svarbu paminėti, kad programuoti išmoks tie, kurie daug dirbs savarankiškai ir įgis tinkamus įgūdžius ir išmoks logiškai mąstyti.

Kadangi, darbe remiamasi 10-12 kl mokymu programuoti, tad aprašomi vidurinio ugdymo uždaviniai. Siekdami tikslo mokiniai:

- plėtoja ir gilina mokantis pagal pradinio ir pagrindinio ugdymo programas įgytas informatikos žinias, ugdosi gebėjimus sumaniai, teisėtai ir tikslingai apdoroti duomenis, automatizuoti problemų sprendimą;
- veiksmingai taiko skaitmenines technologijas mokydamiesi kitų dalykų, asmeninėje veikloje ir tuo gerina savo mokymąsi;
- plėtoja gebėjimus saugiai ir atsakingai bendrauti ir bendradarbiauti naudojantis skaitmeninėmis technologijomis;
- aiškiai ir argumentuotai dėsto mintis, ugdosi informatinį, kritinį ir loginį mąstymą, kūrybingumą, pasitikėjimą savo jėgomis.

Daugumai mokinių sunkiai sekasi programuoti, dirbti savarankiškai, sunkiai įsisavina pamokos medžiagą. Skirtingų gebėjimų mokiniai skirtingai įsisavina medžiagą ir uždavinių sprendimus, todėl pastangų įdėti norint išmokti programuoti reikia skirtingai. Šiame darbe remiamasi atnaujintos informatikos programos [17] pasiekimų sritimi „Algoritmai ir programavimas“ (žr. 1 lentelę).

1lentelė. Informatikos srities „Algoritmai ir programavimas“ mokinių pasiekimų raida

Pasiekimas	Pasiekimų raida					
	1–2 klasių koncentras	3–4 klasių koncentras	5–6 klasių koncentras	7–8 klasių koncentras	9–10 (I–II gimnazijos) klasių koncentras	III–IV gimnazijos klasių koncentras
Algoritmai ir programavimas						
B1. Išvelgia algoritmų, programų naudą, atpažįsta ir vartoja pagrindines sąvokas.	Aptaria, kaip atliekami kasdieniai veiksmai, nurodo juos žingsniais ar komandomis	Pateikia ir apibūdina algoritmų, programų pavyzdžius iš kasdienės aplinkos	Aptaria sprendimų (algoritmų ir programų) automatizavimą, pagrindžia pavyzdžiais	Aptaria algoritmų ir programų kūrimo tikslus, duomenų ir programų sąveiką, integralumą	Atpažįsta realaus gyvenimo problemas, kurių sprendimą galima automatizuoti, įvardija galimus sprendžiamų uždavinių formulavimo sunkumus (daugiaprasmiškumą, netikslumą)	Atrenka nesunkiai kompiuteriu sprendžiamas realaus pasaulio problemas, surenka informaciją apie reikiamą informacinį įrankį (programą) ir pateikia galimo sprendimo idėją
B2. Naudojasi algoritmų kūrimo, programavimo kalbos konstrukcijomis, programavimo aplinkomis.	Atpažįsta ir vykdo komandų sekas, pasirinkimo komandą, skiria logines operacijas: NE, IR, ARBA	Taiko ir paaiškina pasirinkimo (šakojimo) ir kartojimo komandas	Programavimo aplinkoje randa reikiamas komandas, paaiškina programos vykdymo eigą, parodo rezultatus	Sprenddamas problemas naudoja programavimo kalbos konstrukcijas ir aplinką	Sprenddamas problemas naudoja išorinius duomenis	Kuria ir (ar) naudoja programai reikalingas duomenų struktūras (taip pat ir abstrakčiasias). Naudojasi įvairiais duomenų šaltiniais (pavyzdžiui, tekstiniais failais, jutikliais, internetu)
B3. Kuria ir vykdo algoritmus, programas.	Sudaro ir vykdo komandų sekas, naudojasi žaidybinėmis programavimo priemonėmis	Sprendžia uždavinį, sudaro ar pritaiko algoritmą, skaido į mažesnes dalis	Kuria programą, aprašo ir vykdo kelių veiksmų paprogrames, paaiškina jų funkcionalumą	Problemai spręsti kuria programas, parenka ir taiko tinkamus algoritmus	Projektuoja programą, naudoja paprogrames su parametrais	Analizuoja programinės įrangos sąsajos naudotojo poreikius, pateikia sąsajos projektą. Apibūdina ir kuria ar parenka reikiamus algoritmus informacijai apdoroti
B4. Testuoja, derina, tobulina programas.	Randa nurodytas klaidas komandų sekose ir jas	Tikrina, ar algoritmas, programa pateikia numatytus	Tobulina uždavinio sprendimą – algoritmą ir programą.	Kritiškai vertina programos rezultatus, jų pateikimą,	Atlieka programos dokumentavimą. Derina programą, vertina algoritmo efektyvumą	Modifikuoja, papildo funkcijomis savo ir kitų kuriamas

	taiso	rezultatus. Aptinka ir taiso klaidas komandų sekose, algoritmuose	Testuoja programą, parenka kontrolinius duomenis	patogumą naudotojui		programas. Naudoja grupinio programavimo įrankius
B1. Išvelgia algoritmų, programų naudą, atpažįsta ir vartoja pagrindines sąvokas.	Aptaria, kaip atliekami kasdieniai veiksmai, nurodo juos žingsniais ar komandomis	Pateikia ir apibūdina algoritmų, programų pavyzdžius iš kasdienės aplinkos	Aptaria sprendimų (algoritmų ir programų) automatizavimą, pagrindžia pavyzdžiais	Aptaria algoritmų ir programų kūrimo tikslus, duomenų ir programų sąveiką, integralumą	Atpažįsta realaus gyvenimo problemas, kurių sprendimą galima automatizuoti, įvardija galimus sprendžiamų uždavinių formulavimo sunkumus (daugiaprasmiškumą, netikslumą)	Atrenka nesunkiai kompiuteriu sprendžiamas realaus pasaulio problemas, surenka informaciją apie reikiamą informacinį įrankį (programą) ir pateikia galimo sprendimo idėją

Atnaujintoje informatikos programoje pateikiami skirtingiems mokinių amžiaus tarpsniams numatyti pasiekimai – mokymo(si) rezultatai. Mokymo(si) turinys nusako kontekstus, kuriuose ugdomi mokinių pasiekimai, ir mokymo(si) kontekstų pasirinkimo galimybes laipsniškai įgyti žinių ir supratimo, ugdyti gebėjimus ir vertybines nuostatas. Pasiekimai aprašomi keturiais pasiekimų lygiais [17] (žr. 2 lentelę):

- slenkstinis;
- patenkinamas;
- pagrindinis;
- aukštesnysis.

Kiekvienas pasiekimų lygių požymis nurodo mokinio siekiamus rezultatus. Aprašomos svarbiausios į(si)vertinimui reikšmingos žinios ir supratimas, gebėjimai ir vertybinės nuostatos.

Pasiekimų lygių požymiai skirti vertinti mokinių pasiekimus ir daromą pažangą. Jie padeda mokytojui stebėti, fiksuoti, apibendrinti individualius mokinių pasiekimus ir diferencijuoti užduotis. Reikėtų atkreipti dėmesį, kad tas pats pasiekimų lygis skirtinguose centruose skiriasi nagrinėjamos temos sudėtingumu ir gilumu. Pasiekimų lygių požymių aprašai skirti vertinti tam tikro mokymo(si) laikotarpio mokinių pasiekimus ir pažangą, o ne pavienius mokinių darbus ir atsakymus.

Aprašant pasiekimų lygių požymius naudojamos šios mokinių pasiekimų lygius rodančios skalės ir sąvokos:

1. Savarankiškumo:

- padedamas – atlieka užduotį dalyvaujant ar procesą moderuojant mokytojui;

- naudodamasis netiesiogine pagalba – atsakydamas į nukreipiamuosius klausimus, naudodamasis papildomai pateikta medžiaga, vadovaudamasis pateiktais kriterijais;
- konsultuodamasis – atlieka užduotį tikslingai klausdamas ar prašydamas patarimų;
- savarankiškai – užduotį atlieka be pagalbos, susidūręs su kliūtimis, randa būdų jas įveikti.

2. Konteksto:

- artima aplinka – mokiniui pažįstama, kasdienė aplinka (pavyzdžiui, mokyklos, namų);
- įprastas kontekstas – jau nagrinėtos reiškinių, proceso sąlygos, aplinkybės.

3. Naujas, neįprastas kontekstas – dar nenagrinėtos reiškinių, proceso sąlygos, aplinkybės, reikalaujančios naujų sprendimų.

2 lentelė. Informatikos pagrindinio ir vidurinio ugdymo pasiekimų lygiai

	Slenkstinis pasiekimų lygis	Patenkinamas pasiekimų lygis	Pagrindinis pasiekimų lygis	Aukštesnysis pasiekimų lygis
1–2 klasių koncentras	Nurodo kasdienio gyvenimo veiksmus, kurie atliekami žingsniais (B1.1).	Nurodo kasdienio gyvenimo veiksmus žingsniais ar komandomis (B1.2).	Aptaria, kaip atliekami kasdieniai veiksmai, nurodo juos žingsniais ar komandomis (B1.3).	Susieja kasdienes veiksmus su algoritmo samprata, išskiria ir paaiškina žingsnius ir komandas (B1.4).
3–4 klasių koncentras	Nurodo, ką atlieka vienas ar kitas nurodytas kasdienės artimos aplinkos algoritmas ar programa (B1.1).	Aptaria nurodytus algoritmų, programų pavyzdžius iš kasdienės artimos aplinkos (B1.2).	Pateikia ir apibūdina algoritmų, programų pavyzdžius iš kasdienės aplinkos (B1.3).	Randa, paaiškina ir palygina kasdienės aplinkos algoritmų, programų pavyzdžius, įžvelgia jų naudą (B1.4).
5–6 klasių koncentras	Pateikia pavyzdžių apie kompiuterių taikymą realaus gyvenimo problemoms spręsti (B1.1).	Pateikia pavyzdžių iš artimos aplinkos, kaip kompiuteriai palengvina žmogaus darbą ir tinka įvairioms problemoms spręsti (B1.2).	Aptaria sprendimų (algoritmų ir programų) automatizavimą, pagrindžia pavyzdžiais (B1.3).	Diskutuoja apie kompiuterių ir skaitmeninių technologijų taikymą įvairioms problemoms spręsti. Paaiškina automatizavimo svarbą ir galimybes, aptaria naujausias technologijas (B1.4).
7–8 klasių koncentras	Pateikia įvairios paskirties programų pavyzdžių (B1.1).	Pateikia programų taikymo pavyzdžių įvairiems dalykams (B1.2).	Aptaria algoritmų ir programų kūrimo tikslus, duomenų ir programų sąveiką, integralumą (B1.3).	Diskutuoja apie programų kūrimo tikslus, duomenų ir programų sąveiką, integralumą (B1.4).
9–10 (I–II gimnazijos) klasių koncentras	Nurodo realaus gyvenimo problemas, sprendžiamas kompiuteriu, aptaria uždavinių formuluočių problemišumą (B1.1).	Atpažįsta realaus gyvenimo problemas, kurių sprendimus galima automatizuoti, nurodo netikslumus uždavinių formuluočių (B1.2).	Aptaria realaus gyvenimo problemas, kurių sprendimą galima automatizuoti, įvardija sprendžiamų uždavinių formulavimo galimus sunkumus (daugiaprasmiškumą,	Kritiškai vertina, ar realaus gyvenimo problemas sprendimas tinkamas automatizuoti. Formalizuoja užduotį (B1.4).

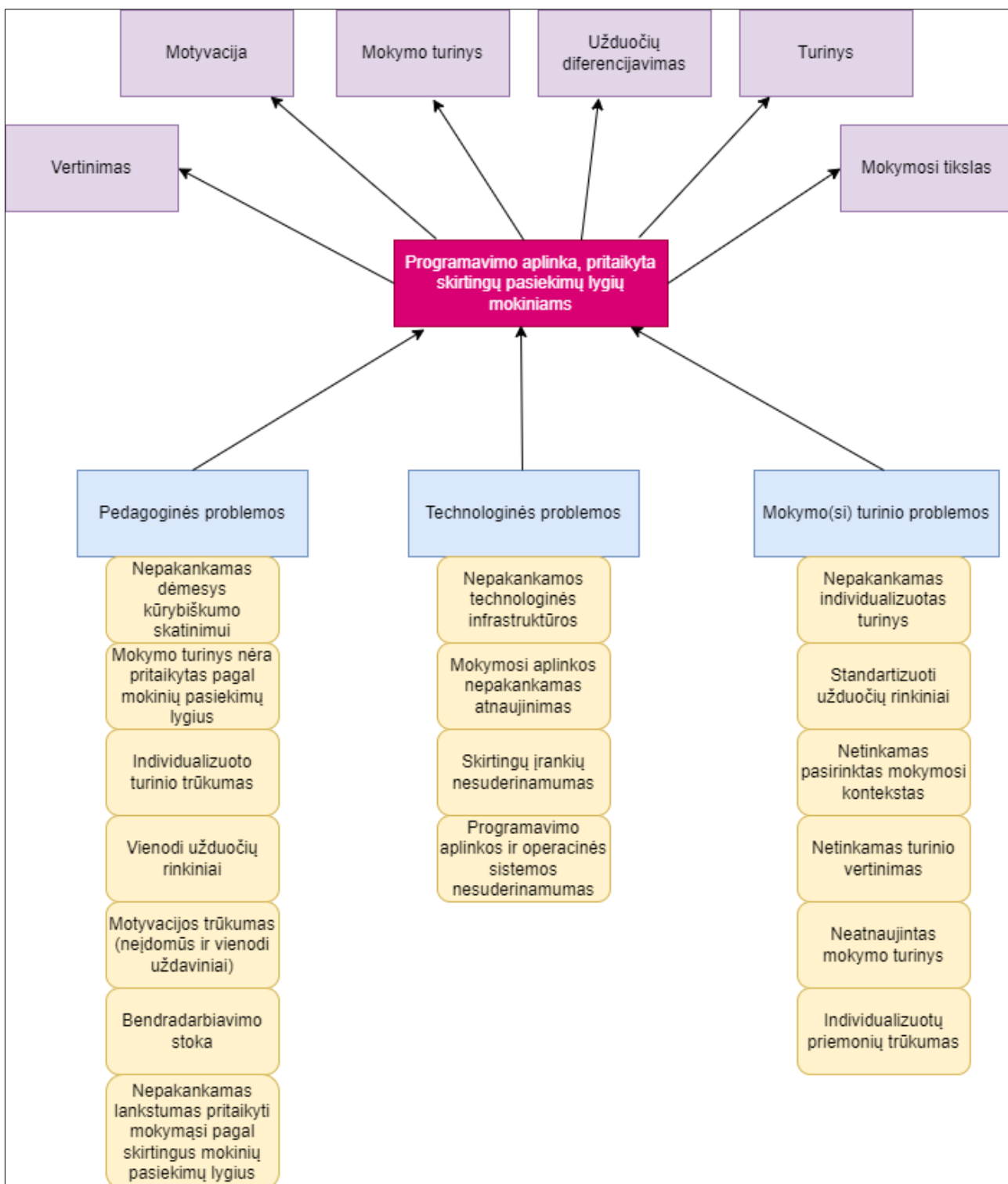
			netikslumą) (B1.3).	
III–IV gimnazijos klasių koncentras	Aptaria pateiktą realaus pasaulio problemą, nurodo galimą sprendimą (B1.1).	Atpažįsta nesunkiai kompiuteriu sprendžiamas realaus pasaulio problemas, diskutuoja apie nurodytą informacinį įrankį (programą) ir pateikia galimo sprendimo idėją (B1.2).	Atrenka nesunkiai kompiuteriu sprendžiamas realaus pasaulio problemas, surenka informaciją apie reikiamą informacinį įrankį (programą) ir pateikia galimo sprendimo idėją (B1.3).	Savarankiškai suranda nesunkiai kompiuteriu sprendžiamas realaus pasaulio problemas, pasirenka reikiamą informacinį įrankį (programą), pateikia ir paaiškina galimo sprendimo idėją (B1.4).

Remiantis Vievio gimnazijos mokinių pažangos ir pasiekimų vertinimo informatikos pamokose aprašu, galima teigti, kad mokymasis programuoti gali:

- padėti mokiniui pažinti save, suprasti savo stipriąsias ir silpnąsias puses, įsivertinti savo pasiekimų lygmenį, kelti mokymosi tikslus;
- padėti mokytojui išvelgti mokinio mokymosi galimybes, nustatyti problemas ir spragas, diferencijuoti ir individualizuoti darbą, parinkti ugdymo turinį ir metodus;
- nustatyti informacinių technologijų bei kompiuterių taikymų įsisavinimo kokybę, planuoti ugdymo turinį ir metodus, suteikti skirtingų mokinių poreikius atliepančią pagalbą.

1.2. Programavimo mokymui(si) skirtos aplinkos

Norint apibrėžti mokymosi aplinkos charakteristikas ir tinkamai jas suprojektuoti, labai svarbu korektiškai nustatyti besimokančiųjų mokymosi ypatumus ir sunkumus bei jų priežastis: bendruosius problemų sprendimo, algoritmų supratimo, modifikavimo ir naujų kūrimo, programavimo technikų naudojimo įgūdžius. Pastebima, kad iškyla problema, kad trūksta tokios programavimo mokymui(si) skirtos aplinkos, kuri būtų pritaikyta skirtingų mokymosi pasiekimų lygių mokiniams, todėl kyla sunkumų mokantis programavimo, trūksta motyvacijos, sunku pasiekti gerų mokymosi rezultatų, taip pat netinkamai pateiktas mokymo turinys.. Prasti mokinių mokymosi rezultatai yra ne tik dėl aplinkos trūkumo, bet ir dėl užduočių diferencijavimo pagal skirtingus mokinių pasiekimų lygius (1 pav.).



