

7-8 klasė

Kompiuterių raida (Neringa Kušlienė)

Pasiekimų sritis	B. Algoritmai ir programavimas
Klasė	7-8 (2 pamokos)
Tema	Kompiuterių raida
Integruojami dalykai, pasiekimai	Istorija, anglų k., matematika, fizika, technologijos, lietuvių k.
Kompetencijos	Pažinimo – (mokėjimas mokytis) žinių gilinimas apie kompiuterių atsiradimo istoriją. Skaitmeninė – ugdyti gebėjimą dirbti su nauju įrankiu kuriant interaktyvią knygą. Komunikavimo – pagalba draugams, atliktų darbų pristatymai, pasidalinimas idėjomis, diskusijos. Kūrybiškumo - idėjų generavimas kuriant plakatą/ interaktyvią knygą.
Tikslas	Susipažinti su kompiuterių raida ir sukurti interaktyvią knygą.
Uždaviniai	<ol style="list-style-type: none"> 1. Naudodamiesi rekomenduotais informacijos šaltiniais išrinksite reikiamą informaciją. 2. Klasėje sukursite interaktyvią knygą apie kompiuterių raidą (atskleisti esmę). Galimos alternatyvos: <ol style="list-style-type: none"> a. surengti protmūšį; b. surengti diskusiją; c. surengti viktoriną; d. sukurti kryžiažodį; e. sukurti mokomąjį plakatą, lankstinuką ir pan. f. pabėgimo kambarys. 3. Apibendrinti tai, ką sužinojote.
Planuojamas rezultatas	<ol style="list-style-type: none"> 1. Gebėsite surasti patikimą informaciją apie kompiuterių raidą. 2. Sukursite interaktyvią knygą apie kompiuterių raidą. 3. Pagilinsite darbo pristatymo įgūdžius.
Specifinės priemonės / programinė įranga	Multimedia projektorius, išmanioji lenta, Canva programa plakatui kurti. Viktorinai Mentimeter, Quizizz, Socrative. Pabėgimo kambariui Genially. WordWorld (jei yra licencija), Learning Apps – kryžiažodžiui.
Mokymosi metodai	Apversta klasė (namuose skaito pateiktą, surastą medžiagą) Diskusija, praktinis darbas kuriant kryžiažodį, plakatą, keitimasis vaidmenimis, darbas grupėmis, interaktyvios knygos kūrimas.
Mokinių atlikto darbo vertinimas ir įsivertinimas	Slenkstinis pasiekimų lygis – su pagalba sukuria struktūrizuotą konspektą. Patenkinamas pasiekimų lygis – sukuria struktūrizuotą konspektą savarankiškai. Pagrindinis pasiekimų lygis – paaiškina atliktą darbą (pasakodami išplečia). Aukštesnysis lygis – analizuoja ir vertina bendraklasių atliktus darbus, suteikia pagalbą.
Žinios prieš	Geba surasti patikimą informaciją įvairiuose šaltiniuose, geba dirbti bendrojo naudojimo programine įranga.
Galimybės taikyti spec.	Mokytojas scenarijų pritaiko priklausomai nuo vaiko specialiųjų poreikių.

poreikių mokiniams	
Patarimai kolegoms, kurie naudos parengtą medžiagą	<ol style="list-style-type: none"> 1. Klasei, kurioje bus vedama pamoka, pasiūlyti išsirinkti norimą veiklą klasėje balsuojant. 2. Peržiūrėti mokinių surinktą medžiagą ir įvertinti jos tinkamumą. 3. Išsiaiškinti dėl interaktyvios knygos kūrimo įrankio, kokie mokinių gebėjimai dirbti su pasirinktu įrankiu (numatyti ar interaktyvios knygos kūrimas nenustums į šoną kompiuterių raidos istorinių žinių).

1 ETAPAS 🕒

Mokiniams siūloma medžiaga darbui namuose (gali ir iš kitų šaltinių):

https://www.livescience.com/20718-computer-history.html?fbclid=IwAR3DkGJqXgmREUSFWA-GhJ8d7oa-ZGaQKF6jdEO_kZtzuAYZz0IP0IneMeM

https://www.youtube.com/watch?v=6owLE0_GW8U

<https://www.youtube.com/watch?v=gjVX47dLIN8&t=101s>

https://www.youtube.com/watch?v=tw_bsmVYIyM

<https://www.youtube.com/watch?v=M4d3FXu9-3l&t=26s>

<https://www.youtube.com/watch?v=MGL047HMxSk>

<https://www.youtube.com/watch?v=ckXcMJlI1ZE>

Jeigu planuojate klasėje kurti interaktyvią knygą, rekomenduojama mokiniams nusifotografuoti savo dabartinį kompiuterį, planšetę (jei nori), kurią įdės į savo kuriamą knygą.

2 ETAPAS 🕒 (10 min.) Darbas klasėje.

Mokiniai pristato paruoštą medžiagą. Darbo sėkmės ir nesėkmės pavaizduojamos voratinklyje. Aptarimas.

3 ETAPAS 🕒 (25 – 30 min.)

Interaktyviosios knygos kūrimas „Mano kompiuterio istorija“. Panaudojama mokinių rinkta medžiaga.

1. Su mokiniais kalbama apie elektroninę knygą, jos privalumus. Demonstruojamas elektroninės knygos pavyzdys (<https://www.storyjumper.com/book/search>)

2. Pristatomas elektroninės knygos kūrimo įrankis *storyjumper*, demonstruojama, kaip maketuojama knyga, kaip kuriami puslapiai, įkeliamas fonas, veikėjai, teksto laukas, garsas:

Rodoma pažingsniui kaip kuriama elektroninė knyga. Spustelėję *+Create Book* pasirenkame baltą knygos kūrimo foną. Pradedame kurti knygos viršelį.

Knygos puslapiai (išsklotinė) – ekrano apačioje. Mėlynas puslapis – knygos viršelis. Antrasis puslapis – priešlapis. Toliau puslapiai sunumeruoti paeiliui, rodykle ar slankmačiu galima nukeliauti iki paskutinio, jį trinti arba pridėti papildomus puslapius, pats paskutinis puslapis – taip pat viršelis.

Viršelio kūrimas. Galima keisti jo spalvą (šalia – pasirinkimo galimybės), tekstūrą, užrašyti autoriaus vardą

ir pavardę, pavadinimą, įterpti bet kokį norimą tekstą ar paveikslėlį iš Meniu juostos (kairėje).

Meniu juosta (kairiajame kampe) **ir turinio kūrimas:** 5 pagrindiniai mygtukai su ikonėlėmis ir rodyklėmis, jas išskleidus galime pasirinkti reikiamą teksto lauką (*text*), foną (*scenes*), veikėją (*props*), savo nuotrauką (*photos*) ir garsą (iš pateikto garsų albumo arba savo).

Violetine spalva paryškinamas langas, kuris koreguojamas ir kuriamas, tuo pačiu matome ir puslapių išklotinėje kurį knygos puslapį kuriame/koreguojame.

Baigus kurti knygą, spaudžiame viršutiniame dešiniajame kampe *Save&Exit*.

Mokytoja prisijungusi prie savo paskyros gali matyti kiekvieno mokinio darbą, jį taisyti, komentuoti, o mokinys bet kada gali knygą koreguoti, skaityti komentarus, bendrinti knygos nuorodą.

3. Mokiniam nusiunčiama prisijungimo prie svetainės nuoroda (ją mokytojas gauna sukūręs klasę), pasakomas klasės prisijungimo slaptažodis, kuriuos mokytoja sugeneravo prieš pamoką
4. Mokiniai spaudžia mygtuką *+Create book* ir bando kurti savo pirmąją knygą – įkelti tekstą, foną, veikėjus.
5. Sukurti elektroninę knygą taip, kad kiekvienas puslapis būtų iliustruotas savo darbais.
6. Galima užduoti sunkinti parenkant puslapių įgarsinimą.

Rekomendacijos mokytojui: Mokytojui reikia prisiregistruoti svetainėje <https://www.storyjumper.com/> ir sukurti savo klasę (*+add class*), sukūrus klasės slaptažodį, sukurti mokinių prisijungimo vardus (rekomenduojama nenurodant asmens duomenų). Mokiniai nesudėtingai jungsis įvesdami savo prisijungimo vardą ir klasės slaptažodį (vėliau jį galės keisti) arba jungdamiesi per sistemos automatiškai sugeneruotą nuorodą. Mokytoja gali stebėti kiekvieno mokinio darbą, komentuoti ir koreguoti.

Šaltiniai, nuorodos: <https://www.storyjumper.com/>

Galimas ir kitas interaktyvios knygos įrankis: „BookCreator“. Supažindina arba prisimena šio įrankio naudojimosi instrukciją. Mokytoja atsiunčia pakvietimą bendros knygos kūrimui. Mokiniai atsirenka jiems tinkamus šaltinius, pasirenka medžiagą, kurią dės į savo knygos puslapį.

3 ETAPAS 🕒 (10 min.)

Interaktyvių knygų pristatymas, aptarimas. Refleksija „Mentimeter“ įrankiu. Klausimus pateikti iš Kompiuterių raidos.

Algoritmai ir programos (Neringa Kušlienė)

Pasiekimų sritis	B. Algoritmai ir programavimas
Klasė	7-8 (2 pamokos)
Tema	Algoritmai ir programos
Integruojami dalykai, pasiekimai	Matematika, technologijos, lietuvių k.
Kompetencijos	Pažinimo – mokosi įvertinti kuo skiriasi algoritmas ir programa, mokosi algoritmą transformuoti į programą. Skaitmeninė – tobulina darbo vaizdinėje programavimo aplinkoje „Scratch“ įgūdžius. Kūrybiškumo – žaidybinio siužeto algoritmo ir programos kūrimas parenkant įvairias raiškos priemones. Komunikavimo – pagalba draugams, atliktų darbų pristatymai, pasidalinimas idėjomis, diskusijos.
Tikslas	Gebėti parengti uždavinio algoritmą ir jį atvaizduoti programa.
Uždaviniai	<ol style="list-style-type: none">1. Išsiaiškinsite pagrindines algoritmo ir programos savybes, panašumus ir skirtumus.2. Sukursite žaidimo algoritmą3. Sukurtą algoritmą įgyvendinsite vaizdinio programavimo aplinkoje Scratch.4. Apibendrinsite ir pristatysite gautus rezultatus.
Planuojamas rezultatas	<ol style="list-style-type: none">4. Nurodysite kuo skiriasi ir kuo panašūs algoritmas ir programa.5. Gebėsite sukurti žaidimo siužetą (algoritmą).6. Sukursite algoritmą įgyvendinančią programą.7. Pagilinsite darbo pristatymo įgūdžius.
Specifinės priemonės / programinė įranga	Multimedia projektorius, išmanioji lenta Scratch programavimo aplinka Padlet virtuali lenta Vaizdinio programavimo aplinka Scratch: https://scratch.mit.edu/ Pažintis su blokeliais https://scratch.mit.edu/projects/278769745/
Mokymosi metodai	Aiškinimas, darbas grupėmis, praktinis tyrimas, diskusija.
Mokinių atlikto darbo vertinimas ir įsivertinimas	Slenkstinis – su mokytojo pagalba sukuria paprasčiausią žaidimo algoritmą ir programą.

	<p>Patenkinamas – savarankiškai sukuria paprasčiausią žaidimo algoritmą ir programą.</p> <p>Pagrindinis – savarankiškai sukuria sudėtingesnį žaidimo algoritmą ir sukuria žaidimą su besikeičiančiomis kaukėmis ir taškų rinkimu.</p> <p>Aukštesnysis – savarankiškai atrenka esminę informaciją, sukuria žaidimą su keletu veikėjų ir taškų rinkimo kelis variantus, prideda komandų, kurie pagyvina žaidimą (pvz, garsus, muziką ir pan.).</p>
Žinios prieš	Moka naudotis Padlet virtualia lenta, žino Scratch aplinkos struktūrą, funkcijas, moka pasirinkti veikėjus, jų kaukes, fonus ir pan.
Galimybės taikyti spec. poreikių mokiniams	Mokytojas scenarijų pritaiko priklausomai nuo vaiko specialiųjų poreikių. Pateikiamas paruoštas kodas, kurį galima pritaikyti pasirinktiems veikėjams.
Patarimai kolegoms, kurie naudos parengtą medžiagą	<ol style="list-style-type: none"> 1. Patys atlikite visas scenarijuje numatytas užduotis. 2. Scenarijų adaptuokite skirtingų gebėjimų mokiniams. 3. Papildykite mokiniams teikiamus šaltinius įvairesniais. 4. Per pamoką mokiniai dirba savarankiškai, mokytojas yra pagalbininkas ir konsultantas.

1 ETAPAS (10- 15 min.). Pamokos temos pristatymas ir sąvokų aiškinimas.

Mokytoja išmanioje lentoje pateikia pamokos tikslą mokiniams – išsiaiškinsite ir suprasite, kas yra **algoritmas ir programa**.

Norint suprasti algoritmo ir programos panašumus ir sirtumus, pirmiausia išsiaiškinkime, kas yra algoritmas.

Algoritmas – tai tam tikra veiksmų seka, kurią reikia atlikti norint pasiekti tam tikrą rezultatą. Kaip mums suprasti, kas yra algoritmas?

Algoritmą gali iliustruoti paprasčiausias kiaušinių išvirimo receptas, kuris galėtų būti toks:

Algoritmo pradžia.

- paimti puodą (sąlyga: kad tilptų vandens apsemti n kiaušiniai);
- į puodą įdėti n sveikų kiaušinių (jeigu netelpa n kiaušinių – paimti didesnę puodą arba sumažinti kiaušinių kiekį n-1);
- pripilti vandens;
- uždėti puodą ant viryklės;
- įjungti viryklę;
- virti (jeigu norime skystai virto kiaušinio – 5 minutes, jei kietai – 15 minučių).
- išjungti viryklę
- nuimti puodą
- išpilti karštą vandenį
- įpilti šaltą vandenį ir palaikyti 1 min.
- išimti kiaušinius

Algoritmo pabaiga.

Panaudojimas

Dažniausiai *algoritmo* sąvoka naudojama informatikoje užrašant kompiuterines programas. Tokiu atveju algoritmų užrašymui naudojami įvairūs susitarimai – programavimo kalbos.

Algoritmas kasdieniniame gyvenime

Gyvenime dažnai susiduriame su *algoritmo* sinonimais: instrukcijomis, nurodymais ir taisyklėmis, kurių nežinodami negalėtume atlikti tam tikrų veiksmų. Tačiau kartais šie aprašymai stokoja tikslumo. Taigi bendrai algoritmą būtų galima apibūdinti kaip **tikslių nurodymų seką tam, kas turės atlikti konkrečią užduotį**. Daugelį kasdieninės veiklos rezultatų pasiekiamo net nesusimąstydami, kad vykdome tam tikrą algoritmą. Jie mums reikalingi: išgyventi (savisaugai), prisitaikyti (adaptacijai), reikiamai vietai pasiekti, prietaisams įjungti, išjungti bei naudoti, pirmajai pagalbai suteikti, maistui pagal receptą gaminti, matematiniams uždaviniams spręsti ir pan. Pagaliau, mūsų visą dieną galima būtų pavadinti algoritmu, nes ji turi savo dienvartę, t. y. veiksmų atlikimo tvarką. Kartais sukeitus algoritmo veiksmus rezultatas nepakinta. Tačiau vykdant kai kuriuos algoritmus veiksmų sukeitimas gali sugriauti visą tolimesnę algoritmo eigą.

Privalomos sąlygos

Algoritmas turi patenkinti šias sąlygas:

1. jis turi atlikti darbą;
2. jis turi būti aiškus ir nedviprasmiškas;
3. jis turi apibrėžti žingsnių seką, reikalingą darbui atlikti, t. y. jis turi nurodyti žingsnių atlikimo tvarką.
 - o Informatikoje dažnai dar reikalaujama, kad algoritmas būtų baigtinis dviem prasmėm:
4. atliekamų žingsnių skaičius turi būti baigtinis, t. y. algoritmas turi tikrai baigti darbą;
5. kiekvienam žingsniui atlikti turi pakakti baigtinio laiko ir baigtinių resursų, t. y. kiekvienas žingsnis turi būti toks, kad jį būtų galima atlikti.

Reikalavimai 4-5 garantuoja, kad algoritmas bus baigtas baigtiniu laiku ir su baigtiniais resursais. Algoritmai, tenkinantys tik sąlygas 1-3, vadinami daliniais algoritmais, o tenkinantys visas penkias sąlygas – pilnais algoritmais.

Algoritmo vykdymas

Parašytas algoritmas yra perduodamas vykdytojui. Vykdytojas gali realizuoti algoritmą, jei yra tam tinkama aplinka. To paties algoritmo efektyvumas (greičio, atminties, patogumo vartotojui ar kitu parametru atžvilgiu) dažniausiai priklauso nuo pasirinktos aplinkos ir sprendimo metodo.

Sudėtingesnių algoritmų sukūrimas, aprašymas bei įdiegimas dažniausiai yra nelengvas darbas, reikalaujantis specialių žinių. Tačiau jų kainą gana greitai atsiperka, jei įdiegti algoritmai vykdomi daug kartų. Vienam vykdytojui algoritmas gali būti aiškus ir nedviprasmiškas, o kitam – visai nesuprantamas. Nuo vykdytojo tiesiogiai priklauso, kokius primityvus galima naudoti, apibrėžiant veiksmus. Jei vykdytojas ne žmogus, algoritmą reikia pateikti specifine, tam vykdytojui priimtina forma, pavyzdžiui, specialia algoritmine kalba. Jei mes turime algoritmą, išreikštą vykdytojo operacijomis, tai jį užrašyti ar perrašyti vienokia ar kitokia kalba nėra sudėtinga.

Informatikoje kaip vykdytojas dažniausia – kompiuteris. Pagrindinės idėjos:

1. kompiuteriai apdoroja duomenis, išreikštus simboliais;
2. jie kontroliuojami instrukcijomis, kurios ir sudaro algoritmą;
3. instrukcijos irgi pateikiamos mašinai kaip simbolių seka.

Taigi viskas, ko reikia algoritmų pateikimui kompiuteriui, tai kalba patogiam instrukcijų užrašymui.

Algoritmo savybės:

- *Diskretumas: algoritmas skaidomas į tiksliai aprašytus vykdymo žingsnius.*

- *Baigtumas*: algoritmas turi turėti pabaigą.
 - *Rezultatyvumas*: algoritmas visada turi pateikti konkretų rezultatą (jei jis egzistuoja) arba paaiškinimą, kodėl jis negautas.
 - *Aiškumas*: algoritmas turi būti pateikiamas taip, kad jį visi vienareikšmiškai suprastų.
 - *Universalumas*: algoritmas turi tikt bet kokiems duomenimis. *Paprastas pavyzdys – elektroninio laiško kūrimas: pradiniai duomenys – reikia turėti adresą A*, reikia turėti laišką L, gali būti ir laiško priedai P. Rezultatas R bus arba išsiųstas, arba neišsiųstas laiškas. O neišsiųsti laiško galime tuo atveju, jei nebus internetinio ryšio.
- Taigi, laiškų siuntimo algoritmą aprašysime kaip vykdomų veiksmų sąrašą:

Darbo pradžia (įjungti kompiuterį)

1. Tikrinti, ar yra ryšys su internetu:
 - jei taip, vykdyti antrą žingsnį;
 - jei ne, vykdyti aštuntą žingsnį.
2. Iškviešti pašto programą.
3. Įvesti A, rašyti laišką (L).
4. Tikrinti, ar reikia siųsti priedus (P):
 - jei taip, vykdyti penktą žingsnį;
 - jei ne, vykdyti šeštą žingsnį.
5. Pridėti prie laiško priedų failus.
6. Išsiųsti laišką.
7. Nustatyti kintamojo R reikšmę „Išsiųsta“. Baigti darbą.
8. Nustatyti kintamojo R reikšmę „Neišsiųsta“.

Darbo pabaiga (baigti darbą su elektroninio pašto programa).