1 pav. Problemų medis

Svarbiausios charakteristikos, apibūdinančios efektyvią mokymosi aplinką:

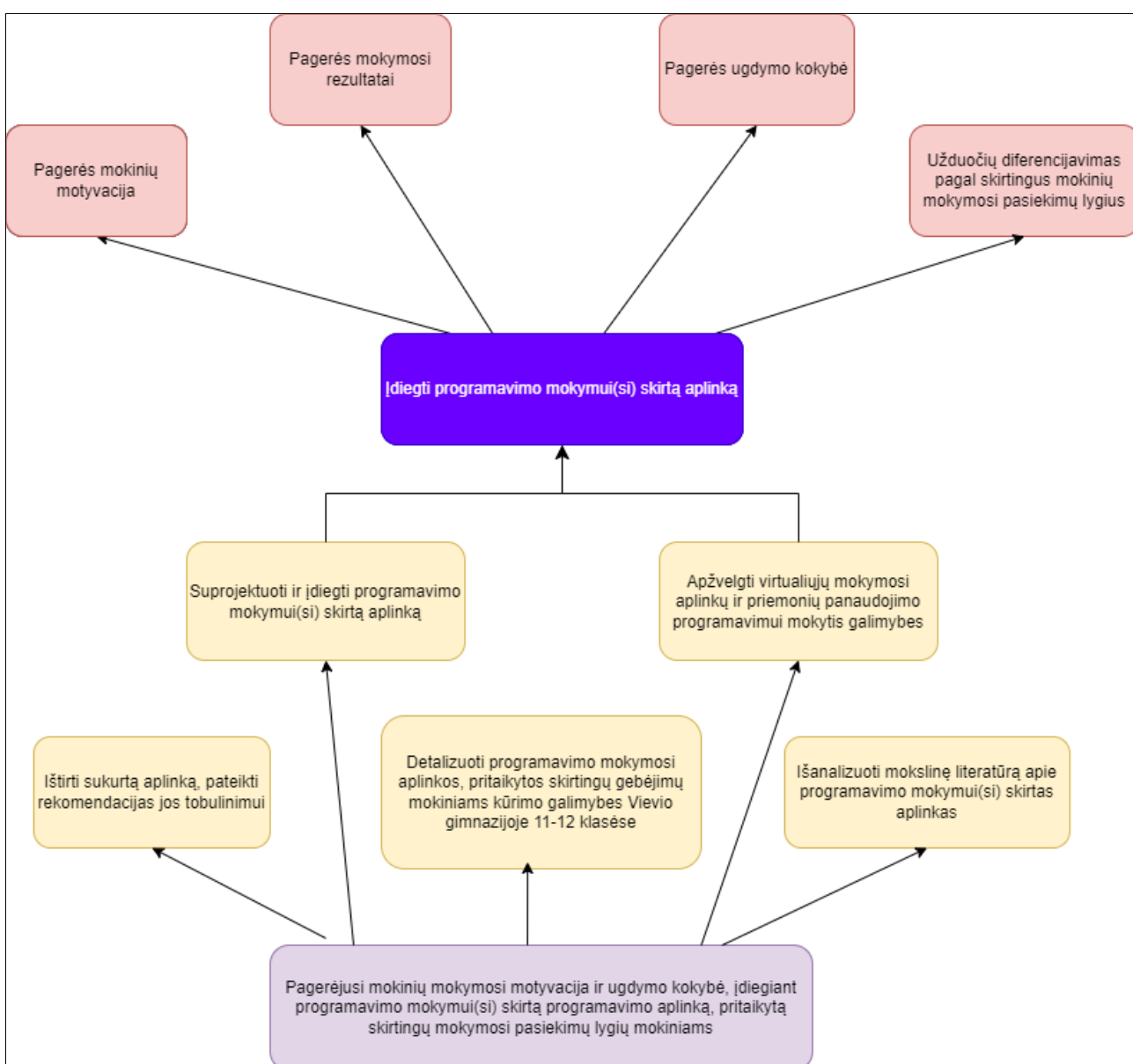
- 1) kiekvieno besimokančiojo žinių lygio nustatymas ir nuolatinis atnaujinimas bei dominuojančio mokymosi stiliaus nustatymas suteikia galimybę mokymąsi padaryti efektyvesnę ir labiau individualizuotą;
- 2) programavimo modelių naudojimas (uždavinio skaidymas į smulkesnes dalis, galimybė besimokančiajam pritaikyti žinias analogiškose situacijose) ugdo problemų sprendimo gebėjimus; 3)

žaidimai stiprina mokymosi motyvaciją, ugdo problemų sprendimo gebėjimus;

4) algoritmų kūrimo įrankių naudojimas aplinkoje suteikia galimybes besimokančiajam įgyti algoritmų sudarymo bei realizavimo žinių ir įgūdžių [21].

Būtina užtikrinti, kad mokiniai galėtų naudotis programavimo aplinka, kuri būtų pritaikyta skirtingų pasiekimų lygių mokiniams. Tam reikalingas nuolatinis mokytojo palaikymas, bendradarbiavimas tarp mokytojo ir mokinio siekiant pasiekti kuo geresnių mokymosi rezultatų. Programavimo aplinka, pritaikyta skirtingų pasiekimų lygių mokiniams gali padėti lengviau atlikti užduotis, stiprinti motyvaciją, norą mokytis.

Atlikus problemos analizę, akivaizdu, kad įdiegiant programavimo mokymui(si) skirtą aplinką, pagerės mokinių motyvaciją, pagerės mokymosi rezultatai. Šiam tikslui pasiekti būtina surasti tinkamą programavimo mokymui(si) skirtą aplinką ir ją įdiegti į naudojamą virtualią mokymo sistemą (2 pav.).



2 pav. Tikslų medis

Šiuo metu itin daug mokymų centrų ir švietimo institucijų siūlo mokytis populiariausių programavimo kalbų, tokių kaip JavaScript, Java, C/C++, Python, PHP. Programuotojo specialybė šiuo metu yra viena populiariausių ir gerai apmokamų specialybių, todėl GitHub yra priemonė gauti pirmą programuotojo darbą.

GitHub [25] yra įrankis peržiūrėti kodui, sekti jo apskeitimams ir padėti komandoms kartu dirbti prie vieno projekto. *GitHub* yra platforma, kurios pagalba programuotojams galima dirbti komandose. *GitHub* yra bene populiariausia, ji ne tik naudojama privačioms kodo saugykloms laikyti, *GitHub*'e programuotojai dažniausia laiko atvirosios prieigos projektus, kur prie jų kartu dirba.

Naudodami *Git* [25] programą, daug programuotojų gali dirbti prie vieno projekto vienu metu. *Git* ypatingai palengvina situacijas kai keli programuotojai keičia tą pačią kodo dalį, taip pat padeda atsekti pakeitimus, rasti kas juos atliko, leidžia keliauti per kodo versijas ir išsirinkti tik reikiamus pakeitimus. Taip pat labai svarbi darbo proceso su *Git* dalis yra programuotojų bendravimas, diskusija apie kodo pakeitimus, kodo peržiūros ir galimybė diegti tik jau patikrintus, tinkamus ir prižiūrėtojų patvirtintus pakeitimus[4].

Dar viena labai plačiai naudojama informatikos mokytojų programavimo mokymui(si) skirta aplinka yra www.onlinegdb.lt. *Onlinegdb* – tai programa, kuri paleidžia tikslią programą kontroliuojamoje aplinkoje, kurioje galima valdyti programos vykdymą ir pamatyti būseną kai programa yra sustabdyta. *Onlinegdb* platformoje galima programuoti *C*, *C++*, *Python*, *PHP*, *HTML* ir kitomis programavimo kalbomis. *Onlinegdb* suteikia galimybę, naudoti programą debesų aplinkoje savo naršyklėje. Taip pat, programavimo aplinka patogi tuo, kad galima nustatyti lūžio taškus. Tai puiki internetinė programavimo aplinka su kodų redaktoriumi, kompiliatoriumi ir derintuvu.

„*GeeksforGeeks*“ [26] yra populiariausias kompiuterių mokslo portalas su daugybe straipsnių ir viena geriausių svetainių, skirtų kodui mokytis ir praktikuoti. Joje taip pat yra išmanioji IDE, kurioje galima žaibišku greičiu paleisti kodą naudojant pasirinktines įvestis. Svetainė turi keletą puikių savybių. Į teksto sritį galima parašyti arba įklijuoti bet kurios populiarios programavimo kalbos kodą ir lengvai paleisti kodą internete. Taip pat galima atsisiųsti ir sugeneruoti kodo URL, kad galima būtų lengvai bendrinti.

Idone [27] – tai nemokamas internetinis kompiliatorius ir derintuvas, specialiai sukurtas programuotojams ir kūrėjams, kuriame vos per kelias sekundes galima sukompiliuoti ir paleisti kodą naudojant pasirinktinę įvestį. Su *Ideone* galima kompiliuoti bet kurią programavimo kalbą. Jame taip pat yra keletas naudingų funkcijų, pvz., galima padaryti kodą slapta, kad būtų galima bendrinti su konkrečiais programuotojais, be to, tai yra geriausia platforma dalytis kodu su savo projekto nariais.

Repl.it [28] yra internetinė programavimo mokymui(si) skirta aplinka, leidžianti vartotojams rašyti programą ir kurti jas, taip pat kurti svetaines naudojant naršyklę ir leidžia vartotojams įvairiais būdais dalytis projektais. Jie taip pat rengia daugybe kodavimo konkursų su įdomiais prizais. Jis turi aukščiausios kokybės pakopą, vadinamą „*Hacker*“ planu, kuri suteikia daugiau vietos saugykloje, privataus kodo ir neriboto projekto bendradarbių, tačiau taip pat galima prieiti prie jų nemokamai 3 mėnesius su „*Github Education*“ planu.

Jdoodle [29] – tai nemokamas internetinis kompiliatorius ir kodų rengyklė, skirta išsaugoti, paleisti ir bendrinti kodą bet kada ir bet kur be jokių sąrankų ir aparatinės įrangos išteklių, o tik aktyvų

interneto ryšį ir žiniatinklio naršyklę. Galima pasirinkti bet kurią iš pageidaujimų programavimo kalbų, parašyti kodą ir įvertinti kodo rezultata.

DELYS yra interaktyvi edukacinė programinė įranga, aplinka, skirta informatikos mokymui vidurinėje mokykloje. Ji sukurta kaip aplinka, skirta padėti įveikti visus tuos dalykus, kurie kelia problemas suprantant sąvokas ir sprendžiant atitinkamas problemas. Ši aplinka turi specializuotas funkcijas ar įrankius, kurie padeda programuotojams greičiau ir efektyviau dirbti. Tai gali būti naudinga, ypač dirbant su specifiniais projektų tipais arba kalbomis, kurios turi savo specifikas ir reikalavimus[6].

Alice yra naujoviška 3D programavimo aplinka, kuri leidžia lengvai kurti animaciją pasakoti istoriją, vaidinti interaktyvų žaidimą arba vaizdo įrašą, kuriuo galima dalytis internete. „Alice“ yra laisvai naudojama mokymo priemonė, kuri skirta mokytis mokinius programuoti. Ji leidžia mokiniams išmokti pagrindinių programavimo sąvokų kuriant animacinius filmus ir paprastus vaizdo žaidimus [6].

Scratch yra vaizdinė programavimo kalba ir aplinka, kuri skirta pradinių programavimo sąvokų mokymui naujokams. Šioje aplinkoje galima kurti interaktyvius pasakojimus, animacijas, žaidimus ir kitus kūrinius naudojant blokelius. Šioje aplinkoje sukurtus projektus galima dalintis su kitais. Tai programavimo mokymui(si) skirta platforma, kuri padeda tobulinti programavimo įgūdžius. Šioje aplinkoje galima rasti projektų idėjų, mokomąją medžiagą ir tai padeda programavimą daryti įtraukiančiu procesu.

Dar viena labai plačiai naudojama programavimo mokymui(si) skirta aplinka – Visual studio code. Ji yra nemokama, kuri plačiai naudojama profesionaliam naudojimui. Ši aplinka tinkama bet kokio tipo programavimui. Ši aplinka palaiko daugelį programavimo kalbų, tokių kaip Python, C++, JavaScript ir daug kitų. Visual studio code turi automatinį programos užbaigimą, kuris padeda greičiau rašyti pačią programą. Šioje aplinkoje lengva rasti klaidas ir jas taisyti. Aplinka taip pat turi daug mokomosios medžiagos.

SCORM (Shareable Content Object Reference Model) – tai elektroninių mokymo medžiagų kūrimui ir valdymui internete skirtas standartas. Šis technologijų ir standartų rinkinys leidžia kurti objektus, jais keistis ir integruoti juos į įvairias mokymosi valdymo sistemas. Pagrindinis SCORM tikslas – užtikrinti, kad mokymo turinys būtų sukurtas taip, jis būtų ne tik perkeliamas iš vienos mokymo valdymo sistemos į kitą, bet, kad būtų bendrinamas tarp skirtingų mokymo valdymo sistemų ir įrankių.

Pagrindinės SCORM savybės:

- interaktyvumas – leidžia kurti interaktyvius mokymo turinius, tokius kaip testai, užduotys, žaidimai ir simuliacijos;
- tarpplatformiškumas – mokymosi turinys gali būti naudojamas skirtingose mokymo valdymo sistemose;
- turinio atnaujinimas – leidžia atnaujinti ir perduoti turinį;
- mokymo(si) rezultatų stebėjimas – suteikia galimybę stebėti mokymo(si) rezultatus. Tai leidžia mokytojams ir administratoriams stebėti mokinių pažangą ir analizuoti pasiekimus;

Šios pagrindinės savybės padeda užtikrinti efektyvų ir lankstų elektroninio mokymo turinio kūrimą ir valdymą.

1.3. Programavimo aplinkos bendrojo lavinimo mokyklose

Programavimo mokymas yra neatsiejamas nuo programavimo kalbos ir jos aplinkos. Programavimo kalbos aplinka turi palaikyti programinės įrangos kūrimo technologijas, skatinti programos priežiūros darbus. Programavimo kalba drauge su aplinka turi būti saugi ir patikima.

Pagal atnaujintas bendrąsias ugdymo programas, pradinėje mokykloje informatikos gebėjimų ugdymas integruojamas į įvairių dalykų pamokas. Mokyklai nutarus, informatika 1–4 klasėse gali būti mokoma kaip atskiras dalykas. Pagrindinės mokyklos 5–10 (II gimnazijos) klasėse informatika mokoma kaip atskiras privalomasis dalykas. Vidurinio ugdymo III–IV gimnazijos klasėse informatika yra pasirenkamasis dalykas. 1 – 6 klasėse mokoma vizualinio programavimo pradmenų. Mokytojai gali rinktis tokias programavimo mokymui(si) skirtas aplinkas kaip:

- Scratch;
- Komenskio Logo;
- Minecraft;
- xLogo online;
- LeGO Education WeDo 2.0;
- code.org ir kt.

Nuo 7-os klasės įvedamas tekstinis programavimas. Mokytojai gali rinktis iš dviejų siūlomų programavimo kalbų:

- C++;
- Phyton.

11 – oje klasėje mokiniams informatikos dalykas tampa pasirenkamuju dalyku.

Kiekvienais mokslo metais šalies mokyklose organizuojamos mokomųjų dalykų olimpiados ar įvairūs konkursai, kuriuose dalyvauja gabūs mokiniai. Kiekvienais metais organizuojama ir Lietuvos mokinių informatikos olimpiada, kuri vyksta keliais etapais, užduotis kuria mokyklos ar aukštesnės institucijos, kurios atsakingos už olimpiadas. Baigiamajame etape mokiniams leidžiama programuoti *C++ ir Phyton* kalbomis.

Apibendrinant galima teigti, kad pasirinkimo laisvė renkantis programavimo kalbas mokykloje nėra didelė, o ypač skirtingų poreikių mokiniams, kadangi norint išlaikyti egzaminą, visi privalo programuoti *C++* arba *Phyton* kalbomis.

1.4. Programavimo mokymosi aplinkos, pritaikytos skirtingų pasiekimų lygių mokiniams, kūrimo galimybės Vievio gimnazijoje

Vievio gimnazijoje pagal atnaujintas bendrąsias programas mokiniai mokomi programuoti nuo 5 klasės. 5–6 klasėse mokiniai mokomi programuoti Scratch ir code.org aplinkose. Nuo 7-os klasės

mokiniai pradedami mokytis programuoti C++ programavimo kalba. Daugelis mokinių, pasirinkusių mokytis informatiką 11-oje klasėje, siekia studijuoti šios krypties specialybes aukštosiose mokyklose. Vievio gimnazijoje naudojama CodeBlocks programavimo aplinka, kadangi ji atitinka egzamino reikalavimus. Nuo 2022-2023 mokslo metų mokiniai supažindinami su Onlinegdb programavimo aplinka. Kadangi programavimo mokymui yra skirta viena savaitinė pamoka, tad silpnesnių gabumų mokiniai susiduria su tam tikrais sunkumais:

- nespėja įsisavinti pamokos medžiagos;
- nemoka mokytis savarankiškai;
- turi mokymosi spragų;
- nėra programavimo aplinkos su kodų pavyzdžiais ir dėstoma medžiaga skirtingų poreikių mokiniams.

Viena iš didžiausių problemų yra ta, kad gimnazijoje nėra programavimo aplinkos, kuri būtų pritaikyta skirtingų mokymosi pasiekimų lygių mokiniams. Įdiegiant tokią programavimo aplinką į mokykloje naudojamą virtualiojo mokymosi aplinką Moodle mokiniams būtų patogiu ir naudinga mokytis programuoti, didėtų jų mokymosi motyvacija, gerėtų ugdymo(si) kokybė.

Renkantis programavimo aplinką, reikia atsižvelgti į skirtingų poreikių mokinius, klasifikuojant užduotis pagal mokinių gebėjimus, suskirstant į lygius, su keliais sprendimo būdų pavyzdžiais. Taip pat aplinkoje turėtų būti išdėstoma teorinė pamokos medžiaga. Pirminiai reikalavimai kuriamai aplinkai:

- mokymosi turinys pateiktas suprantamai ir nuolat atnaujinamas,
- numatytos įvairios veiklos mokymosi tikslui pasiekti,
- įdiegtos bendradarbiavimą užtikrinančios priemonės,
- teikiamas kokybiškas grįžtamasis ryšys.