Algoritmas turi patenkinti šias *sąlygas*:

jis turi atlikti darbą; jis turi būti aiškus ir nedviprasmiškas; jis turi apibrėžti žingsnių seką, reikalingą darbui atlikti, t. y. jis turi nurodyti žingsnių atlikimo tvarką. Informatikoje dažnai dar reikalaujama, kad algoritmas būtų baigtinis dviem prasmėm:

atliekamų žingsnių skaičius turi būti baigtinis, t. y. algoritmas turi tikrai baigti darbą; kiekvienam žingsniui atlikti turi pakakti baigtinio laiko ir baigtinių resursų, t. y. kiekvienas žingsnis turi būti toks, kad jį būtų galima atlikti. (Reikalavimai 4-5 garantuoja, kad algoritmas bus baigtas baigtiniu laiku ir su baigtiniais resursais. Algoritmai, tenkinantys tik sąlygas 1-3, vadinami daliniais algoritmais, o tenkinantys visas penkias sąlygas – pilnais algoritmais.)

Kas yra programa? Programa yra konkrečios kompiuterinės sistemos ar platformos vykdoma forma, kuri atlieka konkrečias užduotis. Ji skirta konkrečioms užduotims ar problemoms spręsti, vykdyti. Programa yra kompiuterinės sistemos vykdoma ir yra konkrečios programavimo kalbos realizacija (įgyvendinimas).

Programa gali būti sukurta konkrečiai platformai (pvz., Windows, Android, iOS) ir dažnai priklauso nuo naudojamos programavimo kalbos. Pavyzdžiui, „Microsoft Word“ yra programa, skirta teksto rašymui, redagavimui ir formatavimui, o „Mozilla Firefox“ yra naršyklės programa, naudojama interneto naršymui. Apibendriname. Algoritmas ir programa yra dvi svarbios kompiuterijos sąvokos, tačiau jos turi skirtingus pritaikymo ir prasmės kontekstus.

Panašumai:

- ✓ Algoritmo ir programos sąvokos yra susijusios su kompiuterio naudojimu ir informacijos tvarkymu;
- ✓ tiek algoritmas, tiek programa gali būti naudojami sprendžiant konkrečias problemas ar vykdant užduotis.

Skirtumai:

- ✓ Algoritmas yra abstrakti žingsnių seka, kuri nurodo konkrečios vykdymo aplinkos ar programavimo kalbos. Programa yra konkretus vykdomas kodas, kuris yra susietas su tam tikra kompiuterine sistema.
- ✓ Algoritmas nėra vykdomas kompiuteriu tiesiogiai, o programa yra vykdoma kompiuterio ar kitos vykdomosios aplinkos.
- ✓ Algoritmas yra teorinė konstrukcija, kuri naudojama problemos sprendimo planavimui, o programa yra konkrečios užduoties įgyvendinimo forma.

Suprasti algoritmų svarbą yra svarbu programuotojams. Programos yra įrankiai, kurie leidžia vykdyti sukurtus algoritmus kompiuteriu.

2 ETAPAS (50 min.) Praktinis tyrimas, gali būti darbas grupėmis.

Užduotis: sukurk žaidimo algoritmą, nurodyk žingsnius, ką veikėjas turi padaryti, kokius veiksmus atlikti. Sukūręs žaidimo algoritmą įgyvendink jį Scratch aplinkoje.

Žaidimo pavyzdys, tinkamas slenkstinio ir patenkinamo lygio mokiniams.

Žaidimas: veikėjo ėjimas ir gaudymas objektų (pelyčių) su rodyklemis. Reikia pagauti 10 pelyčių ir laimėsi. Jeigu pagausi 5 kiaušinius - pralaimėsi ir jeigu praleisi 5 pelytes pralaimėsi:

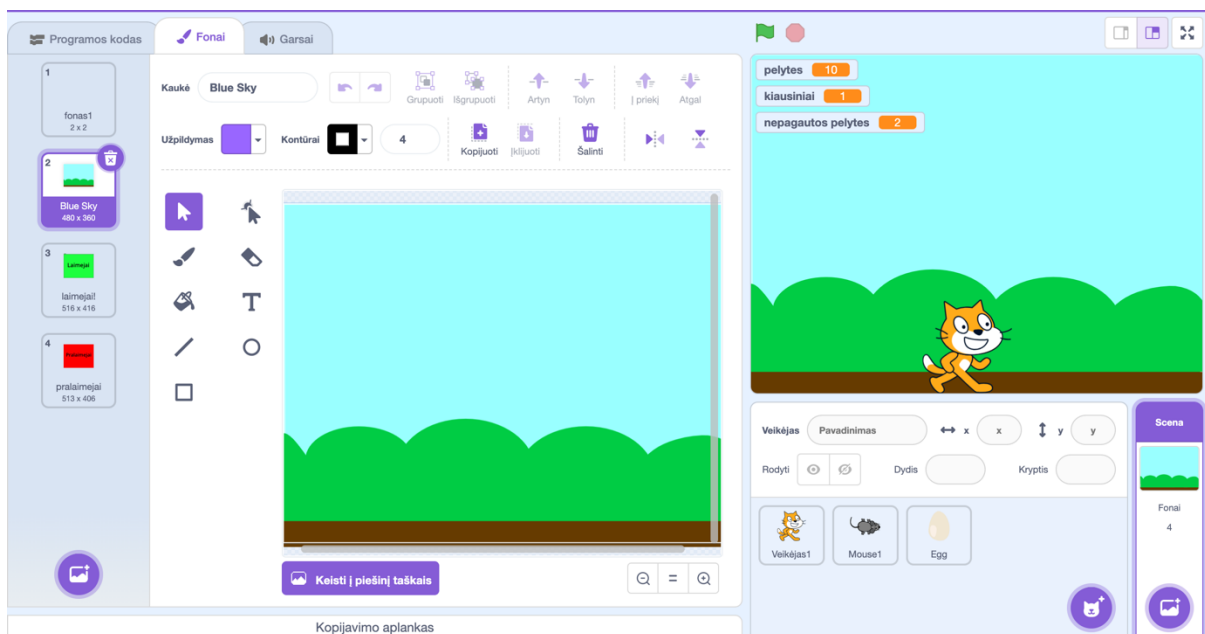
<https://scratch.mit.edu/projects/901784538>

Norint žaidimą laimėti reikia sugauti 10 pelyčių

Pralaimima, jeigu yra sugaunami 5 kiaušiniai arba nepagaunamos 5 pelytės.

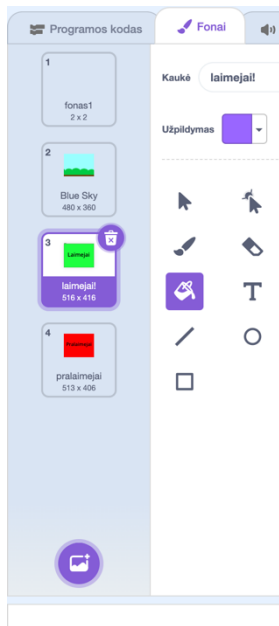
Žaidimo kūrimo etapai:

1. Iš fonų pasirinkti pagrindinį žaidimo foną 1 pav.



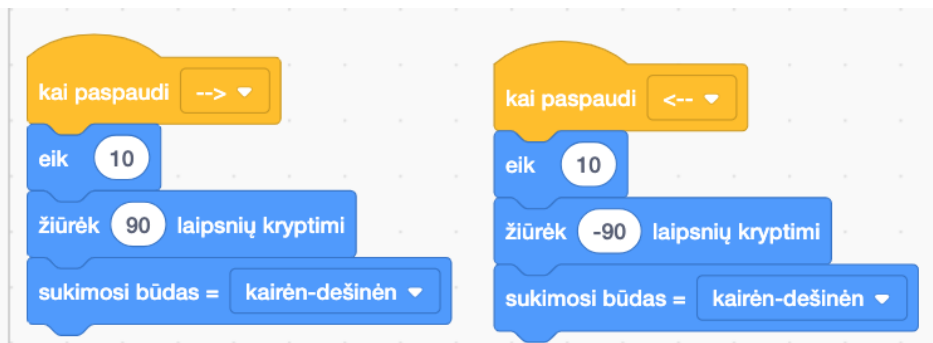
1 pav. Žaidimo fono pasirinkimas.

2. Laimėjimo ir pralaimėjimo fonus galima sukurti, pasirinkus fono spalvą ir nuspalvinius su užpildymo kibirėliu ir užrašant tekstą paspaudus t raidę 2 pav.



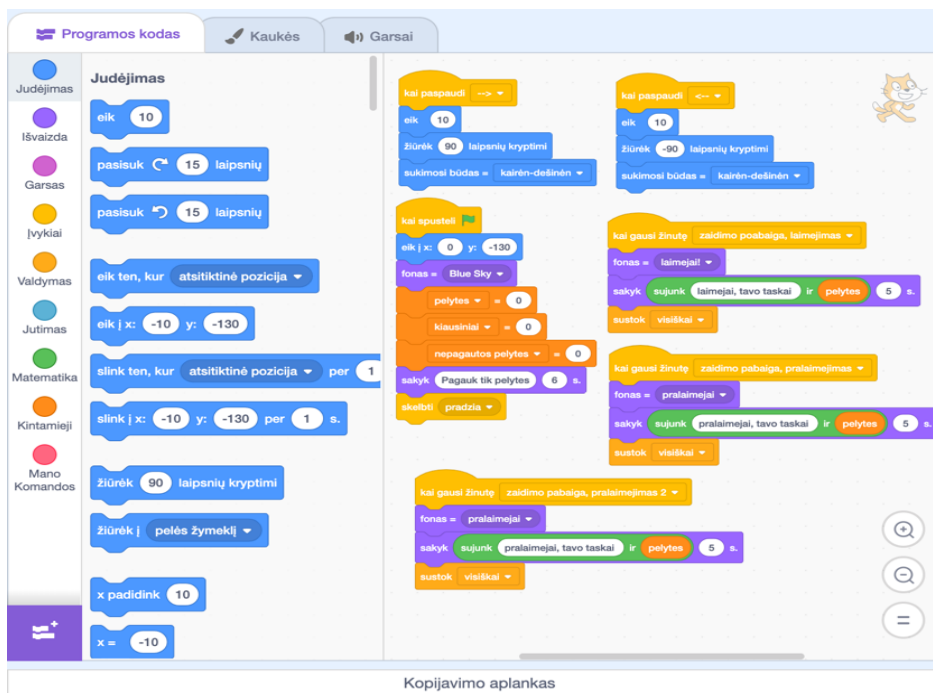
2 pav. Laimėjimo ir pralaimėjimo tekstas

3. Pagrindiniam veikėjui judėti iš „įvykiai“ parinkti „kai paspaudi“ parinkti rodyklę ir iš „judėjimas“ parinkti „eik“ ir parašyti kiek žingsnių, vėliau iš „judėjimas“ parinkti „žiūrėk laipsnių kryptimi“ įrašyti 90 ir -90 ir parinkti sukimosi būdą su blokeliu taip pat iš „judėjimas“ 3 pav.



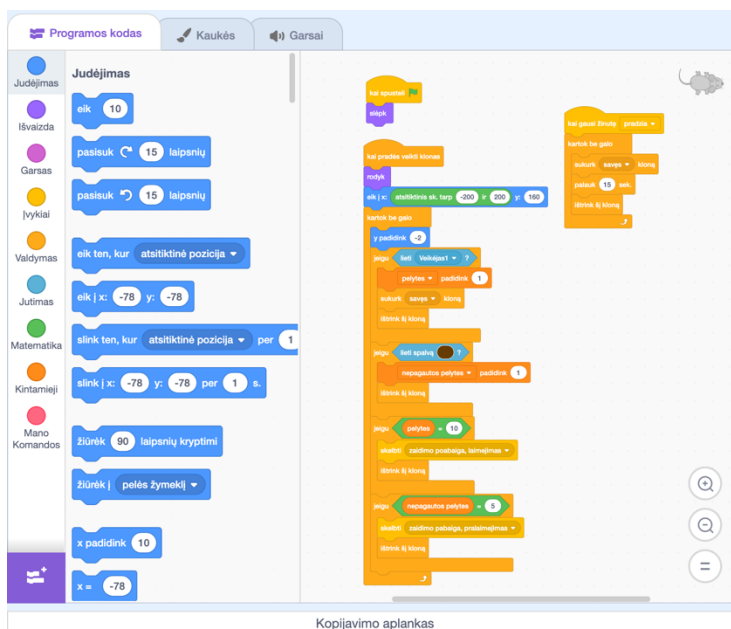
3 pav. Veikėjo judėjimas

4. Vėliau ant pagrindinio veikėjo iš „įvykiai“ parinkti blokelį, kai spusteli žalią vėliavėlę, iš „judėjimas“ parinkti „eik į“, taip nustatyti pradinę pagrindinio veikėjo padėtį. Iš „išvaizda“ paimti fono blokelį ir pasirinkti pagrindinį žaidimo foną. Taip pat iš „kintamieji“ sukurti savo tris kintamuosius „pelytes“, „kiaušiniai“, „nepagautos pelytės“ ir taip pat jų blokus su = 0 įstatyti. Iš „išvaizda“ ikelti „sakyk“ parašyti „pagauti tik pelytes“ ir iš „įvykiai“ ikelti „skelbti“ įvesti naują žinutę pavadinimu „pradžia“ ir taip pat įdėti į šį bloką 4 pav.



4 pav. Sukurti kintamuosius

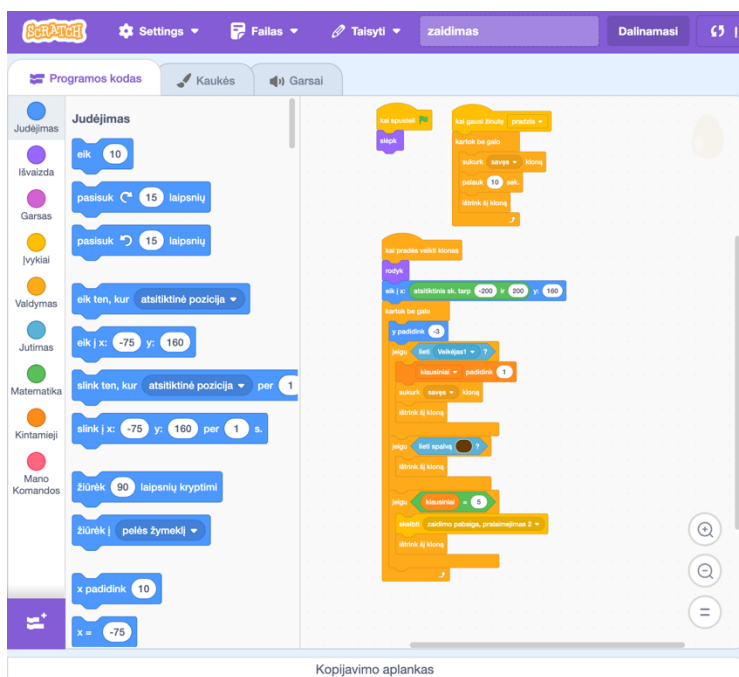
5. Kai pelyte spusteli žalią rodyklėlę - turi pasislepti. Kai tik ji gauna žinutę „pradžia“, tai mes galime įkelti iš „valdymas“ „kartok be galo“. Į jį įdėti „sukurk savęs kloną“, „palauk 15s“, „ištrink šį kloną“ 5 pav.



5 pav. Kai gaunama žinutė „pradžia“.

6. Kuriant kitą bloką iš „valdymas“, „kai pradės veikti klonas“, ir „išvaizda“ įkeliamo „rodyk“ iš „judėjimas“. Įkeliamo „eik į x: y: 160“. Į x vietą įdeda iš „matematika“ „atsitiktinis tarp -200 ir 200“. Toliau iš „valdymas“ įkeliamo „kartok be galo“. Į jį iš „judėjimas“ įkeliamo „y padidink -2“, tada iš „valdymas“

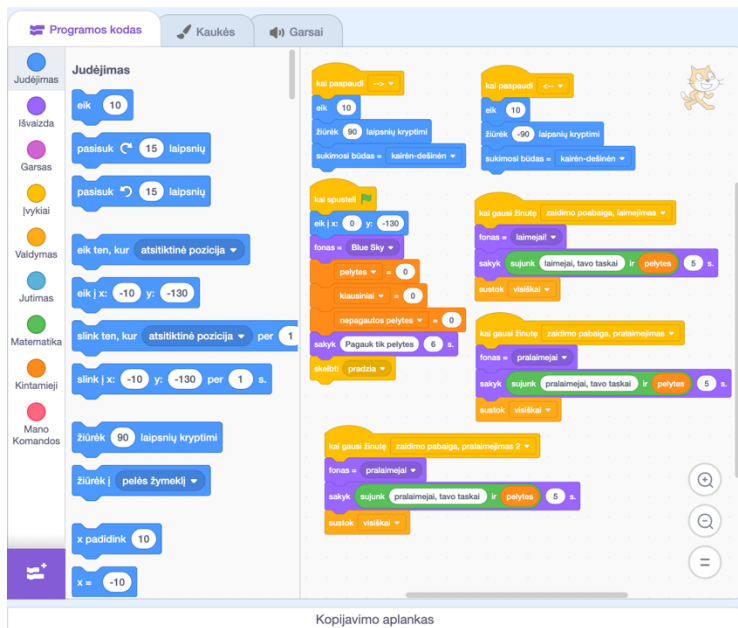
„jeigu” ir į jį iš „jutimas” įdedame „lieti” ir parenkame pagrindinį veikėją. Iš kintamųjų įkeliam „pelytes padidink 1” iš „valdymas” įkeliam „sukurk savęs kloną” ir „ištrink šį kloną”. Po juo „jeigu” įdedamas kitas „jeigu” iš „valdymas”, dabar iš „jutimas” įkeliam „lieti spalvą” ir parenkama fono apačios spalvą. Įkeliam „kintamieji” „nepagautos pelytes padidinti 1” ir iš „valdymas” „ištrink šį šabloną”
 Po šiuo „jeigu” įkeliamas dar vienas „jeigu” iš „valdymas”, tuomet į jį iš „matematika” įkeliamas rombas su lygybes ženklų, vienoje lygybes pusėje iš „kintamieji” įkeliam „pelytes” ir kitoje lygybes pusėje parašoma 10, čia bus taškai surinkti laimėjimui. Tuomet iš „įvykiai” sukuriamas naujas skelbimas ir skelbiama žaidimo pabaiga, laimejimas. Dar įkeliamas vienas „jeigu” iš „valdymas”, iš „matematika” įkeliamas dar vienas rombas su lygybe, vienoje lygybės pusėje iš „kintamieji” nepagautos pelytės, kitoje 5, tuomet įkeliam į jeigu kitą skelbimą sukurta „įvykiai” „žaidimo pabaiga, pralaimėjimas” ir iš „valdymas” įkeliam „ištrink šį kloną” 6 pav.



6 pav. Rinkti taškus laimėjimui arba pralaimėjimui.

Tas pats principas kaip ir su pelytėmis, tačiau skelbiant pralaimėjimą, reikia naują pralaimėjimo žinutę sukurti - „žaidimo pabaiga, pralaimėjai” 7 pav.

7. Iš „įvykiai” įkeliam „kai gausi žinutę „žaidimo pabaiga, laimejimas”” iš „išvaizda” įkeliam „fonas” ir pasirenkam laimejimo foną. Iš „išvaizda” įkeliam „sakyk” į jį iš „matematika” „sujunk ...laimejai, tavo taškai ... ir pelytes (iš kintamieji)”. 8 pav. Tas pats ir su kitom pralaimėjimo žinutėm, tik, foną keičiant reikia pasirinkti pralaimėjimo foną 9 pav.