1.5. Kiekybinis tyrimas siekiant išsiaiškinti mokinių ir mokytojų nuomonę apie programavimo mokymui(si) skirtas aplinkas

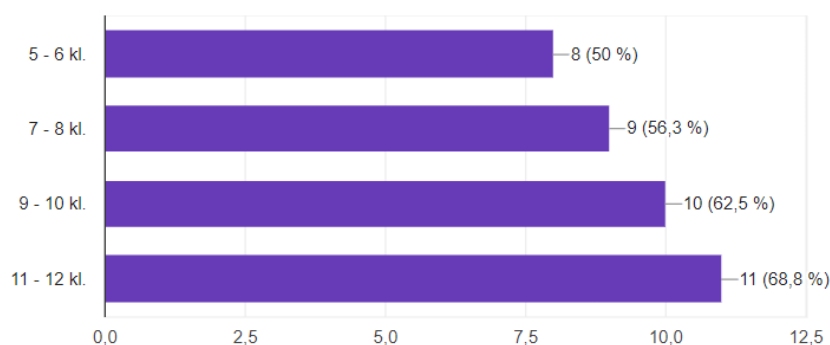
Siekiant išsiaiškinti aplinkos reikalingumą buvo atliktas kiekybinis tyrimas. Tyrime dalyvavo informatikos mokytojai ir programavimo modulį pasirinkę mokyti II-IV gimnazijos (10-12 klasės) mokiniai. Apklausiant mokytojus siekta išsiaiškinti:

- kokias programavimo aplinkas mokytojai naudoja,
- ar naudoja ir žino apie aplinkas pritaikytas skirtingų poreikių mokiniams,
- ar būtų naudinga turėti savo mokykloje programavimui skirtą aplinką, kuri būtų pritaikyta skirtingų poreikių mokiniams.

Apklausiant mokinius siekta išsiaiškinti:

- kodėl pasirinko programavimo pamokas,
- ar naudoja programavimo mokymuisi skirtas aplinkas,
- ar norėtų turėti programavimo mokymuisi skirtą aplinką, kuri būtų pritaikyta skirtingų poreikių mokiniams.

Klausimynas pateiktas 1 priede.



3 pav. Programavimo dėstymas pagal klases

Atliekant tyrimą apklausta 16 informatikos mokytojų iš visos Lietuvos ir 11 Elektrėnų savivaldybės Vievio gimnazijos 10-12 klasių mokinių.

Iš tyrimo dalyvavusių mokytojų beveik 70 % programavimą dėsto 11-12 klasėse (3 pav.). Likusiose klasėse programavimą dėsto ne mažiau negu pusė respondentų. Galima teigti, kad mokytojų atsakymai bus vertingi.

93,8 % iš apklausoje dalyvaujančių respondentų naudoja programavimo mokymui skirtas aplinkas, 6,3 % nenaudoja programavimo mokymui skirtų aplinkų.

Apklaustieji respondentai nurodė, kad naudoja tokias programavimo mokymui skirtas aplinkas:

www.onlinegdb.com, <https://dev.to>, <https://code.org/>, <https://vr.vex.com/>,
<https://xlogo.inf.ethz.ch/release/latest/#/>, <https://blockly.games/bird?lang=en>, <https://angis.lrt.lt/#/>

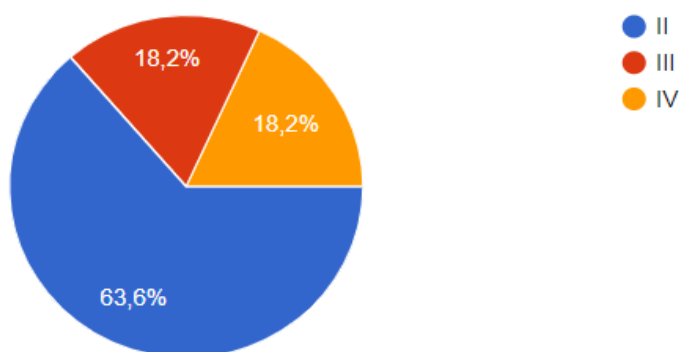
Taip pat buvo klausiama, ar pakankama programavimo mokymui(si) skirtų aplinkų pasiūla? Respondentai atsakė, kad mokant mokinius programuoti 10-12 klasėse, trūksta tinkamų programavimo mokymui(si) skirtų aplinkų.

53,3 % respondentų mano, kad būtų reikalinga programavimo mokymui skirta aplinka, kuri būtų pritaikyta skirtingų mokymosi pasiekimų lygių mokiniams, 46,7 % tikina, kad nereikia tokios aplinkos. Mokytojai teigia, kad mokiniai geriau įsisavintų žinias ir patirtų sėkmę. Respondentų manymu tokia aplinka būtina, nes visas mokymo(si) turinys būtų vienoje vietoje. Tie, kurie mano, kad aplinka nereikalinga motyvuoja tuo, kad aplinkų, skirtų programavimo mokymui(si) yra pakankamai.

80 % respondentų teigia, kad neturi savo mokykloje sukurtos programavimo mokymui skirtos aplinkos ir 20 % turi tokią aplinką. Dažniausiai naudojama aplinka yra www.pikta.lt

Toliau aptarsime mokinių nuomonę. Apklausoje dalyvavo 10-12 klasių mokiniai: 70% vaikinių ir 30 % merginų.

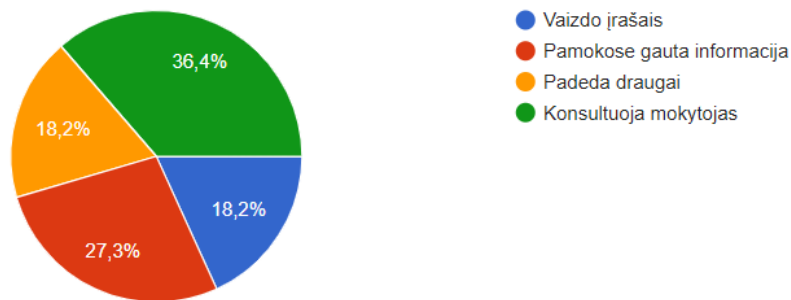
Apklausoje dalyvavusių mokinių pasiskirstymas pagal klases pateiktas 4 pav. Dauguma respondentų programavimo pamokas pradėjo lankyti nuo 10-os klasės. 63,6 % apklausos dalyvių pasirinko programavimo pamokas, kadangi ateityje nori pasirinkti programuotojo profesiją; 18,2 % pasirinko todėl, kad sekasi tikslieji mokslai ir 9,1 % tiesiog nori išbandyti save ir savo galimybes ir mano, kad įdomu išmokti šį dalyką ir tai bus naudinga ateityje. 100 % respondentų mokosi C++ programavimo kalbą.



4 pav. Respondentų pasiskirstymas pagal klases

54,5 % apklaustųjų teigia, kad vienos savaitinės valandos programavimo mokymuisi nepakanka, o 45,5 % sako, kad tenkina viena savaitinė pamoka.

Respondentų buvo klausiama, ar užtenka pamokų metu įgytų žinių? 63,6 % atsakė, kad pamokų metu gauna daug informacijos ir 36,4 % atsakė, kad daug mokosi savarankiškai.



5 pav. Mokinių naudojama papildoma medžiaga

Mokinių naudojama papildoma medžiaga mokymuisi pateikta 5 pav.

Tyrimo metu buvo klausiama, kas didintų motyvaciją mokantis programuoti. 45,5 % atsakė, kad mokytojas turėtų atsižvelgti į skirtingų poreikių mokinius, 36,4 % tenkintų programų kodai su pavyzdžiais ir 18,2 % norėtų, kad būtų skiriami namų darbai.

Į klausimą ar užtenka programavimui skirtų aplinkų, 72,7 % atsakė, kad užtenka. 18,2 % teigia, kad patys susirado mokymosi aplinką ir ja naudojasi ir 9,1 % sako, kad mokykloje nenaudoja jokių programavimo mokymuisi skirtų aplinkų.

72,7 % respondentų mano, kad būtų programavimo mokymuisi skirtos aplinkos yra reikalingos ir 27,3 % neturi nuomonės šiuo klausimu. Mokiniai, kurie atsakė, kad tokia aplinka būtų reikalinga, sako, kad būtų mokiniui patogiau kai vienoje aplinkoje būtų ir užduotys, ir testai, užduočių diferencijavimas.

Remiantis tyrimo rezultatais, būtų tikslinga sukurti programavimo mokymui(si) skirtą aplinką, pritaikytą skirtingų poreikių mokiniams, integruojant ją į Vievio gimnazijoje naudojamą virtualiąją mokymosi aplinką *Moodle*.

2. Virtualiosios mokymosi aplinkos aprašas

Virtualioji mokymosi aplinka (VMA) – tai kompiuterių tinklais ir kitomis informacinėmis komunikacinėmis technologijomis (toliau – IKT) pagrįsta ugdymo sistema, kurioje mokiniai mokosi mokytojų padedami. Kaip ir tradicinė mokymosi aplinka VMA sukurta, kad padėtų mokiniams mokytis, stebėti mokymosi procesą, tobulinti jo turinį. VMA vyksta visas ugdymo procesas: pateikiamas visas kurso turinys, bendraujama diskutuojant (diskusijų forumuose, pokalbiuose ar elektroniniu paštu), atliekamos praktinės užduotys, vyksta darbas grupėse, įgytos žinios bei gebėjimai tikrinami kompiuteriniais testais. Kaip ir tikraja mokymosi aplinka, VMA siekiama padėti mokiniams mokytis, stebėti mokymosi procesą, tobulinti jo turinį. Taigi VMA galėtume apibūdinti kaip mokomosios medžiagos, užduočių, bendravimo ir vertinimo priemonių sistemą, leidžiančią lanksčiai valdyti ugdymo procesą. VMA būtinai turi būti svarbiausi mokymo komponentai: mokomoji medžiaga, užduotys, mokinių mokymosi proceso ir rezultatų stebėjimas, vertinimas, mokinių ir mokytojų bendravimas. Kaip matyti, VMA atlieka nemažai funkcijų. Išskiriamos šios bendriausios virtualiųjų mokymosi aplinkų funkcijos (Dagienė, 2004):

- bendravimas ir bendradarbiavimas;
- vartotojų (mokinių, mokytojų, tėvų, kurso kūrėjų) registracija;
- ugdymo turinio tvarkymas (kompiuterinių priemonių kompleksas);
- užduotys (priemonės joms rengti ir atlikti);
- mokinių mokymasis ir pasiekimų stebėjimas;
- aplinkos sąsajos keitimas;
- informacijos apie naudojimąsi VMA teikimas – žinynas.

VMA reikalavimai:

1. Techninės sąlygos: tinkama kompiuterio ar kitų įrenginių aparatinė ir programinė įranga, kuri atitinka VMA reikalavimus. Stabilus ir greitas interneto ryšys, kad būtų užtikrintas sklandus turinio įkėlimas ir bendravimas su kitais dalyviais.
2. Stabilumas ir patikimumas: VMA turi būti stabiliai veikianti ir patikima. Ji neturėtų patirti dažnų sutrikimų ar klaidų, nes tai gali neigiamai paveikti mokymosi procesą ir naudotojų patirtį.
3. Suprantama naudotojams: VMA turėtų būti lengva naudoti ir suprantama net naujiems naudotojams. Mokymosi medžiaga turėtų būti aiški, patraukli vizualiai ir lengvai suprantama, kad vartotojai galėtų lengvai naviguoti per programą, pasiekti reikiamą turinį ir naudotis funkcijomis.
4. Prieinamumas: VMA turėtų būti prieinama ir pritaikyta skirtingoms platformoms ir įrenginiams. Tai gali apimti kompiuterius, planšetes, išmaniuosius telefonus ir kt. Be to, ji turėtų būti prieinama naudotojams su įvairiais gebėjimais ir galimybėmis, atsižvelgiant į prieinamumo standartus.

5. Turinio įvairovė: VMA turėtų leisti įkelti ir pateikti įvairaus pobūdžio turinį, tokį kaip tekstas, vaizdo įrašai, garso įrašai, interaktyvios simuliacijos, interaktyvūs uždaviniai ir kt. Tai padeda skirtingiems mokinio sugebėjimams ir poreikiams.
6. Interaktyvumas ir praktinės veiklos galimybės: VMA turėtų suteikti galimybę mokiniams aktyviai dalyvauti mokymosi procese. Tai gali apimti interaktyvius uždavinius, eksperimentų simuliacijas, praktinius projektus, diskusijų ir bendradarbiavimo funkcijas. Mokiniai turėtų turėti galimybę taikyti įgytas žinias ir gebėjimus praktikoje.
7. Stebėjimo ir vertinimo galimybės: VMA turėtų suteikti galimybę mokytojams stebėti mokinių pažangą, vykdyti įvairių veiklų ir užduočių vertinimą bei suteikti grįžtamąjį ryšį. Tai padeda pritaikyti mokymosi procesą ir individualizuoti mokymo strategijas.

2.1. Virtualiosios mokymosi aplinkos dalyviai ir jų poreikiai

Virtualiojoje mokymosi aplinkoje organizuojant ugdymo procesą yra įvairių dalyvių, todėl jiems yra priskiriami vaidmenys su atitinkamomis teisėmis pagal poreikius.

Pagrindiniai VMA dalyviai yra ugdymo proceso pagrindiniai dalyviai, kuriuos galima suskirstyti į šias grupes:

- Kurso/sistemos administratorius – vykdo visas leidžiamas VMA funkcijas;
- Dėstytojas/mokytojas/kurso kūrėjas – ugdymo procese naudoja savo sukurtą kursą, kurį gali pagal poreikius modifikuoti, valdyti (įtraukti, pašalinti, suskirstyti į grupes) kurso dalyvius ir pan.
- Dėstytojas be redagavimo teisės – naudojami sukurtais kursais ir organizuoja ugdymo procesą, atlieką kontrolę ir pan.;
- Studentas/mokinys/besimokantysis – gali naudotis kurso, kurio dalyvis jis yra, mokymosi medžiaga, atlikti užduotis ir pan.;
- Svečias – gali naudotis kurse esančia mokymosi medžiaga.

Siekiant efektyvaus mokymosi, visi dalyviai turi žinoti savo funkcijas, galimybes ir jas vykdyti.

2.2. VMA sistemos ir posistemiai

Kiekviena VMA – sistema, turinti pagrindines funkcijas, t. y., panaudojimo atvejai, kuriuos vykdant pasiekiamas atitinkamai sistemos daliai (posistemii) naudotojo iškeltas tikslas. Galima išskirti tokius pagrindinius VMA posistemius:

- administravimo;
- kurso kūrimo ir valdymo;
- mokymosi turinio pateikimo;
- mokymosi proceso organizavimo ir vertinimo;
- bendravimo ir bendradarbiavimo.

Administravimo posistemyje galima administruoti sistemą, registruoti dalyvius prižiūrint jų veiklų funkcijas.

Kurso kūrimo ir valdymo posistemis gali pateikti kurso kūrimo užklausa, valdyti kursą ir jo dalyvius.

Turinio pateikimo posistemis skirtas mokymosi medžiagos pateikimui bei modifikavimui, mokymosi medžiagos peržiūrėjimui, atsisiuntimui ar atsispausdinimui.

Mokymosi turinio rengimo ir proceso organizavimo ir vertinimo posistemyje galima tvarkyti veiklas, vertinti besimokančiuosius, stebėti jų pažangą, atlikti užduotis.

Mokymosi veiklų organizavimo ir vertinimo posistemė apima funkcijas naudojamas vertinimo sistemos kūrimui, stebėti pažangai bei grįžtamajam ryšiui gauti.

Bendravimo ir bendradarbiavimo posistemyje galima rašyti, skaityti žinutes, dalyvauti diskusijų forumuose, dalyvauti vaizdo konferencijose, atlikti užduotis bendradarbiaujant.

Šių posistemių funkcionalumas padaro VMA patrauklia ir prieinama mokantis nuotoliniu būdu.

2.3. Virtualios mokymo aplinkos funkciniai ir nefunkciniai reikalavimai

Kiekvienas VMA proceso dalyvis gali turėti skirtingus poreikius.

VMA reikalavimai gali būti suskirstyti į dvi kategorijas: funkcinis ir nefunkcinis reikalavimus.

Funkciniai reikalavimai:

Funkciniai reikalavimai nurodo, kokias funkcijas VMA turi atlikti naudotojai.

- Naudotojų prisijungimas: užtikrinti funkciją, kad naudotojai gali prisijungti prie sistemos su unikaliais slaptažodžiais, vartotojų vardais ar kitais būdais.
- Turinio pateikimas: užtikrinti, kad VMA turi funkciją pristatyti mokymosi turinį įvairiais būdais, t.y. vaizdo įrašais, pristatymais, įvairiais tekstais ir pan.
- Interaktyvios užduotys: užtikrinti, kad VMA turėtų funkciją ir galimybę teikti interaktyvias užduotis, kurias besimokantieji gali atlikti ir gauti grįžtamąjį ryšį.
- Bendradarbiavimas: užtikrinti, kad VMA turėtų funkciją, kuri leistų besimokantiems bendrauti ir bendradarbiauti su kitais dalyviais.

Nefunkciniai reikalavimai:

Nefunkciniai reikalavimai apibrėžia kaip VMA turi veikti, bet neapibrėžia konkretaus funkcionalumo. Nefunkciniai reikalavimai gali apimti:

- Našumo reikalavimus ir užtikrinti, kad VMA turėtų sklandų sistemos veikimą ir gebėjimą aptarnauti didelį kiekį naudotojų tuo pačiu metu.
- Prieinamumo reikalavimai: užtikrinti, kad VMA būtų prieinama visiems naudotojams.

Apibendrinti VMA funkciniai ir nefunkciniai reikalavimai pateikti 3 lentelėje.