7 pav. Žaidimo pabaiga



8 pav. Žaidimo laimėjimas

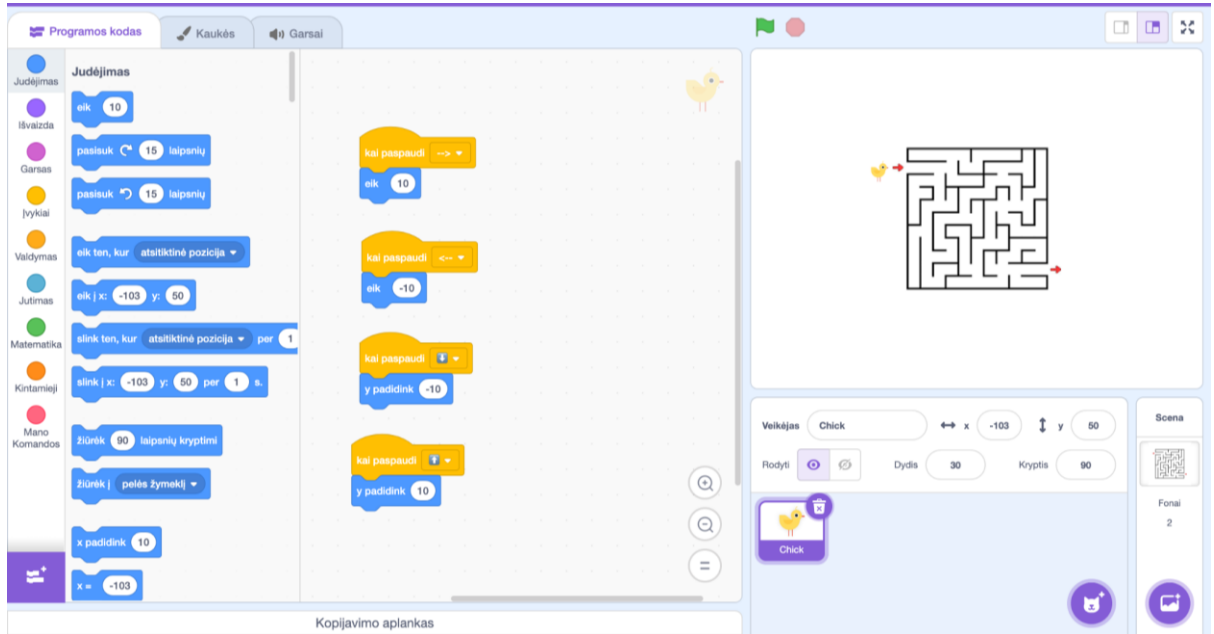


9 pav. Žaidimo pralaimėjimas

Užduotys diferencijavimui.

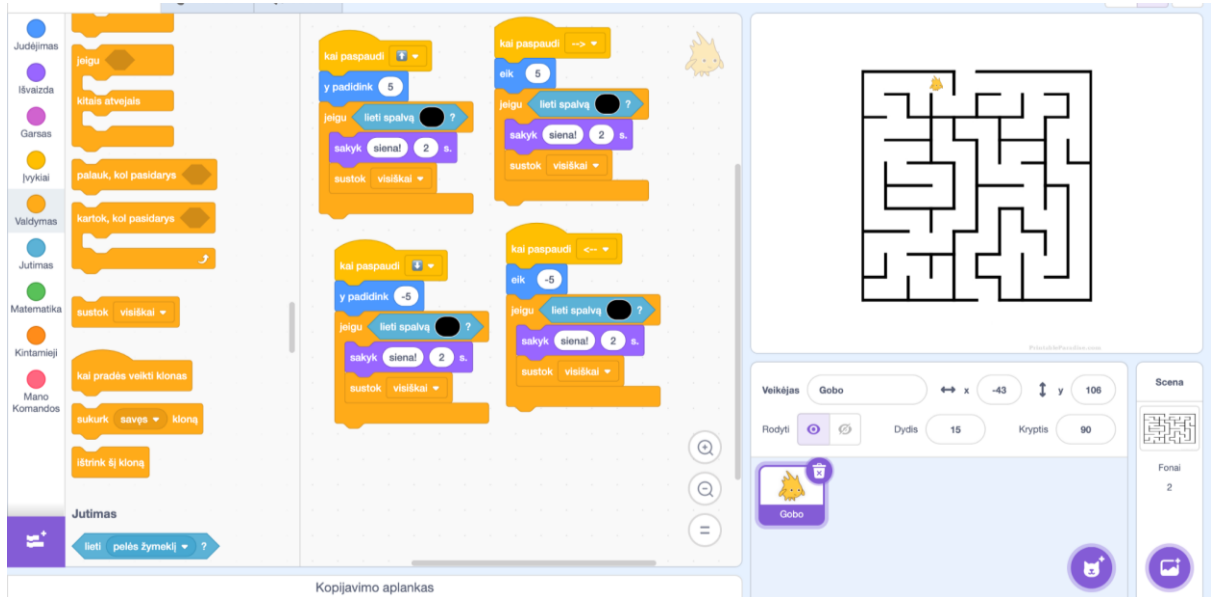
1. Labirintas

<https://scratch.mit.edu/projects/898435494>



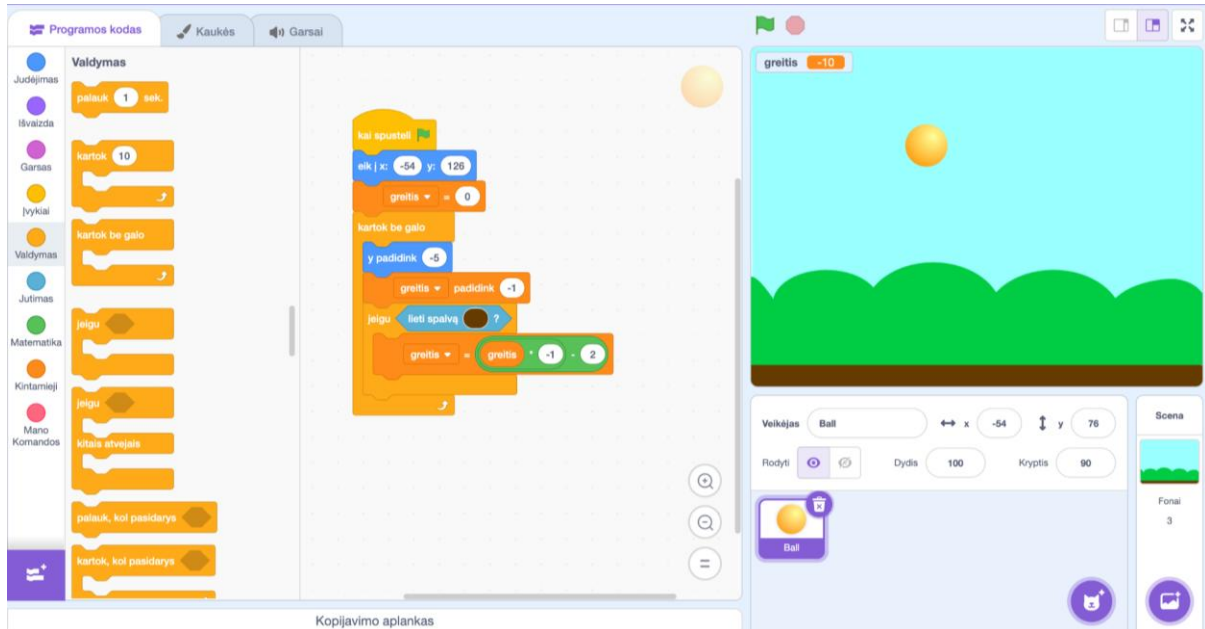
5. Labirintas

<https://scratch.mit.edu/projects/901698546>



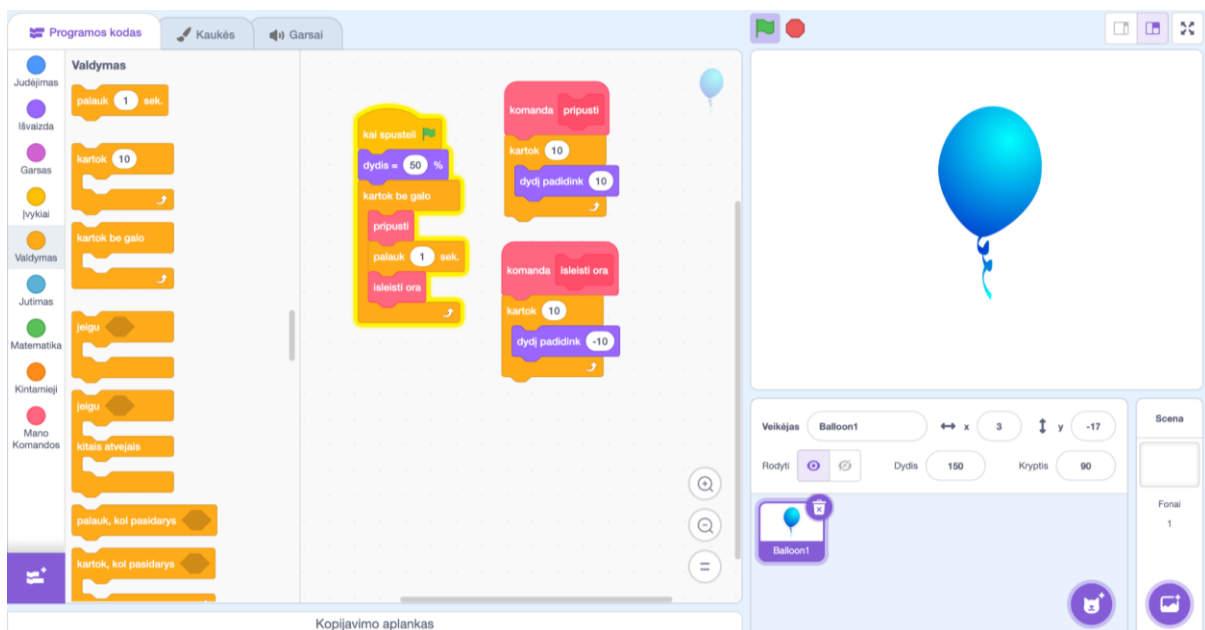
6. Kamuolys šokineja

<https://scratch.mit.edu/projects/901708838>



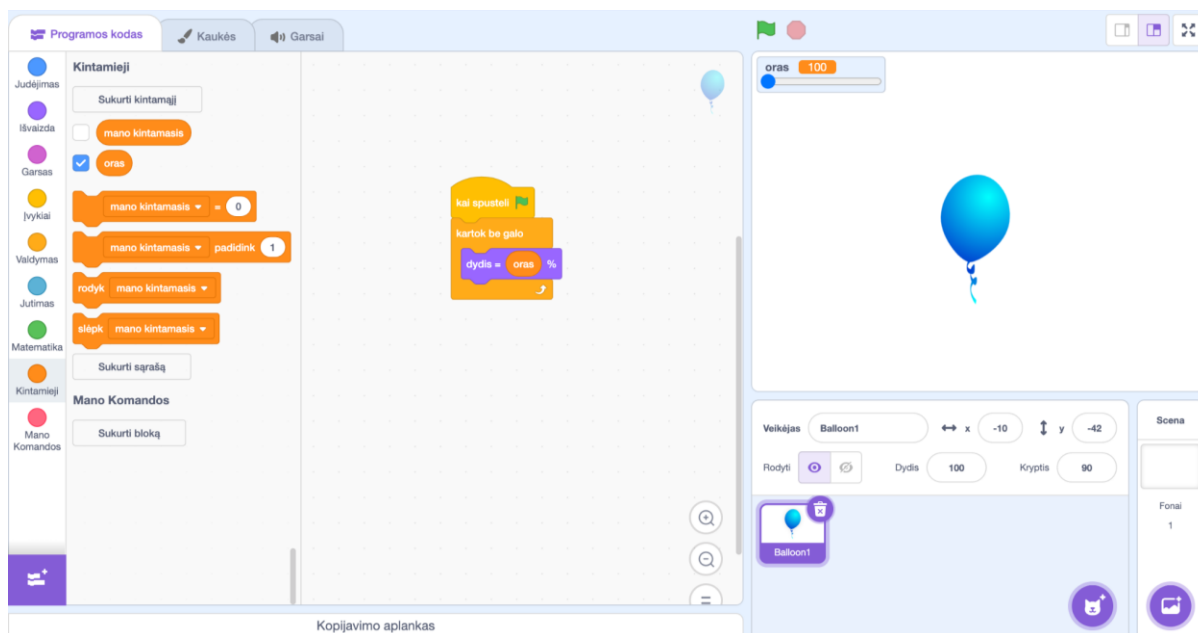
7. Balionas prisipučia ir išsileidžia.