3 lentelė. VMA funkciniai ir nefunkciniai reikalavimai

VMA posistemis	Funkciniai reikalavimai	Nefunkciniai reikalavimai
Administravimo posistemė	<ul style="list-style-type: none"> • Diegti sistemą ir ją valdyti • Parengti sistemą darbui • Valdyti kursus / kategorijas • Parinkti reikalingus įskiepius • Įregistruoti naudotojus • Kurti paskyras dalyviams • Tvarkyti dalyvių paskyras • Keisti sistemos kalbą • Nustatyti įkeliamo failo dydį • Atnaujinti sistemą ir įskiepius • Sukurti atsargines kopijas • Priskirti dalyviams vaidmenis • Pašalinti sistemos naudotoją • Atkurti prisijungimo slaptažodį • Kurti naujas grupes • Pranešti naudotojams apie veiklas • Pildyti kalendorių 	<ul style="list-style-type: none"> • Nustatyti ir keisti VMA išvaizdą • Nesudėtinga navigacija
Kurso kūrimo ir valdymo posistemė	<ul style="list-style-type: none"> • Kurti ir administruoti kursą • Pateikti kurso užklausa • Redaguoti kursų parametrus • Įtraukti į kursą naudotojus • Prisijungti prie kurso • Suteikti roles naudotojams • Pridėti kurso veiklas • Ištrinti kurso veiklas • Redaguoti kurso veiklas • Skirstyti naudotojus į grupes 	<ul style="list-style-type: none"> • Kurso dizaino keitimas • Temų išdėstymo nustatymas • Lengvas prisijungimas prie sistemos • Sistemoje naudojama lietuvių kalba • Nesudėtinga navigacija
Mokymosi turinio pateikimo posistemė	<ul style="list-style-type: none"> • Kurti kurse temas • Įkelti mokomąją medžiagą • Redaguoti mokymosi medžiagą • Ištrinti mokymosi medžiagą • Skaityti medžiagą • Parsisiųsti medžiagą • Išsaugoti medžiagą • Atsispausdinti medžiagą 	<ul style="list-style-type: none"> • Parinkti geriausią formatą mokomosios medžiagos pateikimui • Lengva prisijungti prie sistemos • Sistemoje naudojama lietuvių kalba • Aiškus medžiagos išdėstymas
Mokymosi proceso organizavimo ir vertinimo posistemė	<ul style="list-style-type: none"> • Kurti įvairias mokymosi veiklas • Redaguoti mokymosi veiklas 	<ul style="list-style-type: none"> • Paprasta ir patogi navigacija • Lengva prisijungti prie sistemos

	<ul style="list-style-type: none"> • Informuoti apie mokymosi veiklas • Kurti kontrolės ir savikontrolės testus • Atlikti užduoti • Atlikti testus • Matyti gautą įvertinimą • Skaityti dėstytojo komentarą • Stebėti pažangą ir besimokančiųjų veiklas • Kurti vertinimo sistemą • Gauti grįžtamąjį ryšį • Pateikti komentarus 	<ul style="list-style-type: none"> • Sistemoje naudojama lietuvių kalba • Vizualiai aiškus veiklų išdėstymas • Aiškiai matomi vertinimo reikalavimai • Paprastas informacijos atnaujinimas
Bendravimo ir bendradarbiavimo posistemė	<ul style="list-style-type: none"> • Rašyti, siųsti, trinti, skaityti, gauti, atsakyti, ieškoti žinučių. • Kurti bendravimo forumą • Redaguoti forumą • Valdyti forumo srautą • Rašyti asmeninę žinutę • Atsakyti į asmeninę žinutę • Nusiųsti failą • Dalyvauti tiesioginiame pokalbyje • Dalyvauti vaizdo konferencijose 	<ul style="list-style-type: none"> • Žinučių siuntimas keliems naudotojams • Aiškus žinučių pateikimas • Lengva prisijungti prie sistemos • Sistemoje naudojama lietuvių kalba • Patogi navigacija •

Skirtingos VMA gali turėti skirtingus funkcinis ir nefunkcinius reikalavimus. Renkantis VMA būtina atsižvelgti į naudotojų poreikius ir galimybes.

2.4. Panaudojimo atvejų sąrašas pagal posistemas, dalyvių sąrašas ir jų funkcijos

Analizuojant VMA galimybes, funkcijos gali būti priskiriamos skirtingiems dalyviams pagal jų poreikį ir galimybes. VMA mokymas turi būti efektyvus ir prieinamas visiems jo dalyviams.

VMA dalyviai gali būti:

Kurso administratorius – iš techninės pusės prižiūri sistemą, jos visą sklandų darbą, gali pridėti, pašalinti naudotojus, suteikia dėstytojams ar mokytojams vaidmenis.

Dėstytojas – atsakingas už mokomąją medžiagą, kursų veiklas, bendravimu su studentais ar mokiniais.

Besimokantysis – turi galimybę prisijungti prie kurso, studijuoti mokomąją medžiagą, atlieka užduotis, dalyvauja įvairiose organizuojamose veiklose.

Svečias – turi teisę naudotis tik pateikta mokomąja medžiaga.

VMA posistemiai

Kiekviena Virtualios mokymo aplinkos sistema (VMA) turi panaudojimo atvejus (PA). Pagrindiniai VMA posistemiai: (4 lentelė)

administravimo;

- kurso valdymo;
- mokymosi turinio pateikimo;
- mokymosi proceso organizavimo ir vertinimo;
- bendravimo ir bendradarbiavimo.

Kurso valdymo posistemis įgalima pateikti kurso kūrimo užklausa, valdyti kursą, jo dalyvius;

Turinio pateikimo posistemis skirtas mokymosi medžiagos pateikimui bei modifikavimui, mokymosi medžiagos peržiūrėjimui, atsisiuntimui ar atsispausdinimui;

Mokymosi proceso organizavimo ir vertinimo posistemyje galima tvarkyti veiklas, vertinti besimokančiuosius, stebėti jų pažangą, atlikti užduotis;

Bendravimo ir bendradarbiavimo posistemyje galima rašyti, skaityti žinutes, dalyvauti diskusijų forumuose, dalyvauti vaizdo konferencijose, atlikti užduotis bendradarbiaujant.

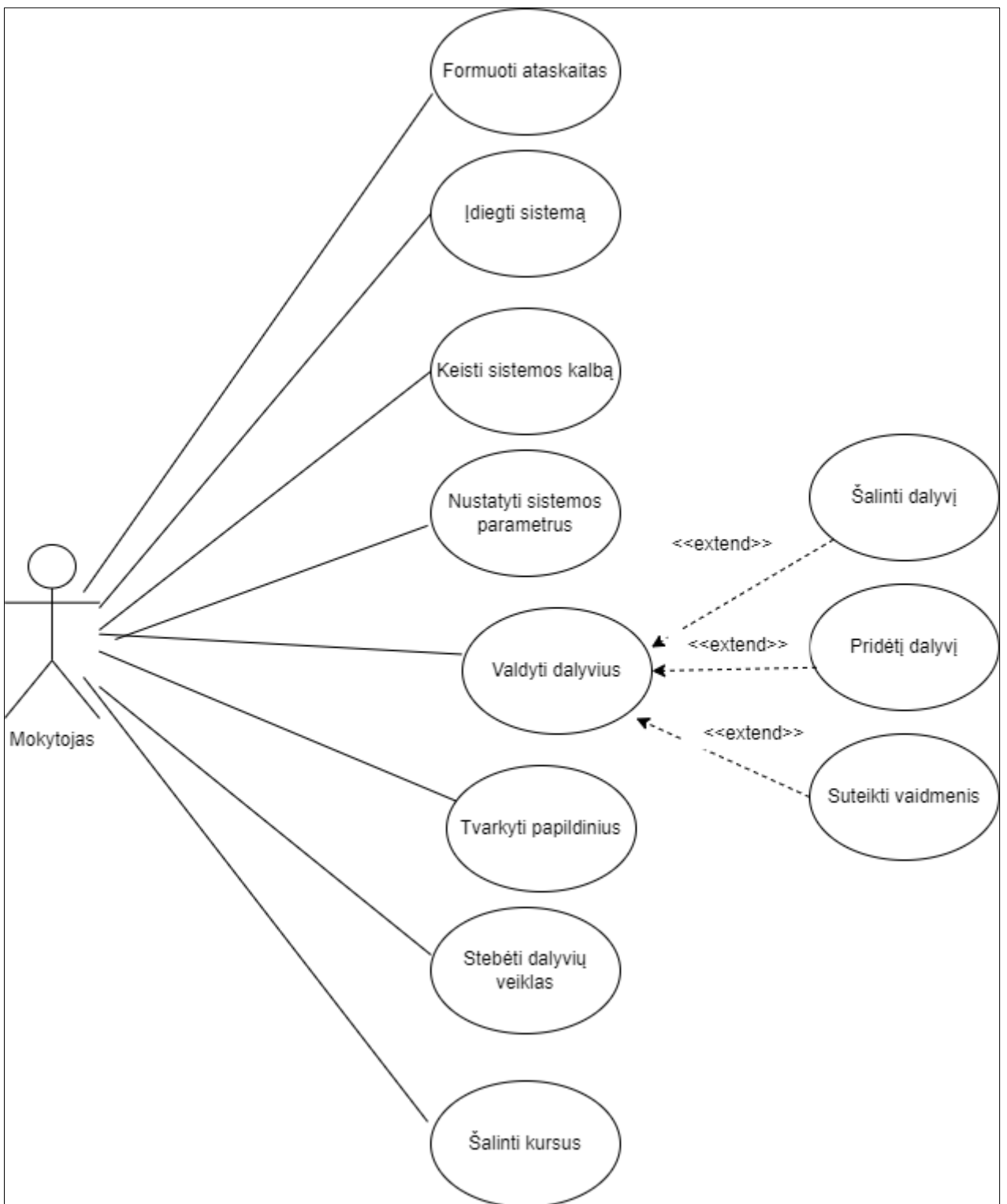
4 lentelė. VMA dalyvių posistemės

Posistemė	Dalyviai			
	Administratorius	Dėstytojas	Besimokantysis	Svečias
Administravimo posistemė	Diegti sistemą ir ją valdyti Nustatyti sistemos parametrus Keisti sistemos kalbą Paringti reikalingus įskiepius Tvarkyti dalyvių profilius Pridėti, šalinti dalyvius Suteikti dalyviams vaidmenis Stebėti naudotojų veiklas Šalinti kursus Siųsti priminimus apie artėjančias ar numatytas veiklas	Kurti, šalinti kursą Registruoti dalyvius Priminti apie artėjančias veiklas Pildyti kalendorių	Skaityti mokomąją medžiagą Parsisiųsti medžiagą Atsispausdinti medžiagą	
Kursų kūrimo ir valdymo posistemė	Kurti kursą Prižiūrėti kursą, šalinti kursą,	Redaguoti kurso duomenis, pateikti kurso užklausa, kurti kursą, valdyti	Prisijungti prie kurso, susipažinti su mokomąja medžiaga ir	Prisijungti prie kurso

	redaguoti kurso parametrus, įtraukti dalyvius, redaguoti kurso veiklas, Suteikti naudotojams roles	naudotojus, įtraukti dalyvius, šalinti dalyvius.	dalyvauti veiklose	
Mokymosi turinio pateikimo posistemė	Įkelia mokomąją medžiagą, redaguoja mokomąją medžiagą, ištrina mokomąją medžiagą, peržiūri mokomąją medžiagą, parsisiunčia, išsisaugoja ir atsispausdina mokomąją medžiagą.	Kuria temas kurse, įkelia mokomąją medžiagą, redaguoja pateikia mokomąją medžiagą atsisiuntimui ir spausdinimui.	Peržiūrėti mokomąją medžiagą, Parsisiųsti mokomąją medžiagą, atsispausdinti mokomąją medžiagą.	Peržiūri mokomąją medžiagą.
Bendravimo ir bendradarbiavimo posistemė	Rašyti žinutę Gauti žinutę Atsakyti žinutę Skaityti žinutę, Išsiųsti žinutę, Trinti žinutę, Ieškoti žinučių, Atsakyti forume žinutę, Kurti forumo temas, jas tvarkyti, šalinti, Ieškoti forumo temų,	Rašyti žinutę Gauti žinutę Atsakyti žinutę Skaityti žinutę, Išsiųsti žinutę, Trinti žinutę, Ieškoti žinučių, Atsakyti forume žinutę, Kurti forumo temas, jas tvarkyti, šalinti, Skaityti forumo temas, jas redaguoti, šalinti, ieškoti forumo temų, pranešti dalyviams apie naujas forumo temas.	Rašyti žinutę Gauti žinutę Atsakyti žinutę Skaityti žinutę, Išsiųsti žinutę, Trinti žinutę, Ieškoti žinučių, Atsakyti forume žinutę, Kurti forumo temas, jas tvarkyti, šalinti, Skaityti forumo temas, jas redaguoti, šalinti, ieškoti forumo temų.	Skaityti forumą ir jo temas.

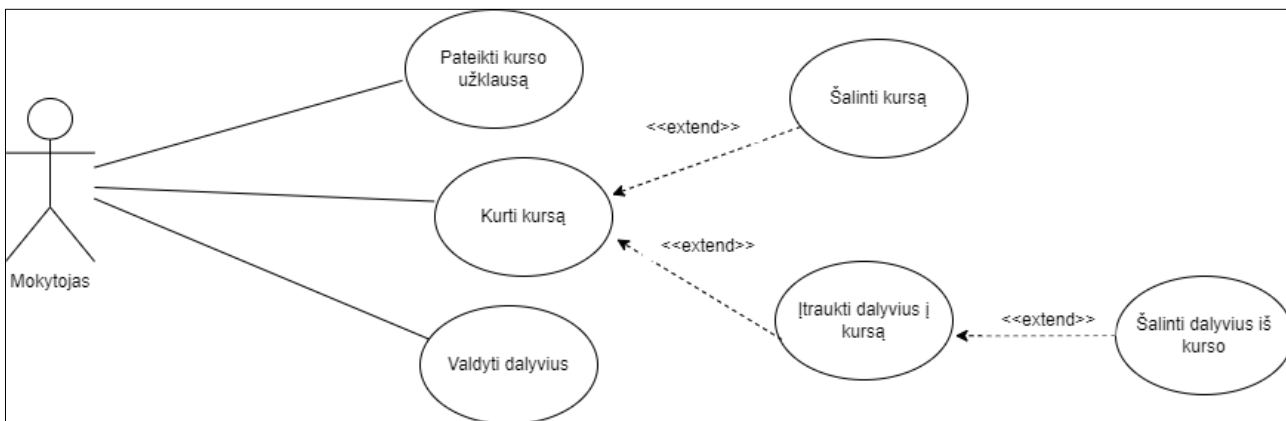
Panaudojimo atvejų diagramos ir atvejų specifikacijos

Administravimo posistemyje (6 pav.) galima diegti sistemą, nustatyti sistemos parametrus, valdyti dalyvius, kursus, formuoti ataskaitas, pašalinti kursus.



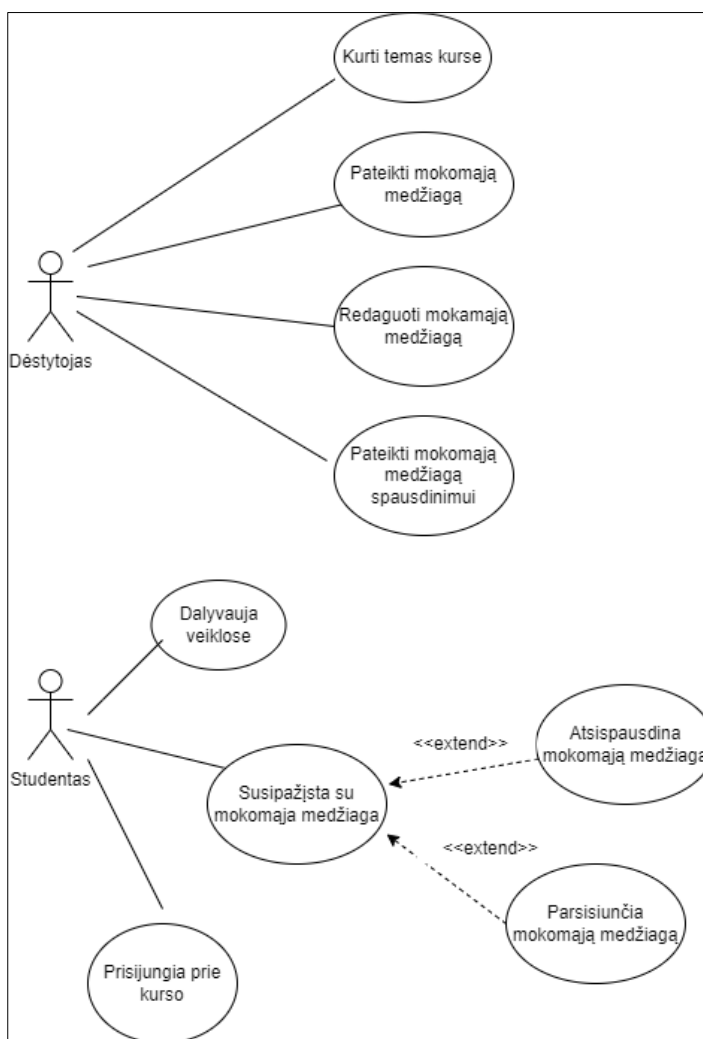
6 pav. Administratoriaus PA diagrama administravimo posistemyje

Kurso valdymo ir kūrimo (7 pav.) posistemis įgalina pateikti kurso kūrimo užklausą, kurti kursą, valdyti kursą, jo dalyvius;



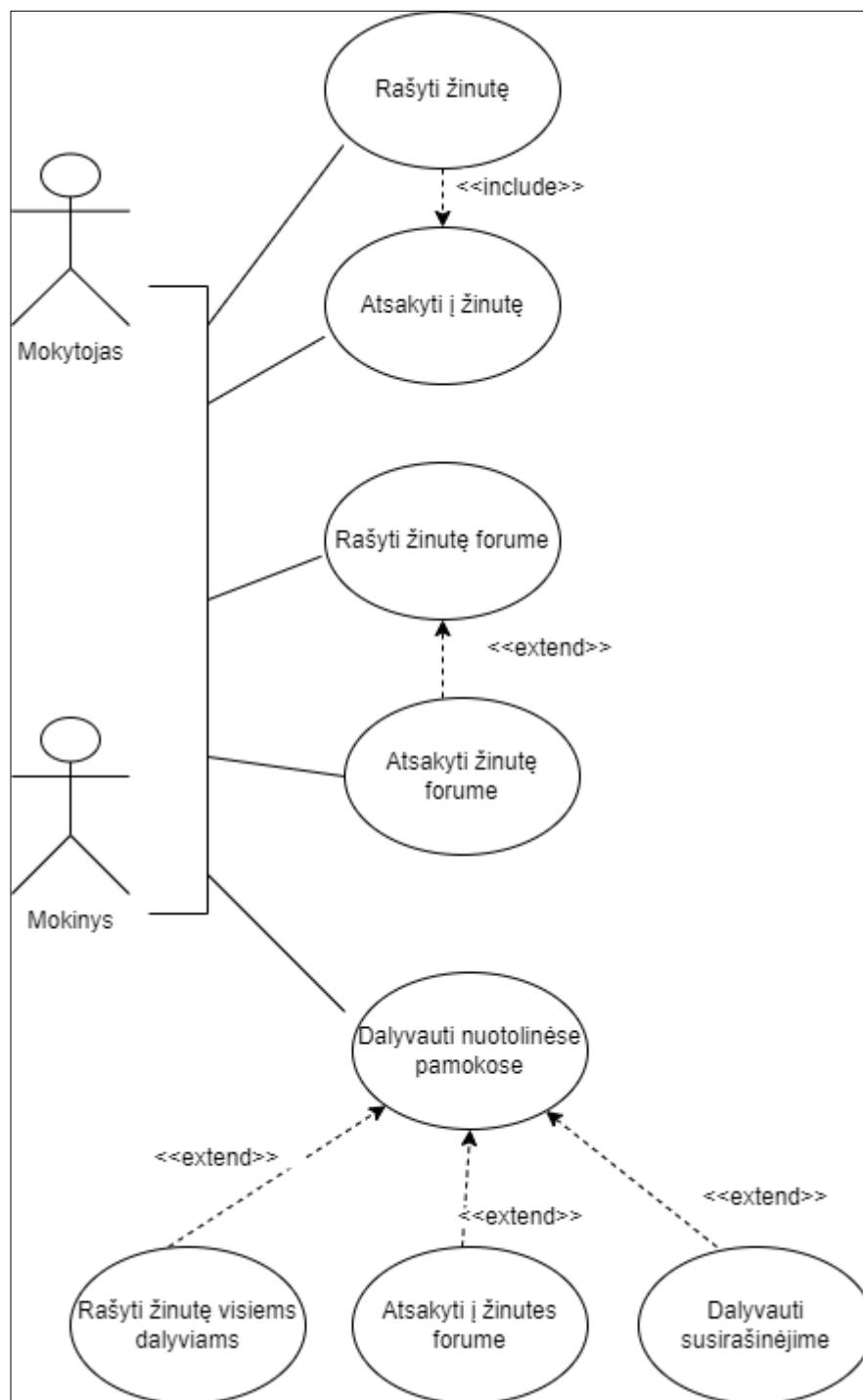
7 pav. Mokytojo PA diagrama kurso kūrimo posistemyje

Turinio pateikimo posistemis (8 pav.) skirtas mokymosi medžiagos pateikimui bei modifikavimui, mokymosi medžiagos peržiūrėjimui, atsisiuntimui ar atsispausdinimui, jo keliami funkciniai reikalavimai skirstomi į mokytojo ir mokinio vaidmenis.



8 pav. Mokymo turinio rengimo ir teikimo posistemės panaudojimo atvejų diagrama

Bendravimo ir bendradarbiavimo posistemyje (9 pav.) galima rašyti, skaityti žinutes, į jas atsakyti, dalyvauti diskusijų forumuose, dalyvauti vaizdo konferencijose, atlikti užduotis bendradarbiaujant.



9 pav. Bendravimo ir bendradarbiavimo posistemės panaudojimo atvejų diagrama

2.5. Programinės įrangos parinkimas VMA realizuoti

Mokymosi procesas sudėtingas ir daug pastangų ir darbo reikalaujantis procesas. Skaitmeninį mokymosi turinį galima perduoti įvairiais būdais. Dėstytojas privalo parinkti patogiausią ir geriausią medžiagos perdavimo priemonę. Siekiant parinkti tinkamą mokymosi valdymo sistemą kuriamai programavimo mokymosi aplinkai, buvo įdiegtos ir išbandytos pagal sudarytą projektuojamos VMA (ir jos posistemių) funkcionalumą dvi populiaros VMA: Moodle ir Openeclass. Abi sistemos atitinka

pagrindinius VMA keliamus funkcinis reikalavimus, jose yra realizuoti pagrindiniai įrankiai ir priemonės mokymui, mokymuisi, bendravimui ir bendradarbiavimui, tačiau abi sistemos turi ir nemažai skirtumų, kurių pagalba galima išskirti, kuri aplinka yra tinkamesnė pritaikyti kuriamai aplinkai (žr. 5 lentelę).

5 lentelė. VMA palyginimas

VMA posistemis	Funkcinės specifikacijos	MOODLE	OPENECLASS
Administravimo	Diegti sistemą ir ją valdyti	+	+
	Nustatyti sistemos parametrus	+	+
	Keisti sistemos kalbą	+	-
	Paringti reikalingus įskiepius	+	-
	Tvarkyti dalyvių profilius	+	+
	Pridėti , šalinti dalyvius	+	+
	Suteikti dalyviams vaidmenis	+	+
	Stebėti naudotojų veiklas		
	Šalinti kursus	+	+
	Siųsti priminimus apie artėjančias ar numatytas veiklas	+	+
		+	+
		+	+
Kurso valdymo	Kurti kursą	+	+
	Prižiūrėti kursą,	+	+
	šalinti kursą,	+	+
	redaguoti kurso parametrus,	+	+
	įtraukti dalyvius,	+	+
	redaguoti kurso veiklas,	+	+
	Suteikti naudotojams roles	+	+
Turinio pateikimo	Atskiri tinklalapiai	+	+
	Susietų tinklalapių rinkiniai	+	+
	Minčių žemėlapis	+	+
	Apklausa	+	+
	Vaizdo ir garso įrašai	+	+
	Pateiktys	+	+
	SCORM paketų naudojimas	+	-
Mokymosi proceso organizavimo ir vertinimo	Kurti kurse temas	+	+
	Įkelti mokomąją medžiagą	+	+
	Redaguoti mokymosi medžiagą	+	-
	Ištrinti mokymosi medžiagą	+	+
	Skaityti medžiagą	+	+
	Parsisiųsti medžiagą		

3. Programavimo mokymui(si) skirtos aplinkos modeliavimas

Informatikos mokymas bendro lavinimo mokyklose ugdo informatinį mokinių mąstymą ir skaitmeninį raštingumą įvairiose klasėse. Programavimas skatina ne tik kūrybiškumą, bet ir ugdo kritinį mąstymą bei padeda spręsti realias problemas. Programavimo mokymui(si) skirtos aplinkos gali būti naudojamos:

- mokyti ar mokytis programavimo pradmenų arba tobulinti esamus įgūdžius;
- mokytojai gali naudoti aplinką kaip pagalbines priemones mokymo procesui organizuoti;
- aplinką galima naudoti bendradarbiavimui;
- mokytis programavimo savarankiškai;
- virtualios mokymo platformos gali naudoti aplinką kaip mokymo įrankį kursams suteikdami mokiniams prieigą prie kokybiškos medžiagos bei praktinių užduočių;
- aukšto lygio programuotojai gali naudoti aplinką kaip priemonę tobulinti savo įgūdžius.

3.1. Aplinkos modeliavimas

Ontologija apie programavimo mokymosi aplinką nusako svarbias sąvokas, pabrėžiant jų svarbą programavimo mokymosi procese. Tai padeda naudotojams lengviau orientuotis ir rasti tinkamas mokymosi priemones bei išteklius.

Programavimo mokymui(si) skirtos aplinkos ontologija (10 pav.) atvaizduoja informatikos III-IV gimnazijos klasių informatikos mokymo(si) turinį, programavimo mokymui(si) skirtas platformas, programavimo redaktorių, mokomąją medžiagą, mokymosi resursus.

Programuoti galima įvairiose aplinkose. Galima rinktis teksto redaktorių, internetines platformas.

Programavimo aplinkos pasirinkimui pasirinktas požymiais grįstas modeliavimas (11 pav.). Požymių diagrama pateikia išsamų programavimo mokymui(si) skirtų aplinkų vaizdavimą. Diagrama padeda suprasti kaip kiekviena aplinka yra pritaikyta mokymui(si) programavimo srityje. Iš diagramos matyti įvairūs požymiai, tokie kaip mokymo išteklių, aplinkos paieška, aplinkų tipai, grįžtamasis ryšys.

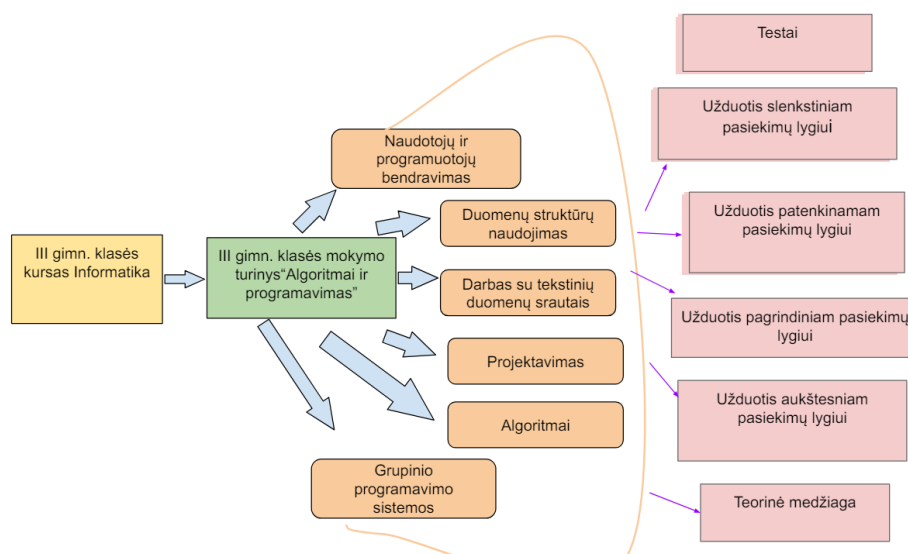


10 pav. Kuriamos aplinkos ontologija

- [-] Programavimo_mokymuisi_skirtos_aplinkos
- [-] ● Mokymosi_resursai
 - Skaitmenines_knygos
 - Mokomoji_medziaga
 - Nuotolines_pamokos
- [-] ● Aplinkos_paieska
- [-] ○ Mokinia_mokymosi_ypatumai
 - Motyvacija
 - Individualumas
 - Socialiniai_veiksniai
 - Priemoniu_panaudojimas
 - Virtualios_mokymosi_aplinkos
- [-] ● Programavimo_kalbos
- [-] ▲ [1..*]
 - Ci_plus_plus
 - Python
 - Java
 - Java_Script
- [-] ● Programavimo_mokymo_aplinku_tipai
- [-] ○ Internetines_platformos
- [-] ▲ [1..*]
 - Idle
 - Joodle
 - OnlineGDB
 - repl_it
 - Kitos
- [-] ○ Idiegiamos_programavimo_aplinkos
- [-] ▲ [1..*]
 - CodeBlocks
 - Visual Studio Code
 - Dev_C_plus_plus
 - Eclipse
 - Kitos
- [-] ▲ [1..*]
- [-] ○ Programu_kodo_redaktorai
- [-] ▲ [1..*]
 - Notepad_plus_plus
 - Atom
 - Sublime_text
 - Kiti
- [-] ● Programavimo_mokymo_budai
- [-] ▲ [1..*]
 - Grupiniai_projektai
 - Pamokos
 - Savarankiskas_mokymasis
 - Mentoryste
 - Kiti
- [-] ● Mokymosi_turinys
- [-] ● Pradedantiesiems
- [-] ● Pazengusiesiems
- [-] ● Vertinimas
- [-] ▲ [1..*]
 - Ilgalaikiai_projektai
 - Kaupiamasis_vertinimas
 - Kontroliniai_darbai
- [-] ○ Testai
- [-] ▲ [1..*]
 - Atviro_tipo
 - Su_pasirenkamaisiais_atsakymais

11 pav. Programavimo aplinkos pasirinkimo požymių diagrama (naudotas įrankis: <http://splot-research.org/>)

Informatikos kurso modelis, kurį sudaro 11 klasės koncentro „Algoritmai ir programavimas“ mokymo(si) turinys pateiktas 12 pav.



12 pav. Informatikos kurso III gimn. kl. mokymo turinys

Aplinkos modelio įgyvendinimui reikia pasirinkti tinkamą įrankį.

6 lentelėje pateikiamas vidurinio ugdymo bendrųjų programų koncentro „Algoritmai ir programavimas“ mokymo(si) turinys, kuriame nurodoma, kokius įgūdžius, žinias ir supratimą turi įgyti besimokantieji.

6 lentelė. III gimn. klasės mokymo turinys ir gebėjimai

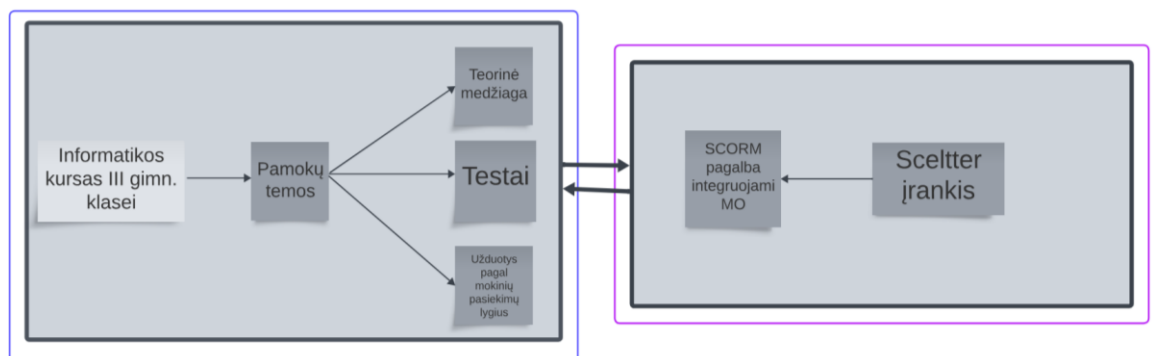
Mokymo turinys	Gebėjimai
<u>Naudotojų ir programuotojų bendravimas.</u>	Aptariamos naudotojų ir programuotojų bendravimo problemos, mokomasi atpažinti lengviau sprendžiamas problemas. Kuriant nedidelius projektus mokomasi rasti potencialių užsakovų (naudotojų), išsiaiškinti duomenų šaltinius, pageidaujamus rezultatus. Projektai gali būti išplėtoti į brandos darbus.
<u>Duomenų struktūrų naudojimas.</u>	Apibrėžiama duomenų tipo sąvoka. Mokomasi naudoti vienmačiu masyvu, tekstine eilute. Kuriamos programos nesudėtingiems skaičiavimams, taikomosioms užduotims spręsti. Taikomi įvairūs duomenų tipai, akcentuojamas tekstinių eilučių ir masyvų naudojimas.
<u>Darbas su tekstinių duomenų šaltais.</u>	Mokomasi skaityti iš tekstinių failų ir į juos rašyti įvairių tipų duomenis. Susipažįstama su atviraisiais duomenų formatais (pavyzdžiui, CSV, HTML, XML, JSON). Mokomasi skaityti duomenis iš failų, interneto svetainių, išorinių įrenginių (pavyzdžiui, Arduino nuoseklaus prievado) analizuojant tekstą ir (ar) taikant paprogramių bibliotekas.
<u>Projektavimas.</u>	Prisimenamas inžinerinis programų projektavimo ciklas. Kuriami nedidelės apimties taikomieji projektai. Juos kuriant aiškinamasi, kokie naudotojo

	sąsajos poreikiai. Piešiama ir pristatoma tekstinės ar grafinės naudotojo sąsajos schema, vėliau ji realizuojama pasirinkta programavimo kalba.
<u>Algoritmai.</u>	Prisimenami sumos, sandaugos, kiekio, vidurkio skaičiavimo, paieškos algoritmai. Susipažinama su rikiavimo, reikšmės paieškos nerikiuotame ir rikiuotame masyve, reikšmių šalinimo iš masyvo, masyvo papildymo naujomis reikšmėmis algoritmais.
<u>Grupinio programavimo sistemos.</u>	Mokomasi darbo pasidalijimo, naudojant atskiras programos dalis (procedūras, funkcijas, klases, modulius, bibliotekas). Pristatomi grupinio programavimo įrankiai (pavyzdžiui, versijų kontrolės sistema git, projektų valdymo sistemos Jira, Trello, vienalaikio programavimo sistema). Mokomasi prisijungti prie draugo kuriamo projekto, skaityti, analizuoti, ieškoti klaidų, modifikuoti ne savo rašytą programos tekstą.

Mokiniam programavimo kurse bus pateikta kiekvienos temos interaktyvi teorinė medžiaga, užduotys pagal pasiekimų lygius ir testai įvertinimui. Slenkstiniam ir pagrindiniam pasiekimų lygiui turintiems mokiniams bus pateiktos užduotys su programos fragmentais. Programas mokiniai mokymosi aplinkoje testuos ir matys rezultatus.

3.2. Programavimo mokymui(si) skirtos aplinkos funkcionalumas

Programavimo mokymui(si) skirta aplinka integruota į VMA Moodle. Programavimo aplinkoje galima testuoti programas ir jas vykdyti. Tokia aplinka labiau motyvuoja mokinius ir daro ją išskirtinę ir patrauklią. Mokymosi procesas nuolat stebėjimas ir vertinamas. Mokinys gali nuolatos stebėti savo pažangą, patikrinti žinias, įvertinti savo silpnąsias ir stipriąsias programavimo mokymosi sritis ir gauti iš mokytojo grįžtamąjį ryšį.



13 pav. Programavimo mokymui(si) skirtos aplinkos struktūra

Iš programavimo mokymui(si) skirtos aplinkos struktūros matyti, kad aplinkoje realizuotos pamokų temos, kuriose įdėtos užduotys, testai, teorinė medžiaga. Sceltter įrankio pagalba panaudotas SCORM paketas ir integruoti mokymosi objektai (13 pav.).

Mokymo proceso organizavimas

Naudojantis šiuo programavimo mokymui(si) skirtu kursu, mokiniai gali:

- Susipažinti su teorine medžiaga;
- Nagrinėti programų pavyzdžius;
- Savarankiškai atlikinėti užduotis ir tokiu būdu pasitikrinti žinias;
- Atlikti testus ir gauti įvertinimą;
- Stebėti pažangą;
- Programuoti mokymo(si) aplinkoje;
- Analizuoti kodo klaidas.

Šiame kurse mokiniai gali mokytis savarankiškai, užpildyti žinių spragas, pasiruošti egzaminui.