<https://scratch.mit.edu/projects/901774983>



8. Baliono valdymas su linija viršuje.

<https://scratch.mit.edu/projects/901721275>



3 ETAPAS (25 minutės). Darbų pristatymas, aptarimas, vertinimas.

Mokiniai pristato sukurtus žaidimus. Galima sukurtus žaidimus kitiems klasės draugams pažaisti. Mokiniai rašo atsiliepimus „Mentimeter“ apie sukurtus žaidimus, įvardija sunkumus ir sėkmes. Aptarimas.

Algoritmai, jų savybės, užrašymo būdai (Dalia Balčiūnienė)

Pasiekimų sritis	Algoritmai ir programavimas
Klasė	7
Tema	Algoritmai, jų savybės, užrašymo būdai.
Integruojami dalykai	Matematika, lietuvių kalba.

Kompetencijos	<p>Pažinimo: susipažįsta su klasikiniiais algoritmais, išvelgia, kaip panaudoti algoritmavimą sprendžiant realaus gyvenimo uždavinius.</p> <p>Skaitmeninė: kuria skaitmeninį turinį ir juo dalijasi.</p> <p>Kūrybiškumo: mokiniai kūrybiškai taiko išmaniąsias technologijas, skatinama įvairiapusė mokinių veikla.</p> <p>Komunikavimo: darbo pristatymas, diskusija.</p>			
Tikslas	Pasirinkta tema ir skaitmenine priemone užrašyti algoritmą			
Uždaviniai	<p>aptarti algoritmo sąvoką, savybės, pateikimo būdus;</p> <p>sukurti algoritmą pasirinktam uždaviniui išspręsti;</p> <p>pristatyti darbą, įsivertinti.</p>			
Planuojamas rezultatas	<p>gebės savais žodžiais nusakyti algoritmo sąvoką, savybes, kaip galima juos pateikti;</p> <p>gebės kurti algoritmus, įvertins draugų darbus, įsivertins savo darbą pamokoje.</p>			
Specifinės priemonės / programinė įranga	<p>Žinių patikrinimui ir įtvirtinimui: wordwall.net</p> <p>Algoritmo kūrimui mokiniai gali rinktis: canva.com, mindmeister.com, teksto rengyklės SmartArt grafikas, scratch.mit.edu.</p> <p>Darbus įkelia: jamboard.google.com</p> <p>Refleksija: Google forms</p>			
Mokymosi metodai	Apversta klasė, darbas poromis, praktinis darbas kompiuteriu, minčių žemėlapis, schemų braižymas, diskusija.			
Mokinių pasiekimų lygių požymiai	(1)	(2)	(3)	(4)
	Naudodamasis kitų pagalba susipažįsta su algoritmo rengimo aplinka (B2.1).	Naudodamasis pagalba atlieka nurodytus veiksmus, randa reikiamas komandas (B2.2).	Randa reikiamas komandas paaiškina algoritmo vykdymo eigą, parodo rezultatus (B2.3).	Be kitų pagalbos randa ir taiko algoritmui kurti tinkamas programas, komandas, paaiškina vykdymo eigą demonstruoja rezultatus. (B2.4).
Žinios prieš	Mokiniai geba naudotis skaitmeninėmis priemonėmis, kurių pagalba kuria algoritmus, yra susipažinę su algoritmo sąvoka, savybėmis, pavyzdžiais.			

Galimybės taikyti spec. poreikių mokiniams	Konsultuojasi su mokytoju. Pateikiamos instrukcijos, pavyzdžiai. Vykdo mokytojo užrašytus algoritmus pažingsniui, Aukštesnių gebėjimų mokiniai gali rašyti algoritmą programavimo aplinkoje.
Patarimai kolegoms, kurie naudos parengtą medžiagą, galimos rizikos	Įkeltos nuorodos mokiniams su informacija apie algoritmus savarankiškam mokymuisi, kad galėtų pasiruošti pamokai. Laiko vadyba, nepasiruošę pamokai mokiniai, neišbandytos programos, Paruoštos užduotys pasirinkimui ir papildomos užduotys gabiems mokiniams.

Mokomoji veikla pamokoje

Mokomoji veikla pamokoje suteikia mokiniams galimybę įgyti žinių ir įgūdžių, skatina kritiškai mąstyti ir analizuoti informaciją. Tai padeda jiems suprasti sudėtingas sąvokas, priimti sprendimus, ugdyti bendradarbiavimo įgūdžius.

Sėkminga mokomoji veikla gali padidinti mokinių mokymosi motyvaciją, pasitikėjimą savimi. Egzistuoja daugybė įvairių mokomosios veiklos formų ir metodų, kuriuos mokytojai gali naudoti savo pamokose. Planuojant mokomąją veiklą galima rinktis darbą grupėmis, durstinio metoda, abipusį mokymąsi, praktinį tyrimą, naudoti skaitmeninius mokymosi objektus. Svarbu pasirinkti tokią mokomąją veiklą, kuri atitiktų mokinių poreikius ir gebėjimus. Ji turėtų būti įdomi ir patraukli mokiniams.

Pradedant mokyti naujus dalykus, reikia išsiaiškinti, kas jau žinoma naujai dėstoma tema ir sudominti mokinius būsima veikla, kad jie emociškai įsitrauktų į temos nagrinėjimą. Norint pasiekti šį tikslą, gali būti taikomi įvairūs metodai, pavyzdžiui, minčių lietus, vaizdo įrašų demonstravimas, pristatymų ar skaitmeninių mokymosi priemonių naudojimas [11].

Galima pamokos „Algoritmai, jų savybės, užrašymo būdai“ mokomoji veikla pateikta 2 lentelėje.

1 lentelė. Mokomoji veikla.

Eil. nr.	Mokomoji veikla	Skiriamas pamokos laikas, min.
1.	Pasiruošimo pamokai patikra wordwall.net (1 priedas), pamokos uždavinių pristatymas, svarbiausių dalykų akcentavimas. Sudominimas. Japoniškas logikos testas.	10 min.



1 pav. Japoniškas logikos testas

Užduotis: Perkelti visus žmones per upę į kitą krantą, laikantis šių sąlygų:

Vaikai negali plaukti be suaugusiųjų priežiūros.

Kalinys negali būti paliktas vienas su niekuo kitu, išskyrus šerifą.

Vyras negali palikti savo sūnų su moterimi be priežiūros, o moteris negali palikti savo dukterų su vyrais be priežiūros.

Plauste turi būti bent vienas žmogus, bet ne daugiau nei du [12].

Algoritmo sprendimą rasite nurodytame šaltinyje [13].

2.

Mokiniai poromis atlieka mokytojo pasiūlytas užduotis.

Padalomoji medžiaga. Algoritmų temos:

Kaip algoritmai veikia mūsų namuose?

Kaip algoritmai veikia mūsų buitinius prietaisus?

Algoritmas geometrinės figūros braižymui ir perimetro radimui. (Stačiakampis)

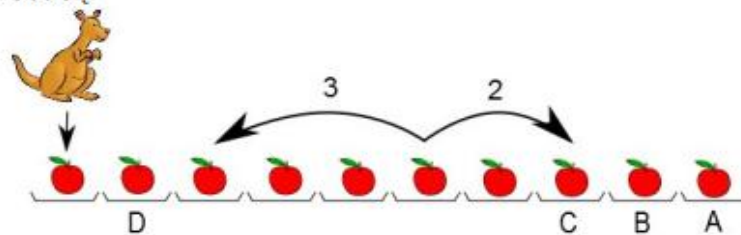
Medelio sodinimo algoritmas.

Visų obuolių surinkimo algoritmas.

20 min.



Eilėje yra 10 lėkščių. Kiekvienoje lėkštėje yra po vieną obuolį. Kengūriukas Tomas mėgsta šokinėti. Pirmiausia jis šoka prie kairiojo krašto esančios lėkštės. Po to kiekvienu šuoliu jis gali šokti arba per dvi lėkštes pirmyn, arba per tris lėkštes atgal. (Paveikslėlyje rodyklės vaizduoja galimus šuolius nuo vienos iš lėkščių.) Tomas šoka tik prie lėkščių, kuriose yra obuolių. Prišokęs prie lėkštės, pasiima iš jos obuolį.



Jei Tomas susirinks visus obuolius, kurį paims paskutinį?

2 pav. Obuolių surinkimo algoritmas. Bebras.lt

Algoritmas, kurį atlikus rastume kriaušaičių augančių sode procentą.

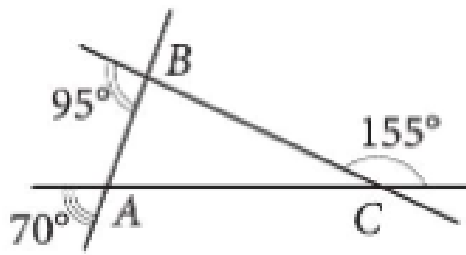
23. Sode auga 180 vaismedžių. Paveikslėlyje pavaizduota, kiek sode auga obelaidžių, kiek – kriaušaičių ir kiek – slyvaičių.

81	45	54
Obelaitės	Kriaušaitės	Slyvaitės

3 pav. Sodas. <https://evadoveliai.lt/demo/978-9955-879-65-7D/index.html>

Elektroninio laiško rašymo algoritmas.

Algoritmas, kuris apskaičiuotų trikampio ABC kampų dydžių sumą.



4 pav. Trikampis. <https://evadoveliai.lt/demo/978-9955-879-65-7D/index.html>

Pora kuria algoritmą pasirinktu būdu ir tinkama skaitmenine priemone. Pavyzdžiui:

schemos sudarymas;

veiksmų kortelių dėliojimas;

paveikslėliais;

programa kompiuterinėje programavimo aplinkoje;

pasakojimo kūrimas.

Atliktų darbų nuorodas įkelia į <https://jamboard.google.com/>

	<p>Darbų pristatymas, įsivertinimas.</p> <p>Klausimai refleksijai:</p> <p>Kokių naujų žinių ir įžvalgų įgavai atlikdamas šią užduotį? Kodėl šios žinios tau atrodo vertingos?</p> <p>Su kokiais didžiausiais iššūkiais susidūrei atlikdamas užduotį? Kaip juos įveikei?</p> <p>Kaip tavo indėlis prisidėjo prie bendro darbo rezultato? Kokie tavo pasiekimai labiausiai džiugina?</p> <p>Kokį parašysi sau pažymį? Kodėl?</p> <p>Ką galiu padaryti geriau kitą kartą atlikdamas užduotį?</p> <p>Refleksijos klausimai pateikiami „Google“ formoje</p>	15 min
--	--	--------

Kyla problemų peržiūrint arba pateikiant šią formą?

UŽPILDYTI „GOOGLE“ FORMOSE

Pakviečiau jus užpildyti formą:

Algoritmai jų savybės, užrašymo būdai.

Klausimai refleksijai:

Vardas, klasė, grupė

Ką naujo sužinojai atliekes užduotį? Kodėl?

Kas buvo sunkiausia atliekant užduotį?

Koks tavo indėlis bendrame darbe?

- 100%
- daugiau nei 75%
- apie 50%
- mažiau nei 25%

Kita:

Ką galiu padaryti geriau kitą kartą atlikdamas užduotį?

Kompetencijų vertinimo rubrika

Švietimo, mokslo ir sporto ministerija patvirtino bendrąsias ugdymo programas, kuriose numatyta, kad mokyklos turėtų pereiti prie kompetencijomis grįsto mokymosi. Tuo siekiama, kad mokiniai įgytų ne tik žinias, bet ir įgūdžius, nuostatas ir vertybes, reikalingas sėkmei įvairiose gyvenimo srityse. Skaitmeninės ir komunikavimo kompetencijos vertinimas pateiktas 3 lentelėje.

2 lentelė. Kompetencijų rubrika.

Kriterijai	Slenkstinis (35–44 %)	Patenkinamas (45-64%)	Pagrindinis (65-84%)	Aukštesnysis (85-100%)
Skaitmeninė kompetencija				
Skaitmeninis turinys: kuria įvairių formų ir formatų skaitmeninį turinį ir dirba su juo.	Naudojasi mokytojo siūlomais skaitmeninio turinio kūrimo įrankiais. Mokytojo padedamas kuria paprastą skaitmeninį turinį.	Naudojasi mokytojo siūlomais skaitmeninio turinio kūrimo įrankiais, savarankiškai pasirenka paprasto, nesudėtingo skaitmeninio turinio kūrimo ir sudarymo būdus, kuria paprastą skaitmeninį turinį..	Geba savarankiškai pasirinkti skaitmeninio turinio kūrimo priemones, savarankiškai kuria paprastą, nesudėtingą skaitmeninį turinį.	Geba argumentuotai pasirinkti algoritmo kūrimo priemones. Savarankiškai atlikdamas užduotis, nurodo, kaip kurti ir redaguoti įvairaus formato turinį; kuria skaitmeninį turinį.
Komunikavimo kompetencija				
Skatinamas sumanumas, kūrybiškumas, gebėjimas taikliai reikšti mintis, generuoti idėjas.	Supranta kai kurių užduočių sąlygas, savais žodžiais bando paaiškinti užduočių sprendimo būdus. Mokytojo padedamas komentuoja savo veiksmus žodžiu, atsako į pateiktus klausimus.	Supranta daugelio užduočių sąlygas, savais žodžiais bando paaiškinti užduočių sprendimo būdus. Mokytojo padedamas komentuoja savo veiksmus žodžiu, atsako į pateiktus klausimus,	Savais žodžiais paaiškina skaitmeninio turinio kūrimo pasirinktus būdus ir priemones. Supranta užduočių sąlygas, savais žodžiais paaiškina užduočių sprendimo būdus. Tinkamai komentuoja savo veiksmus žodžiu. Daugeliu atvejų taisyklingai atsako į pateiktus klausimus.	Argumentuotai geba pasirinkti skaitmeninio turinio kūrimo būdus ir priemones. Keisdamasis patirtimi beveik neklysdamas naudoja taisyklingus terminus ir sąvokas. Pasiūlo galimus užduoties sprendimo būdus. Tinkamai komentuoja savo veiksmus žodžiu ir raštu. Taisyklingai atsako į pateiktus klausimus

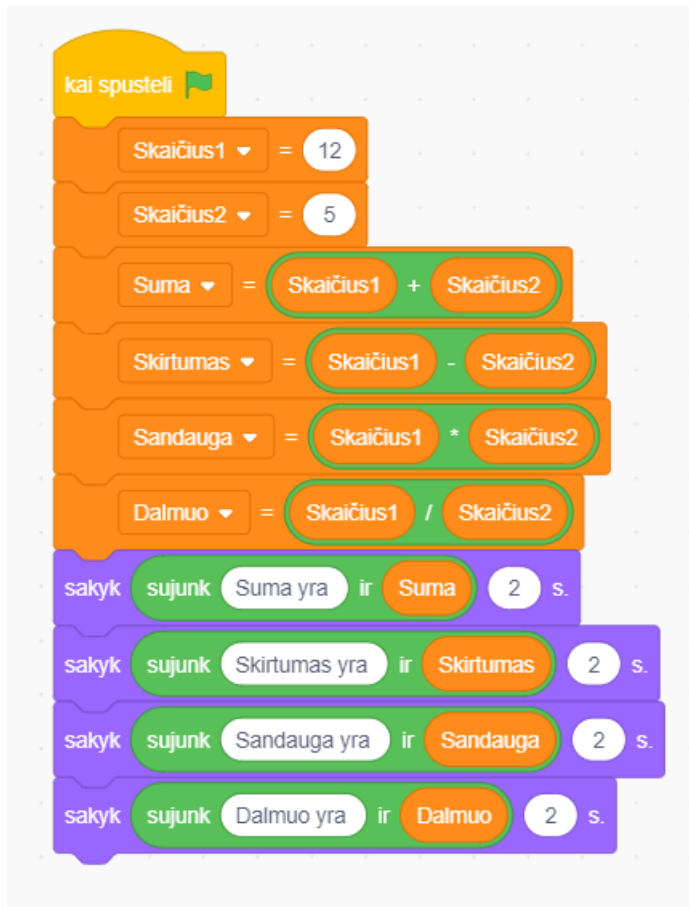
Nuosekliai atliekami veiksmai (tiesiniai algoritmai) (Neringa Kušlienė)

Pasiekimų sritis	B. Algoritmai ir programavimas
Klasė	7-8 (2 pamokos)
Tema	Nuosekliai atliekami veiksmai (tiesiniai algoritmai)
Integruojami dalykai, pasiekimai	Matematika, technologijos, lietuvių k.
Kompetencijos	Pažinimo – turi žinių apie tiesinius algoritmus, kritiškai mąsto atlikdami užduotis. Skaitmeninė – savarankiškai ir mokytojo padedamas kuria skaitmeninį turinį programuodamas. Komunikavimo - naudojantis skaitmeninėmis priemonėmis, skatinamas sumanumas, gebėjimas taikliai reikšti mintis, generuoti idėjas.
Tikslas	Suprasti nuosekliai atliekamų veiksmų loginę seką ir įtvirtinti įgūdžius sprendžiant uždavinius.
Uždaviniai	9. Išsiaiškinsite, kokie algoritmai vadinami tiesiniais. 10. Išmoksite pradinius duomenis įvesti klaviatūra. 11. Išmoksite atlikti pagrindinius aritmetinius veiksmus su sveikaisiais skaičiais. 4. Išmoksite gautus rezultatus pateikti ekrane.
Planuojamas rezultatas	8. Gebėsite paaiškinti kokiomis savybėmis pasižymi tiesiniai algoritmai. 9. Gebėsite įvesti pradinius duomenis klaviatūra. 10. Gebėsite atlikti aritmetinius veiksmus su sveikaisiais skaičiais. 11. Gebėsite pateikti gautus rezultatus ekrane.
Specifinės priemonės / programinė įranga	Multimedia projektorius, išmanioji lenta Viktorinai Mentimeter, Quizizz, Socratic. Kahoot Scratch programavimo aplinka Padlet virtuali lenta Python programavimo aplinka internete: https://www.onlinegdb.com/ „Python“ programavimo kalbos vadovėlis: https://www.emokykla.lt/titulinis/-python-programavimo-kalbos-vadovelis/47343 Išmokti programuoti žaidžiant: https://angis.lrt.lt/#/
Mokymosi metodai	Darbas grupėmis, diskusija.
Mokinių atlikto darbo vertinimas ir įsivertinimas	<i>Slenkstinis</i> – padedamas naudojasi programavimo kalbos konstrukcijomis, atkartoja mokytojo veiksmus programavimo kalboje (B2.1). <i>Patenkinamas</i> – užrašo kelių eilučių algoritmus naudodamas programavimo konstrukcijas ir padedamas naudojasi programavimo aplinka (B2.2).

	<p><i>Pagrindinis</i> – sprendamas problemas naudoja programavimo kalbos konstrukcijas ir aplinką (B2.3).</p> <p><i>Aukštesnysis</i> – randa ir išbando įvairias programavimo aplinkos funkcijas, eksperimentuoja su programavimo kalbos konstrukcijomis (B2.4).</p>
Žinios prieš	Moka naudotis Padlet virtualia lenta, žino „Scratch“ aplinkos struktūrą, funkcijas, moka pasirinkti veikėjus, jų kaukes, fonus ir pan.
Galimybės taikyti spec. poreikių mokiniams	Mokytojas scenarijų pritaiko priklausomai nuo vaiko specialiųjų poreikių. Pateikiamas paruoštas kodas, kurį mokiniai atkartoja programavimo aplinkoje.
Patarimai kolegoms, kurie naudos parengtą medžiagą	<ol style="list-style-type: none"> 1. Patys atlikite visas scenarijuje numatytas užduotis. 2. Scenarijų adaptuokite skirtingų gebėjimų mokiniams. 3. Papildykite mokiniams teikiamus šaltinius įvairesniais. 4. Per pamoką mokiniai dirba savarankiškai, mokytojas yra pagalbininkas ir konsultantas.