Susipažinę su teorine medžiaga, mokiniai gali savarankiškai atlikti testus, iš karto mato savo įvertinimą. Taip pat mokiniai VMA *Moodle* gali rašyti programas bei jas vykdyti ir testuoti. Programavimo mokymui(sio) skirta aplinka mokiniai gali naudotis pamokoje, bibliotekoje, namuose, t.y. mokytis iš bet kurios pasaulio vietos, kur yra internetas. Naudodami šią programavimo mokymui(si) skirtą aplinką, mokiniai gali bendrauti tarpusavyje ir bendradarbiauti, diskutuoti kurso forume.

3.3. Skyriaus išvados

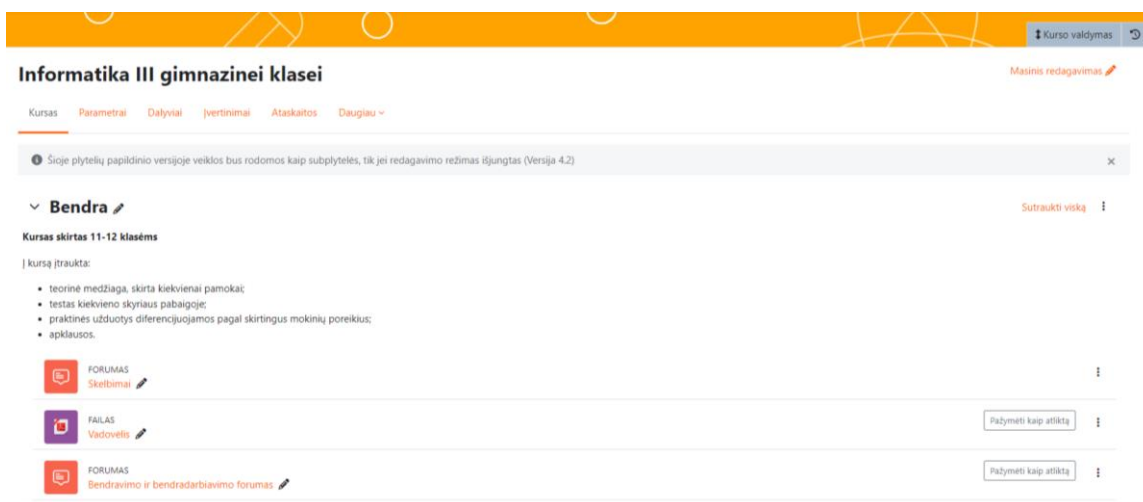
1. Sukurtas programavimo mokymui(si) skirtos aplinkos modelis. Šis modelis, kuris pritaikytas skirtingų mokymosi pasiekimų lygių mokiniams, užtikrina ir suteikia galimybę mokiniams mokytis savo tempu ir pagal savo žinių lygį, o tai skatina gilesnį žinių supratimą ir įsisavinimą. Programavimo mokymui(si) skirtoje aplinkoje užduotys pateikiamos pagal mokinių pasiekimų lygius, o tai suteikia galimybę mokytojams suteikti pagrindines žinias kiekvienam mokiniui ir tai padeda jiems greičiau ir efektyviau įsisavinti medžiagą. Šis modelis pasižymi inovatyviomis ir efektyviomis mokymo strategijomis, kurios padeda individualizuoti mokymą, didina mokinių motyvaciją, įsitraukimą ir užtikrina kokybišką žinių įsisavinimą. Šis modelis prisitaiko ne tik prie kiekvieno mokinio poreikių, bet ir ugdo svarbius socialinius ir profesinius įgūdžius, taip prisidedant prie mokinių tobulėjimo.
2. Užduotims kurti pasirinktas Shelter įrankis. Naudojant interaktyvius įrankius, mokiniai labiau įsitraukia į mokymo procesą ir yra labiau motyvuoti siekti geresnių rezultatų. Shelter įrankis yra labai naudingas diegiant ir valdant programavimo mokymosi aplinką, nes jis suteikia lengvą diegimą, saugumą, patikimumą, prisitaikymą prie mokymosi valdymo sistemų, kadangi įdiegiamas naudojant *SCORM* mokymo objektus, įvairias mokymo priemones ir profesionalią techninę pagalbą. Šie privalumai padeda sukurti efektyvią ir kokybišką mokymosi aplinką, kuri atitinka skirtingų mokymosi pasiekimų lygių mokinių poreikius ir skatina jų sėkmingą mokymąsi.

4. Kurso kūrimas VMA Moodle ir jos dalyviai

Praėjusiame skyriuje aprašius ir palyginus dvi virtualaus mokymo aplinkas ir išnaginėjus jų funkcinius ir nefunkcinius reikalavimus galima daryti išvadą, kad programavimo mokymui, mokymosi proceso organizavimui, bendravimui ir bendradarbiavimui geriausiai tinka VMA Moodle. Šioje mokymo aplinkoje užduotys bus kuriamos naudojant *Sheltter* įrankį ir integruojat jį į VMA SCORM paketą. Šiame skyriuje nagrinėjama, kaip tai galima praktiškai panaudoti. Analizuojama, kaip šią technologiją galima pritaikyti mokantis programuoti.

Programavimo mokymosi aplinkos integravimui buvo pasirinkta Elektrėnų savivaldybės Vievio gimnazija, o programavimo mokymosi aplinkos kursas sukurtas šios įstaigos VMA Moodle, kurios adresas yra <https://vieviogimnazija.vma.lm.lt/>.

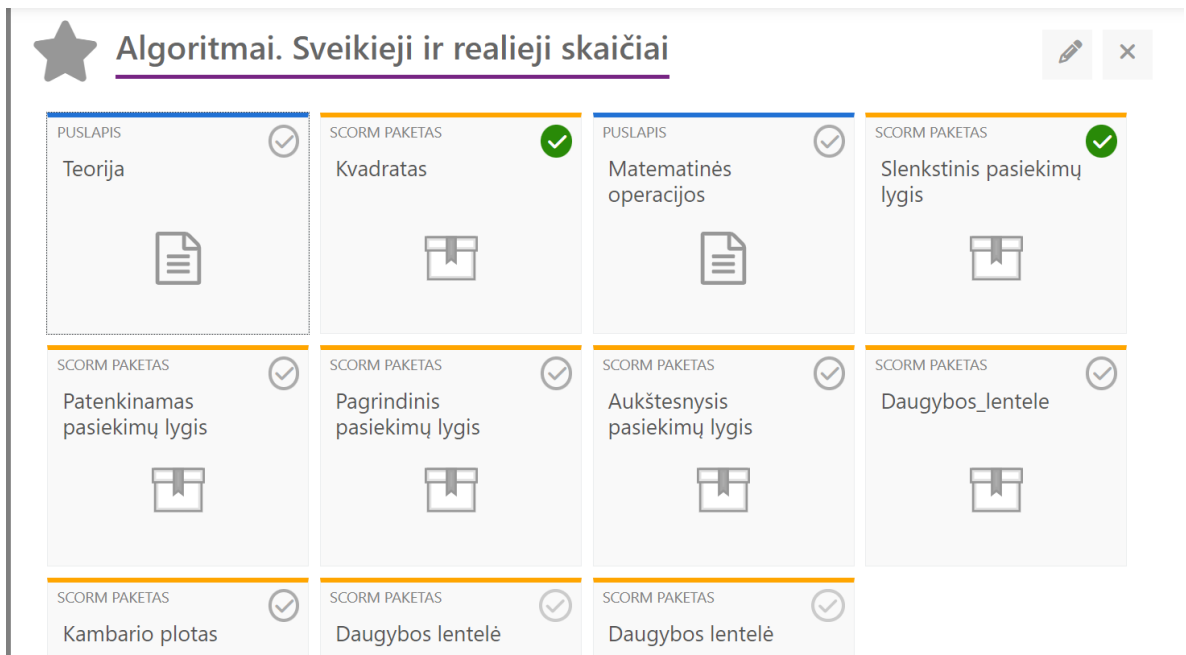
VMA Moodle kursą gali sukurti šios aplinkos naudotojas, kuris pateikia kurso užklausą. Moodle administratorius patvirtina užklausą ir naudotojui sukuriamas kursas ir jam suteikiamas redaguojančio dėstytojo vaidmuo. Kursą taip pat gali sukurti administratorius ir įtraukti kurso dalyvius į sukurtą kursą ir suteikti jiems reikiamas teises (14 pav.).



14 pav. Sukurtas kursas „Informatika III gimnazinei klasei“

Kurso kūrėjas yra vienas iš kurso dalyvių, kuriam dažniausiai priskiriamas dėstytojo (mokytojo) vaidmuo. Mokinį į kursą registruoja mokytojas (Kurso administravimas – Dalyviai - Įregistruoti naudotojus).

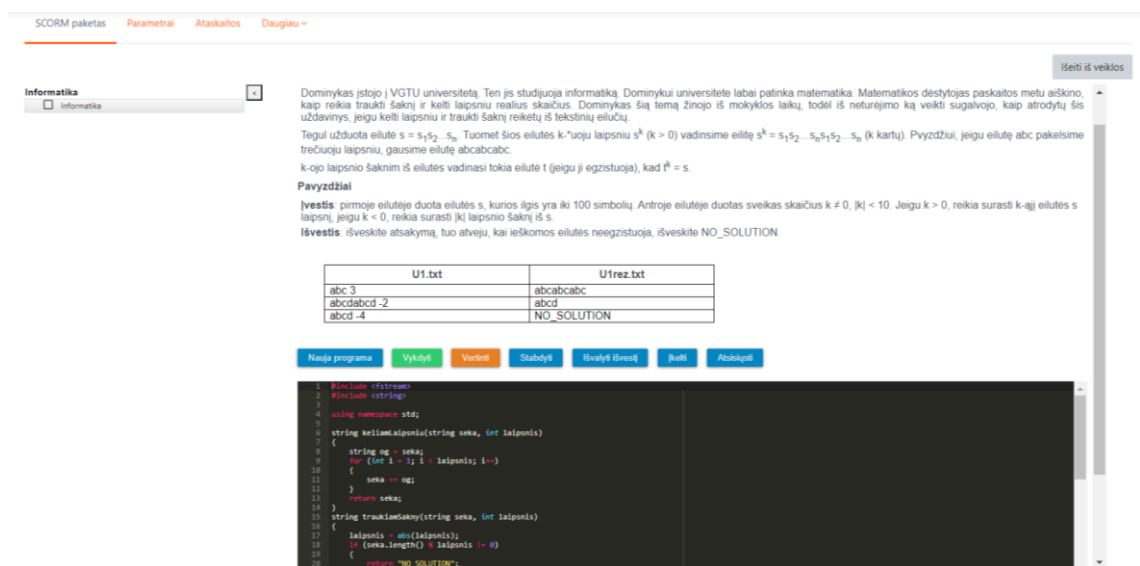
VMA Moodle kurse mokymosi medžiaga pateikiama struktūrizuotai, kuri yra suskirstyta į temas. Kurso pagrindiniame puslapyje esančios temos pateikiamos plytelės formatu (13 pav.). Kiekviena tema turi išskleidžiamą meniu, kuriame pateikiamos užduotys pritaikytos skirtingų poreikių mokiniams, testai ir teorinė medžiaga. Kiekvienoje temoje yra teorinė medžiaga, užduotys, kurios pritaikytos skirtingų poreikių mokiniams, testai (15 pav.).



15 pav. Vienos temos struktūra VMA Moodle kurse

Užduotys, esančios VMA Moodle kurse pateikiamos naudojant *Scorm* paketą. Šis išteklius leidžia įkelti programavimo mokymui skirtą aplinką, kuri sukurta panaudojus *Shellder* įrankį www.arzina.lt. Šis išteklius leidžia pateikti tiek programavimo mokymui skirtą aplinką, tiek teorinę medžiagą, kurioje galima kompiliuoti pavyzdines užduotis.

Mokiniai, atsidarę užduotį gali atlikti programavimo užduotį pačioje VMA Moodle, kurioje yra integruota programavimo aplinka (16 pav.).

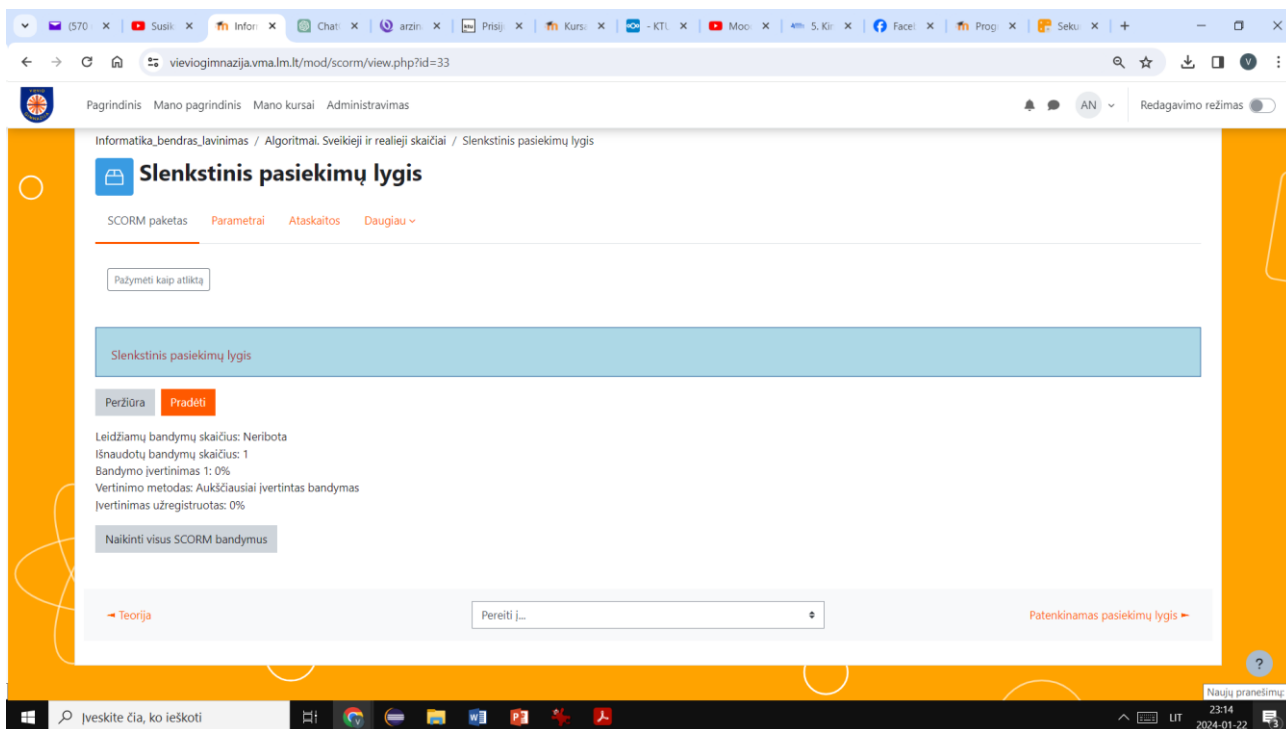


16 pav. Integruota programavimo mokymo aplinka

Kiekvienoje temoje pradžioje pateikiama teorinė medžiaga, toliau pateikiamos užduotys, kurias mokytojas paskiria pagal pasiekimų lygį, testas ir užduotys, kurias mokiniai gali atlikti savarankiškai. Atliktas užduotis mokytojas mato pačioje VMA Moodle sistemoje.

4.1. Užduotys pagal mokinių pasiekimų lygius

VMA Moodle kurse mokymosi medžiaga suskirstyta į temas. Kiekvienoje temoje yra teorinė medžiaga ir keturių lygių užduotys, kurios pritaikytos skirtingų poreikių mokiniams (17 pav.).



17 pav. Vieno pasiekimų lygio užduotis VMA Moodle kurse

Pamokos užduotys VMA Moodle kurse pateikiamas naudojant išteklių „Scorm paketas“. Šis išteklius leidžia pateikti programavimo aplinką SCORM formatu.

Kiekviename pasiekimų lygyje, mokiniai gali atlikti užduotis pagal savo gebėjimus.

4.2. Integruojamos programavimo mokymui(si) skirtos aplinkos įrankio „Shellter“ aprašas

Bendra apžvalga

Kadangi, panaudojimo atvejų yra daug, tai pradžioje pateikta įrankio vartotojo sąsaja. Pradžioje įvedamas kurso pavadinimas, toliau – užduoties pavadinimas. Toliau galima įrašyti trumpą užduoties pavadinimą, kuris yra skirtas apps'ui. Mokymo sistemoje jo nesimatys.

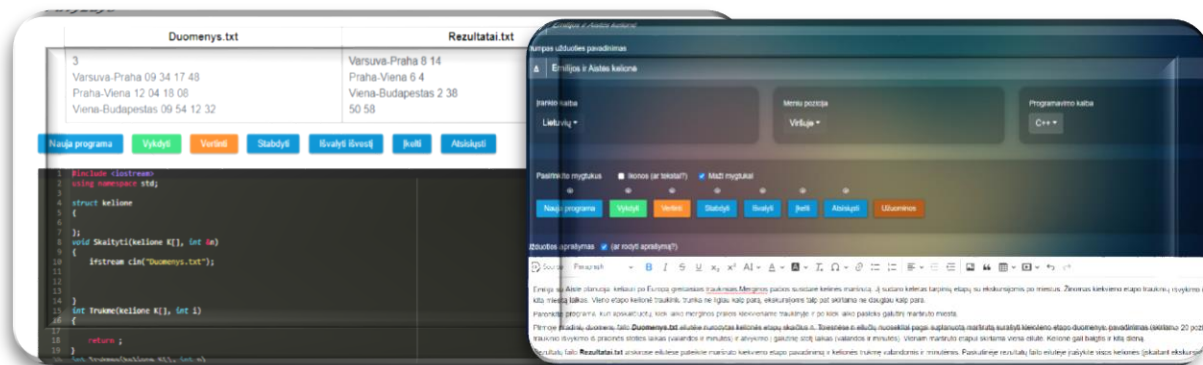
Įrankio kalba. Įrankio kalbą galima sugeneruoti ne tik lietuvių. Ateityje jų prisidės daugiau.

„Meniu pozicija“ patogiu, kad galima skirti viršutinį išdėstymą. Uždaviniams patogiu skirti viršutinį išdėstymą, o pamokoms – kairiąją poziciją. Galima pateikti užduoties aprašymą, taip pat įvesties ir išvesties pavyzdžius. „Pradinis programos skeletas“ – čia galima įkelti kodo fragmentą, patį kodą ar tuščią lauką, kur mokinys mokymo sistemoje turėtų rašyti savo kodą. Būtina nurodyti kūrėjo vardą ir pavardę. Užduoties ID būtina parašyti ir šis neturi kartotis kitose užduotyse arba kituose teoriniuose fragmentuose./ ID galima įvesti ranka arba, dešinėje pusėje paspaudus mygtuką, jį sugeneruoti. Galima kurti tekstinius failus, taip pat testus, kurie tikrintų užduotį, Galima sugeneruoti paketą, tačiau prieš tai rekomenduotina ištestuoti paketą. Pažiūrėti, ar gerai išdėstyti mygtukai, ar teistai gerai veikia ir pan. Šioje testavimo aplinkoje galima nustatyti mygtukų prisitaikymo aukštį, tuomet būtina atnaujinti objektą. Jeigu tai yra pamokai skirtas kodas, galima tiesiog kopijuoti

nuorodą ir įsikelti į mokymo sistemą. Mygtukus galima įjungti ir išjungti. Išvalyti – išvalo išvedimo lauką. Vykdomasis langas – prisitaikantis. „Įveikimo procentas“ – tai kai mes susisiejam su mokymosi aplinka ir reikia pranešti ar užduotis įvykdyta, tai reikia nurodyti kiek ji turi būti įvykdyta procentais.

Taip pat šis įrankis turi dirbtinio intelekto asistentą. Parašius programą, jeigu joje yra sintaksės ar vykdymo klaidų, tuomet galima paprašyti dirbtinio intelekto asistento ištaisyti klaidas. Bus pateikta ištaisyta programa. DI pateikia savo atsiliepimus ką galima programoje tobulinti ir kas yra gerai.

Shelster – naujas įrankis. Jis skirtas mokytis C++ ir Phyton programavimo kalbų. Turi dvi aplinkas, t.y. vykdomąją aplinką ir kūrimo aplinką, kurioje galima sukurti programinius intarpus ir uždavinius (18 pav.).

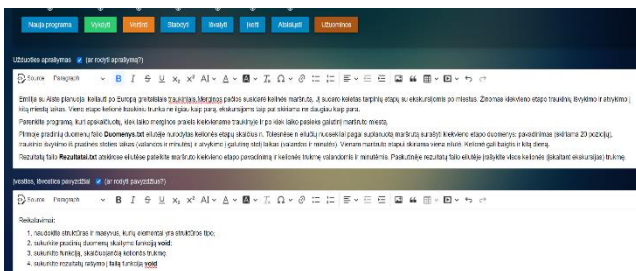


18 pav. Vykdomoji ir kūrimo aplinkos

Šis įrankis leidžia vykdyti C++ ar Phyton ‘o kodą naršyklėje. Šio įrankio nereikia įdiegti į serverius ir dėl to mažiau darbo tenka administratoriams ir tuo pačiu vykdytojui, nes nereikia atlikinėti įvairių keitimų. Šis įrankis įtraukia failų operacijas, nes pagrindinis trūkumas, kad mokiniai negali mokytis tų dalykų, kurių jiems reikės. Galima kurti ne tik programavimo uždavinius, bet ir programavimo kodo ištraukas „Snipetus“, kuriuos galima dėti į teorinę medžiagą ir iliustruoti tam tikrų algoritmų veikimą. Pagrindinis tikslas yra programavimo uždaviniai ir jų testavimas. Šis įrankis leidžia saugoti programas ir taip pat teikia rezultatus mokymosi sistemoms, pvz.: Moodle. Taip pat šiame įrankyje yra DI (dirbtinio intelekto) asistentas, kuris paaškina atsiradusias klaidas, kurios gali būti tiek vykdymo metu pastebėtos, tiek ir sintaksės klaidos ir kartu šis asistentas ištaiso kodą, jeigu tai leidžiama Kad mokyti būtų smagiau, galima pasirinkti „numestukus“, angl. „Drobs“, kurie yra skirti žaidybiniui. Galima organizuoti programavimo varžybas be jokios mokymosi sistemos. Dar yra įgyvendintos užuominos, kurios yra programavimo vedlys. Jos pataria kaip pradėti spręsti uždavinį ir laikas nuo laiko pateikia naujų detalių.

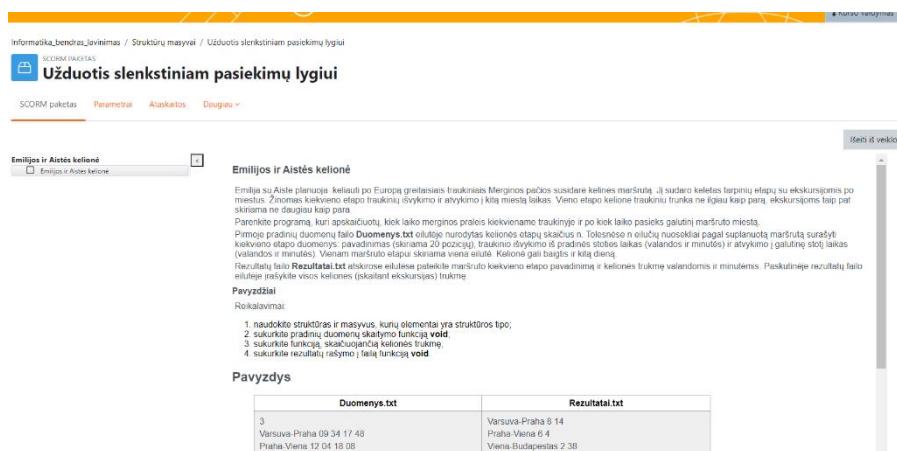
Publikavimas

Sukurtą uždavinį vieno mygtuko paspaudimu galima publikuoti www.arzinai.lt serveryje ir ta nuoroda dalintis. Be nuorodos sukurtus uždavinius galima įterpti į kitą turinį naudojant „įterptinį kodą“ (19 pav.).



19 pav. Uždavinių publikavimas

Patogu naudoti norint į mokymo sistemą Moodle įterpti mokomąją medžiagą. Yra palaikomas Scorm paketas. Šis įrankis generuoja Scorm paketą. Šį paketą galima įkelti beveik į kiekvieną mokymosi aplinką, kuri palaiko Scorm paketo standartus (20 pav.).

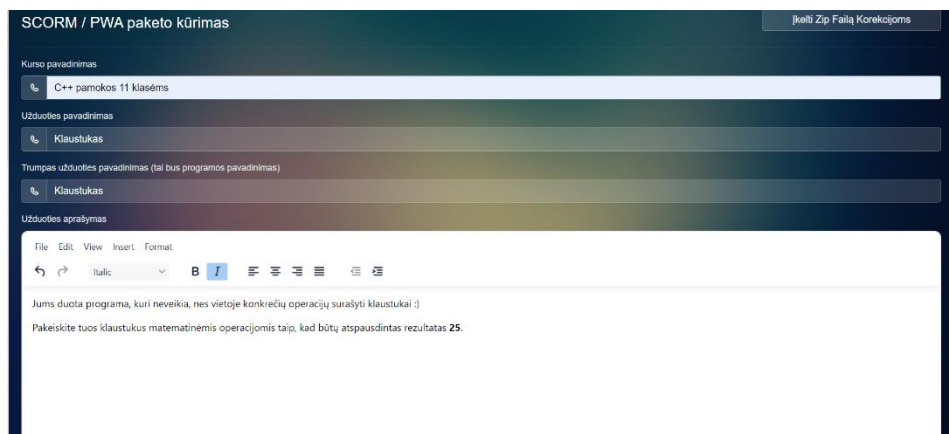


20 pav. Užduotis įkelta naudojant SCORM paketą

4.3. Interaktyvus mokymosi turinio pateikimas SCORM

Atliekant užduotis pagal pamokos temą, reikalinga naudoti programavimo aplinką. Tam numatyta naudoti SCORM paketą.

SCORM paketas – tai yra standartas, skirtas turinio kūrimui ir perdavimui. SCORM apibrėžia, kaip turi būti sukuriamas ir perduodamas elektroninis mokymosi turinys. Naudojant SCORM paketą mokiniai mokymosi aplinkoje galės rašyti kodus ir juos kompiliuoti, taip pat gauti įvertinimus. Užduotis galima kurti www.arzina.lt/scorm/ aplinkoje ir jas įkelti į VMA Moodle mokymosi aplinką (21 pav.).



21 pav. Užduočių kūrimas naudojant SCORM paketo kūrimo aplinką

4.4. Sukurtos mokymosi aplinkos vertinimas

Programavimo mokymui(si) skirta aplinka įdiegta Vievio gimnazijoje (žr. 6 priedą). Ją išbandė III gimnazinės klasės mokiniai, kurie pasirinkę informatikos dalyką. Norint įvertinti įdiegtos programavimo mokymui(si) skirtos aplinkos galimybes, privalumus, trūkumus, naudingumą bei tobulinimo galimybes, mokiniams buvo pateiktas klausimynas (žr. 5 priedą).

Iš viso buvo apklausti 5 mokiniai, iš kurių 4 (80 %) vaikinai ir 1 (20 %) mergina. Buvo *sukurtas VMA Moodle* kursas, kuriame pateiktos užduotys pagal skirtingus mokinių mokymosi poreikius. Į klausimą kiek laiko mokiniai naudojami šia aplinka, 100 % mokinių atsakė, kad aplinka naudojasi metus laiko.

100% apklaustųjų atsakė, kad naudojantis *VMA Moodle* problemų nekyla, viskas aišku ir konkrečiu. Kadangi į mokymosi sistemą buvo integruota programavimo mokymui(si) skirta aplinka, todėl į klausimą „Kiek laiko naudojate programavimo aplinką integruotą į *VMA Moodle*“, 60 % mokinių atsakė, kad programavimo aplinką naudoja metus laiko ir 40 % atsakė, kad naudoja pusę metų.

Mokinių buvo klausama apie programavimo mokymui(si) skirtą aplinką, kuri integruota į *VMA Moodle* ir ar programavimo mokymui(si) skirta aplinka atitinka jų poreikius. 80 % mokinių mano, kad programavimo mokymui(si) skirta aplinka visiškai atitinka jų poreikius, tačiau 20 % mano, kad atitinka tik iš dalies.

Mokinių paklausus, kokių funkcijų trūksta programavimo mokymui(si) skirtoje aplinkoje, kad ji būtų patogesnė ir funkcionalesnė, iš penkių galimų atsakymų buvo pateikti du atsakymai, jog netrūksta nieko ir viskas tinka.

Net 80 % respondentų atsakė, kad programavimo mokymui(si) skirta aplinka suteikia motyvacijos ir paskatinimo siekti tikslo ir tik 20 % mano, kad nei suteikia, nei nesuteikia.

Mokiniai taip pat vertino ir *VMA Moodle* pateiktas užduotis. Daugelis apklaustųjų, t.y. 80 % teigia, kad pamokų skaičius atitinka pamokos trukmę ir 20 % mano, kad pateikiama per daug užduočių.

Mokinių buvo klausama, kaip jie vertina užduoties struktūrą. Dauguma atsakiusiųjų, 80 % mano, kad užduotys suformuluotos aiškiai ir konkrečiai ir 20 % mokinių nuomone, kad užduotyse trūksta aiškumo.

VMA Moodle aplinkoje be užduočių yra pateikiama trumpa teorinė medžiaga apie kiekvieną temą. Paklausus mokinių, gal jie norėtų *VMA Moodle* kurse daugiau teorinės medžiagos, mažiau nei pusė (40 %) sako, kad jiems įdomios tik praktinės užduotys, tokia pat dalis (40 %) mano, kad teorinės medžiagos kurse pateikta užtektinai ir tik maža dalis (20 %) teigia, kad teorinės medžiagos kurse galėtų būti daugiau.

Siekiant išsiaiškinti, kokias užduotis mokiniai renkasi pagal pasiekimų lygius, 80 % mokinių renkasi užduotis, kurios atitinka aukštesnįjį pasiekimų lygį ir 20 % mokinių renkasi slenkstinį pasiekimų lygį.

Paklausus, ar mokiniams patinka užduotys pagal skirtingus mokinių pasiekimų lygius, 80 % atsakė, kad patinka, 20 % atsakė, kad patinka iš dalies.

100% apklaustųjų teigia, kad programavimo mokymui(si) skirta aplinka, kuri yra integruota į *VMA Moodle* visada veikia tinkamai.

Net 100% mokinių atsakė, kad testai, esantys mokymosi aplinkoje atitinka skirtingus mokinių poreikius.

Anketos pabaigoje apklausos dalyviai galėjo išsakyti savo pasiūlymus, pageidavimus ir pastabas. Keli mokiniai atsakė, kad neturi jokių pasiūlymų, vienas mokinys norėtų, kad sistema leistų saugoti darbus panašiai kaip *Onedrive*.

4.5. Mokymosi aplinkos tobulinimas ir tęstinumas

Programavimo mokymui(si) skirtos aplinkos, kuri integruota į VMA *Moodle* tyrimo rezultatai parodė, į ką reikėtų atkreipti dėmesį, ką galima būtų tobulinti.

Kad aplinka būtų funkcionalesnė, reikėtų sudaryti galimybę saugoti atliktas užduotis. Reikėtų pateikti daugiau teorinės medžiagos.

Mokymo aplinkoje galima būtų pateikti nuorodas į teorinę medžiagą, kur mokiniai ras papildomos medžiagos.

Tobulinant mokymo aplinką, reikėtų didinti temų ir užduočių skaičių. Pateikti mokiniams kuo įvairesnių užduočių.

Dėl palankių mokinių atsiliepimų tiek mokymo aplinkai, tiek ir programavimo mokymui(si) skirtai aplinkai, ji bus naudojama ir tobulinama toliau. Planuojama sukurti kursus ir jaunesnėms klasėms. Mokymosi aplinka bus pristatyta Elektrėnų savivaldybės informatikos mokytojams.

4.6. Skyriaus išvados

1. Vievio gimnazijoje buvo įdiegta, išbandyta ir naudojama mokymo(si) procese programavimo mokymui(si) skirta aplinka, kuri yra integruota į VMA *Moodle*.
2. Sukurtos užduotys, kurios atitinka skirtingų mokymosi pasiekimų lygius, publikuojamos Vievio gimnazijos VMA *Moodle* sukurtame III gimnazinės klasės kurse.
3. Į VMA *Moodle* įdiegta programavimo mokymui(si) skirta aplinka, kuri pritaikyta skirtingų mokymosi pasiekimų lygių mokiniams.
4. Atliktas programavimo mokymui(si) aplinkos vertinimas bei, pasinaudojant jo rezultatais, parengtas programavimo mokymui(si) aplinkos tobulinimo planas ir numatytas jos tęstinumas.

Išvados

1. Literatūros analizė parodė, kad programavimas yra svarbus ir naudingas besimokantiejiems. Šiuo metu mokiniams mokantis informatiką pagal atnaujintas programas skiriamas ypatingai didelis dėmesys programavimui, todėl būtina pasirinkti tinkamą programavimo mokymui(si) skirtą aplinką. Programavimas bendro lavinimo mokyklose padeda mokiniams pasiruošti ateičiai, pasiekti aukštesnįjį programavimo lygį. Taip pat programavimas mokiniams stiprinti pasitikėjimą savimi, lavina problemų sprendimo įgūdžius, atkaklumą ir organizuotumą. Programavimui mokyti geriausia naudoti tas aplinkas, kurios skatina ieškoti problemų sprendimų, įtraukia besimokantįjį į mokymosi procesą bei suteikia greitą grįžtamąjį ryšį, leidžia eksperimentuoti ir per pažinimą susieti abstrakčius programavimo objektus su realiame gyvenime esančiais objektais. Svarbu mokiniams suteikti ne tik techninius įgūdžius, tačiau skatinti jų kritinį mąstymą, problemų sprendimo gebėjimus. Programavimo mokymas turi būti orientuotas į visapusišką mokinio tobulėjimą.
2. Atlikto kiekybinio tyrimo rezultatai parodė, kad tiek mokytojai, tiek mokiniai norėtų savo naudojamoje virtualioje mokymo aplinkoje turėti programavimo mokymui(si) skirtą aplinką, kuri būtų pritaikyta skirtingų mokymosi pasiekimų lygių mokiniams. Tikimasi, kad įdiegta programavimo mokymui(si) skirta aplinka, kuri bus pritaikyta skirtingų mokymosi pasiekimų lygių mokiniams, didins mokinių motyvaciją mokytis, ugdyt programavimo įgūdžius, skatins savarankiškumą.
3. Remiantis tyrimo rezultatais, taip pat išanalizavus VMA funkcines ir nefunkcines savybes, palyginus virtualias mokymo aplinkas, VMA Moodle sukurtas kursas III klasių mokiniams, kuriame įrankio *Schellter* pagalba, naudojant SCORM objektus yra įdiegta programavimo mokymui(si) skirta aplinka, kurioje pateiktos užduotys pagal mokinių pasiekimų lygius ir kuri pritaikyta skirtingų mokymosi pasiekimų lygių mokiniams.
4. Vievio gimnazijoje įdiegta, išbandyta ir naudojama programavimo mokymui(si) skirta aplinka. Šioje aplinkoje mokiniai dirba individualiai pagal savo gebėjimų lygį. Taip pat savo žinias gali patikrinti atlikdami testus. Programavimo mokymui(si) skirta aplinka turi dirbtinio intelekto asistentą, kuris mokiniams patinka ir esant galimybei juo būtinai pasinaudoja.
5. Įvertinant sukurtą ir naudojamą programavimo mokymui(si) skirtą aplinką, mokiniams buvo pateiktas klausimynas. Išskirta kaip reikėtų tobulinti sukurtą aplinką: pateikti daugiau teorinės medžiagos, daugiau užduočių ir testų. Sudaryti galimybę saugoti mokinių atliktas užduotis debesyse.

Literatūros sąrašas

1. Dagienė V. ir Urbonienė J (2010). *Programavimo mokymasis: lyginamoji kalbos ir aplinkos analizė. Information & Media* [žiūrėta 2022-11-20]. Prieiga per internetą: <https://www.ceeol.com/search/article-detail?id=92620>
2. Fessakis, G ir kt. (2013). *Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. Computers and Education* [žiūrėta 2022-11-20]. Prieiga per internetą: <https://www.sciencedirect.com/science/article/abs/pii/S0360131512002813>
3. Louka, K. (2022). *Programming environments for the development of CT in preschool education: A systematic literature review. Adv Mobile Learn Educ Res, 3(1), 525-540* [žiūrėta 2022-11-25]. Prieiga per internetą: <https://www.syncsci.com/journal/AMLER/article/view/AMLER.2023.01.001>
4. Krėpšta E. (2018). *GitHub yra priemonė gauti pirmą programuotojo darbą.* [žiūrėta 2022-12-01]. Prieiga per internetą: <https://kodinu.lt/github-yra-priemone-gauti-pirma-programuotojo-darba/>
5. ERUMIT, K. A. ir BENZER, A. (2020). *İhsan; ŞAHİN, Güven. A framework for studying programming teaching in secondary education. Croatian Journal of Education: Hrvatski časopis za odgoj i obrazovanje, 2020, 22.2: 563-595.* [žiūrėta 2022-12-01]. Prieiga per internetą: <https://doi.org/10.15516/cje.v22i2.3560>
6. Sklirou ir Theofani S. (2017). *Programming in secondary education: Applications, new trends and challenges. In: 2017 IEEE Global Engineering Education Conference (EDUCON). IEEE, 2017. p. 580-585.* [žiūrėta 2022-12-05]. Prieiga per internetą: <https://ieeexplore.ieee.org/abstract/document/7942904>
7. Kelleher C. ir Pausch R. (2005). *Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. ACM computing surveys (CSUR), 2005, 37.2: 83-137.* [žiūrėta 2022-12-10]. Prieiga per internetą: <https://www.syncsci.com/journal/AMLER/article/view/AMLER.2023.01.001>
8. Wright T. ir Cockburn A. (2003). *A language and task-based taxonomy of programming environments. In: IEEE Symposium on Human Centric Computing Languages and Environments, 2003. Proceedings. 2003. IEEE, 2003. p. 192-194.*MCGILL. [žiūrėta 2022-12-20]. Prieiga per internetą: <https://ieeexplore.ieee.org/abstract/document/1260227>
9. Monica M. ir Decker A.(2020). *Construction of a taxonomy for tools, languages, and environments across computing education. In: Proceedings of the 2020 ACM Conference on International Computing Education Research. 2020. p. 124-135.* [žiūrėta 2023-01-05]. Prieiga per internetą: <https://dl.acm.org/doi/abs/10.1145/3372782.3406258>
10. Šiaulys T. (2020). *Engagement taxonomy for introductory programming tools: failing to tackle the problems of comprehension. In: Informatics in Schools. Engaging Learners in Computational Thinking: 13th International Conference, ISSEP 2020, Tallinn, Estonia, November 16–18, 2020, Proceedings 13. Springer International Publishing, 2020.*

- p. 94-106. [žiūrėta 2023-03-25]. Prieiga per internetą: https://link.springer.com/chapter/10.1007/978-3-030-63212-0_8
11. Saito ir Daisuke, et al. (2017). *Program learning for beginners: Survey and taxonomy of programming learning tools*. In: *2017 IEEE 9th International Conference on Engineering Education (ICEED)*. IEEE, 2017. p. 137-142. [žiūrėta 2023-03-25]. Prieiga per internetą: <https://ieeexplore.ieee.org/abstract/document/8251181>
 12. Lindberg R., Renny S., Laine, Teemu H. Ir Haaranen L. (2019). *Gamifying programming education in K-12: A review of programming curricula in seven countries and programming games*. *British Journal of Educational Technology*, 2019, 50.4: 1979-1995. [žiūrėta 2023-06-11]. Prieiga pe internetą: <https://berajournals.onlinelibrary.wiley.com/doi/abs/10.1111/bjet.12685>
 13. Rolandsson L. ir Skogh I. B. (2014). *Programming in school: Look back to move forward*. *ACM Transactions on Computing Education (TOCE)*, 2014, 14.2: 1-25. [žiūrėta 2023-06-10]. Prieiga per internetą: <https://dl.acm.org/doi/abs/10.1145/2602487>
 14. Sun D. (2023). *Transitioning From Introductory to Professional Programming in Secondary Education: Comparing Learners' Computational Thinking Skills, Behaviors, and Attitudes*. *Journal of Educational Computing Research*, 2023, 07356331231204653. [žiūrėta 2023-06-10]. Prieiga per internetą: <https://journals.sagepub.com/doi/abs/10.1177/07356331231204653>
 15. Jumankuziev U. (2023). *USING NEW APPROACHES TO TEACHING PROGRAMMING LANGUAGES IN SECONDARY SCHOOLS*. *Galaxy International Interdisciplinary Research Journal*, 2023, 11.5: 280-283. [žiūrėta 2023-06-12]. Prieiga per internetą: <https://www.giirj.com/index.php/giirj/article/view/5298>
 16. Hsu, T. C., Chang, S. C. ir Hung, Y. T. (2018). *How to learn and how to teach computational thinking: Suggestions based on a review of the literature*. *Computers & Education*, 126, 296-310. [žiūrėta 2023-06-15]. Prieiga per internetą: <https://www.sciencedirect.com/science/article/abs/pii/S0360131518301799>
 17. *Atnaujinta bendroji programa* [žiūrėta 2023-05-04]. Prieiga per internetą: . <https://www.emokykla.lt/bendrosios-programos/pradinis-ugdymas/3?st=1>
 18. *Vidurinio ugdymo bendrosios programas: informacinės technologijos* (2011) [žiūrėta 2022-04-24]. Prieiga per internetą: https://duomenys.ugdome.lt/saugykla/bp/2016/vidurinis/IT_7_priedas.pdf.
 19. *Informacinių technologijų brandos egzamino programa* (2011) [žiūrėta 2022-04-24]. Prieiga per internetą: <https://www.nsa.smm.lt/wp-content/uploads/2022/01/Informaciniu-technologiju-VBE-programa-2021-10-29-V-1983.pdf>.
 20. *Pagrindinio ugdymo bendrosios programas: informacinės technologijos* (2008) [žiūrėta 2023-01-02]. Prieiga per internetą: https://duomenys.ugdome.lt/saugykla/bp/2016/pagrindinis/8_Informacines_technologijos.pdf.

21. Burbaitė, R. (2014). *Išplėstiniai generatyviniai mokymosi objektai informatikos mokymuisi: koncepcija, modeliai ir realizacija*. Kaunas: Technologija [žiūrėta 2023-05-23]. Prieiga per internetą:
22. Balbieris, G. (2005). *Virtualioji mokymosi aplinka mokyklai*. [žiūrėta 2023-06-14]. Prieiga per internetą: <https://www.mtp.smm.lt/aktualijos.htm>
23. Winslow, L. E. (1996). *Programming pedagogy—a psychological overview* [žiūrėta 2022-12-01]. Prieiga per internetą: <https://dl.acm.org/doi/pdf/10.1145/234867.234872>.
24. Lodi M. ir Martini S. (2021). *Computational thinking, between Papert and Wing*. *Science & education*, 2021, 30.4: 883-908. [žiūrėta 2023-06-10]. Prieiga per internetą: <https://link.springer.com/article/10.1007/s11191-021-00202-5>
25. <https://github.com/> [žiūrėta 2024-05-02]
26. <https://www.geeksforgeeks.org/> [žiūrėta 2023-05-02]
27. <https://ideone.com/> [žiūrėta 2023-05-02]
28. <https://replit.com/cloud-development-environment> [žiūrėta 2023-05-03]
29. <https://www.jdoodle.com/> [žiūrėta 2023-05-03]

Priedai

1 priedas. Tyrimas „Programavimo aplinkos kūrimas, skirtingų poreikių mokiniams“

Gerbiamas respondente,

Esu Kauno technologijos universiteto Nuotolinio mokymosi Informatikos studijų programos magistrantė. Atlieku tyrimą, kurio tikslas - išsiaiškinti kokias aplinkas naudojate, mokant moksleivius programuoti. Anketa skirta informacinių technologijų mokytojams.

Nuoširdžiai atsakydami į anketos klausimus, Jūs suteiksite reikiamą informaciją ir padėsite kurti man tokią programavimui skirtą aplinką, kuri bus pritaikyta skirtingų poreikių mokiniams. Ši apklausa yra anoniminė, bus naudojami tik statistiškai apibendrinti anketos duomenys. Dėkoju už atsakymus .

1. Kokiose klasėse dėstote programavimo pamokas?*

 - 5 - 6 kl.
 - 7 - 8 kl.
 - 9 - 10 kl.
 - 11 - 12 kl.

2. Kokiose klasėse dėstote programavimo pamokas?*

 - 5 - 6 kl.
 - 7 - 8 kl.
 - 9 - 10 kl.
 - 11 - 12 kl.

3. Ar naudojate programavimo mokymui skirtas aplinkas?*

 - Taip
 - Ne

4. Jeigu atsakėte "taip", įvardinkite kokias?
5. Ar jus tenkina programavimo mokymui skirtų aplinkų pasiūla? Argumentuokite atsakymą.*
6. Ar jūsų manymu reikalinga būtų programavimo mokymui skirta aplinka, skirtingų poreikių mokiniams?*

 - Taip
 - Ne

7. Pagrįskite savo atsakymą.

8. Ar turite savo mokykloje sukurta programavimo mokymui skirta aplinka?
- Taip
 - Ne
 - Jei taip, tai kokia.
9. Ar naudojate programavimo mokymui skirtas aplinkas?*
- Taip
 - Ne
10. Jeigu atsakete "taip", ivardinkite kokias?
11. Ar jus tenkina programavimo mokymui skirtu aplinku pasiula? Argumentuokite atsakyma.*
12. Ar jusu manymu reikalinga butu programavimo mokymui skirta aplinka, skirtingu poreikiu mokiniam?*
- Taip
 - Ne
13. Pagriskite savo atsakyma.
14. Ar turite savo mokykloje sukurta programavimo mokymui skirta aplinka?
- Taip
 - Ne
15. Jei taip, tai kokia.

2 priedas. Tyrimas „Programavimo aplinkų poreikis, kuris reikalingas skirtingų poreikių mokiniams“.

Gerbiamas respondente,

Esu Kauno technologijos universiteto Nuotolinio mokymosi Informatikos studijų programos magistrantė. Atlieku tyrimą, kurio tikslas - išsiaiškinti kokias aplinkas naudojate, mokant moksleivius programuoti. Anketa skirta informacinių technologijų mokytojams.

Nuoširdžiai atsakydami į anketos klausimus, Jūs suteiksite reikiamą informaciją ir padėsite kurti man tokią programavimui skirtą aplinką, kuri bus pritaikyta skirtingų poreikių mokiniams. Ši apklausa yra anoniminė, bus naudojami tik statistiškai apibendrinti anketos duomenys. Dėkoju už atsakymus .

1. Jūsų lytis:

- Mot.
- Vyr.

2. Kurioje klasėje mokotės?

- II
- III
- IV

3. Nuo kelintos klasės pradėjote mokytis programavimo?

4. Kodėl nusprendėte pasirinkti programavimo pamokas?

- Noriu ateityje pasirinkti programuotojo specialybę
- Mokytojo iniciatyva
- Sekasi tikslieji mokslai
- Kita

5. Kokią programavimo kalbą naudojote pamokų metu?

- C++
- Phytton

6. Ar užtenka programavimo mokymuisi skirtos vienos savaitinės valandos?

- Taip
- Ne

7. Ar užtenka įgytų žinių pamokų metu?

- Taip, gauname daug informacijos

- Ne, daug mokausi savarankiškai
 - Pamokų metu nesuprantu, todėl lankau papildomas pamokas
8. Ar naudojate programavimo mokymuisi papildoma medžiaga?
- Vaizdo įrašais
 - Pamokose gauta informacija
 - Padeda draugai
 - Konsultuoja mokytojas
9. Kokia būtų jūsų didesnė motyvacija mokantis programuoti?
- Programų kodai su pavyzdžiais
 - Namų darbų skyrimas
 - Užduočių diferencijavimas
 - Atsižvelgti į skirtingų mokinių poreikius
10. Ar užtenka mokantis programavimo skirtų aplinkų?
- Taip
 - Mokykloje nenaudojame jokių aplinkų
 - Pats susiradau ir naudoji mokymosi aplinką
11. Ar jūsų manymu reikalingos mokymosi aplinkos, mokantis programavimo?
- Taip, būtų naudinga pačiam mokiniui
 - Ne, tai neturi reikšmės
 - Neturiu nuomonės
12. Kokias žinote programavimo mokymui skirtas aplinkas?
13. Ar žinote, kokia jūsų mokykloje įdiegta nuotolinio mokymo aplinka? Įvardinkite.
14. Ar jūsų manymu būtų naudinga savo mokykloje, mokantis programuoti turėti programavimo mokymuisi skirtą aplinką, kuri būtų pritaikyta skirtingų poreikių mokiniams?
- Taip
 - Ne
 - Neturiu nuomonės
15. Pagrįskite savo atsakymą jeigu atsakėte "taip".

3 priedas. „Programavimo mokymui(si) skirtos aplinkos naudojimas“

Esu Kauno technologijos universiteto Nuotolinio mokymosi informacinės technologijos studijų programos magistrantė. Atlieku tyrimą, kurio tikslas – įvertinti integruotą programavimo mokymosi aplinką, mokymosi aplinkos privalumus ir trūkumus, jos naudingumą bei tobulinimo galimybes. Maloniai prašyčiau atsakyti į žemiau pateiktus klausimus. Anketa yra anoniminė. Gauti tyrimo rezultatai bus naudojami moksliniams tikslams, rengiant baigiamąjį projektą. Nuoširdžiai dėkoju už sutikimą atsakyti į visus anketos klausimus bei už sugaištą laiką. Sėkmės!

1. Jūsų lytis:
 - Vyr.
 - Mot.
2. Kiek laiko naudojate VMA Moodle?
 - 1 metus
 - 2 metus
 - Pusė metų
3. Ar kyla problemų naudojantis VMA Moodle?
 - Nepatinka Moodle mokymosi aplinka
 - Nekyla, viskas aišku ir konkrečiu
 - Kita:
4. Kiek laiko naudojate programavimo aplinką integruotą į VMA Moodle?
 - 1 metus
 - 2 metus
 - Pusė metų
5. Ar manote, kad ši programavimo aplinka, integruota į VMA Moodle, atitinka jūsų poreikius?
 - Visiškai atitinka
 - Neatitinka
 - Atitinka iš dalies
6. Kokių funkcijų šioje programavimo aplinkoje trūksta, kad ji būtų patogesnė ir funkcionalesnė?
7. Jūsų atsakymas.....
8. Ar programavimo mokymosi aplinka integruota į VMA Moodle suteikia motyvacijos ir paskatinimo siekti tikslo?

- Tikrai suteikia
 - Nei suteikia, nei nesuteikia
 - Nesuteikia
9. Ar pateikiamas pakankamas užduočių skaičius?
- Pateikiama per daug užduočių
 - Pateikiama per mažai užduočių
 - Užduočių skaičius atitinka pamokos trukmę
10. Kaip vertinate užduoties struktūrą, kuri pateikta VMA Moodle?
- Užduotys per sunkios
 - Užduotys per lengvos
 - Užduotys suformuluotos aiškiai ir konkrečiai
 - Užduotyse trūksta aiškumo
11. Ar VMA Moodle reiktų pateikti daugiau teorinės medžiagos?
- Man įdomios tik praktinės užduotys
 - Teorinės medžiagos pateikta užtektinai
 - Galėtų būti daugiau
12. Pagal pasiekimų lygius kokio sudėtingumo užduotis renkatės?
- Slenkstinį pasiekimų lygį
 - Patenkinamą pasiekimų lygį
 - Pagrindinį
 - Aukštesnįjį pasiekimų lygį
13. Ar esate patenkintas užduotimis, kurios pritaikytos skirtingų poreikių mokiniams (pagal pasiekimų lygius)?
- Patenkintas
 - Patenkintas iš dalies
 - Nei patenkintas nei nepatenkintas
 - Kita:

14. Ar visada tinkamai veikia programavimo aplinka integruota į VMA Moodle?

- Veikia visada
- Neveikia
- Kartais stringa

15. Ar VMA Moodle esantys testai atitinka skirtingus mokinių poreikius?

- Taip
- Ne
- Neturiu atsakymo

16. Ar turite pasiūlymų, kaip galėtų būti geriau pritaikytos programavimo užduotys pagal mokinių pasiekimų lygius?

Jūsų atsakymas.....

17. Jūsų pastabos, pasiūlymai apie VMA moodle integruotą programavimo aplinką.*

18. Jūsų atsakymas.....

4 priedas. Programavimo mokymosi aplinkos, pritaikytos skirtingų mokymosi lygių pasiekimų mokiniams, diegimo aktas Vievio gimnazijoje



ELEKTRĖNŲ SAV. VIEVIO GIMNAZIJA

Biudžetinė įstaiga. Šviesos g. 4A, LT-21375 Vievis, Elektrėnų sav.
Tel.: +370 528 26 211, +370 528 26 884, +370 528 26 652, +370 528 26 433. El. p. vieviogimnazija@gmail.com
Duomenys kaupiami ir saugomi Juridinių asmenų registre, kodas 190669038.

Kauno technologijos universitetui

**PAŽYMA APIE VYRESNIOSIOS INFORMATIKOS MOKYTOJOS
VITOS STRASEVIČIENĖS SUKURTĄ KURSĄ III G. KLASIŲ MOKINIAMS IR
ĮDIEGTĄ PROGRAMAVIMO APLINKĄ, PRITAIKYTĄ SKIRTINGŲ MOKYMOŠI
PASIEKIMŲ LYGIŲ MOKINIAMS**

2024-05-16 Nr. 1S-73

Pažymime, kad Elektrėnų sav. Vievio gimnazijos naudojamoje VMA Moodle aplinkoje išbandyta ir naudojama programavimo mokymosi aplinka, skirta skirtingų mokymosi pasiekimų lygių III gimnazijos klasės mokiniams, kurią įdiegė vyresnioji informatikos mokytoja, KTU informatikos magistrantė Vita Strasevičienė. Aplinkoje pateiktos užduotys, testai pritaikyti skirtingų mokymosi pasiekimų lygių mokiniams.

Direktorius



Gintaras Dobilaitis