1 ETAPAS (7 min.). Pamokos temos pristatymas. „Scratch“ aplinkoje sukurto skaičiuotuvo pristatymas.

Mokytoja išmanioje lentoje pristato skaičiuotuvą, sukurtą naudojant „Scratch“ aplinką.

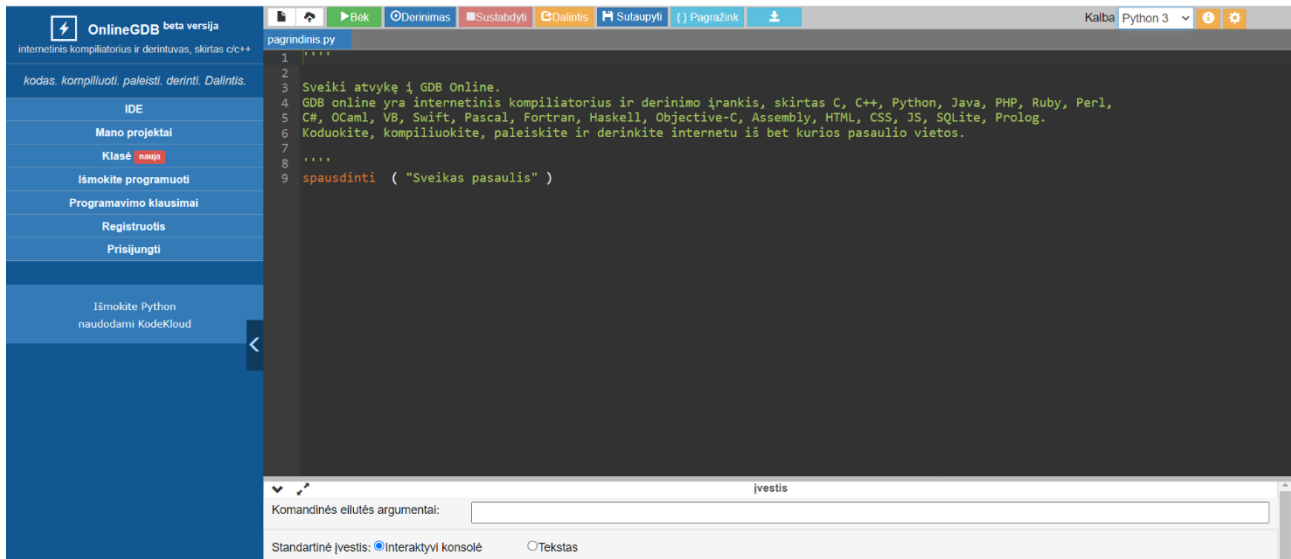


Skaičius1	12
Skaičius2	5
Suma	17
Skirtumas	7
Sandauga	60
Dalmuo	2.4



2 ETAPAS (15-20 min.). Skaičiuotuvo kūrimas „Python“ programavimo kalba.

Programavimo aplinka <https://www.onlinegdb.com/>



```
main.py
1 Skaičius1 = 12
2 Skaičius2 = 5
3 Suma = Skaičius1 + Skaičius2
4 Skirtumas = Skaičius1 - Skaičius2
5 Sandauga = Skaičius1 * Skaičius2
6 Dalmuo = Skaičius1 / Skaičius2
7 print("Suma: ", Suma)
8 print("Skirtumas: ", Skirtumas)
9 print("Sandauga: ", Sandauga)
10 print("Dalmuo: ", Dalmuo)
```

Rezultatas ekrane:

```
Suma: 17
Skirtumas: 7
Sandauga: 60
Dalmuo: 2.4
```

Pasunkiname uždavinį: pradinius duomenis įvedame klaviatūra

```
main.py
1 Skaičius1 = int(input("Įveskite pirmąjį skaičių: "))
2 Skaičius2 = int(input("Įveskite antrąjį skaičių: "))
3
4 Suma = Skaičius1 + Skaičius2
5 Skirtumas = Skaičius1 - Skaičius2
6 Sandauga = Skaičius1 * Skaičius2
7 Dalmuo = Skaičius1 / Skaičius2
8
9 print(Skaičius1, " + ", Skaičius2, " = ", Suma)
10 print(Skaičius1, " - ", Skaičius2, " = ", Skirtumas)
11 print(Skaičius1, " * ", Skaičius2, " = ", Sandauga)
12 print(Skaičius1, " / ", Skaičius2, " = ", Dalmuo)
```

Rezultatas ekrane:

```
Įveskite pirmąjį skaičių: 15
Įveskite antrąjį skaičių: 18
15 + 18 = 33
15 - 18 = -3
15 * 18 = 270
15 / 18 = 0.8333333333333334
```

Parodome, kaip atlikti veiksmus su realiaisiais skaičiais

```
main.py
1 Skaičius1 = float(input("Įveskite pirmąjį skaičių: "))
2 Skaičius2 = float(input("Įveskite antrąjį skaičių: "))
3
4 Suma = Skaičius1 + Skaičius2
5 Skirtumas = Skaičius1 - Skaičius2
6 Sandauga = Skaičius1 * Skaičius2
7 Dalmuo = Skaičius1 / Skaičius2
8
9 print(Skaičius1, " + ", Skaičius2, " = ", Suma)
10 print(Skaičius1, " - ", Skaičius2, " = ", Skirtumas)
11 print(Skaičius1, " * ", Skaičius2, " = ", Sandauga)
12 print(Skaičius1, " / ", Skaičius2, " = ", Dalmuo)
```

Rezultatas ekrane:

```
Įveskite pirmąjį skaičių: 5.5
Įveskite antrąjį skaičių: 2.5
5.5 + 2.5 = 8.0
5.5 - 2.5 = 3.0
5.5 * 2.5 = 13.75
5.5 / 2.5 = 2.2
```

3 ETAPAS (10-15 min.). Uždavinių sprendimas

Uždaviniai:

- Pietų kaina.** Trys draugai Karolis, Edgaras ir Dainius pietavo kavinėje. Karolio pietūs kainavo m eurų, Edgardo – k eurų, o Dainiaus – j eurų. Parašykite programą, skaičiuojančią, kiek eurų s iš viso sumokėta.

Pradiniai duomenys įvedami klaviatūra	Rezultatai rodomi ekrane
Kiek eurų sumokėjo Karolis? 5	Iš viso sumokėta eurų: 18
Kiek eurų sumokėjo Edgaras? 8	
Kiek eurų sumokėjo Dainius? 5	

```
1 m = int (input("Kiek eurų sumokėjo Karolis? "))
2 k = int (input ("Kiek eurų sumokėjo Edgaras? "))
3 j = int (input("Kiek eurų sumokėjo Dainius? "))
4 |
5 suma = m + k + j
6
7 print (f"Iš viso sumokėta eurų: {suma}")
```

Rezultatas ekrane:

```
Kiek eurų sumokėjo Edgaras? 8
Kiek eurų sumokėjo Dainius? 5
Iš viso sumokėta eurų: 18
```

2. **Skola.** Matas iš turimų k eurų Rimui paskolino r eurų, o Tomui - s eurų. Parašykite programą skaičiuojančią, kiek eurų j liko Matui.

Pradiniai duomenys įvedami klaviatūra	Rezultatai rodomi ekrane
Kiek eurų turėjo Matas? 10	Matui liko eurų: 4
Kiek eurų paskolino Rimui? 4	
Kiek eurų paskolino Tomui? 2	

```

1  k = int (input ("Kiek eurų turėjo Matas? "))
2  r = int (input ("Kiek eurų paskolino Rimui? "))
3  s = int (input ("Kiek eurų paskolino Tomui? "))
4
5  j = k - r - s
6
7  print (f"Matui liko eurų: {j}")

```

Rezultatas ekrane:

```

Kiek eurų turėjo Matas? 10
Kiek eurų paskolino Rimui? 4
Kiek eurų paskolino Tomui? 2
Matui liko eurų: 4

```

3. **Stačiakampis.** Stačiakampio ilgis yra lygus i centimetrų, plotis – j centimetrų. Parašykite programą, skaičiuojančią stačiakampio perimetrą prm ir plotą plt .

Pradiniai duomenys įvedami klaviatūra	Rezultatai rodomi ekrane
Koks stačiakampio ilgis? 5	Stačiakampio perimetras: 14
Koks stačiakampio plotis? 2	Stačiakampio plotas: 10

```

1  i = int(input("Koks yra stačiakampio ilgis? "))
2  j = int(input("Koks yra stačiakampio plotis? "))
3
4  prm = i*2 + j*2
5
6  plt = i*j
7
8  print(f"Stačiakampio perimetras: {prm}")
9  print(f"Stačiakampio plotas: {plt}")

```

Rezultatas ekrane:

```

Koks yra stačiakampio ilgis? 5
Koks yra stačiakampio plotis? 2
Stačiakampio perimetras: 14
Stačiakampio plotas: 10

```

4 ETAPAS (5 minutės). Darbų vertinimas, įsivertinimas, refleksija.

<https://wordwall.net/resource/20102418/python-quiz>

Mokiniai rašo atsiliepimus Mentimeter apie sukurtus skaičiuotuvus arba išspręstus uždavinius, įvardija sunkumus ir sėkmes. Aptarimas.

Tiesiniai algoritmai. Veiksmai su sveikaisiais skaičiais (Aldona Alkauskienė)

Pasiekimų sritis	(B) Algoritmai ir programavimas B2 Naudojasi algoritmų kūrimo, programavimo kalbos konstrukcijomis, programavimo aplinkomis. B2.3 Spręsdami problemas naudoja programavimo kalbos konstrukcijas ir aplinką.
Klasė	7-8 kl.
Tema	Tiesiniai algoritmai. Veiksmai su sveikaisiais skaičiais.
Integruojami dalykai, pasiekimai	Matematika, anglų kalba, lietuvių kalba.
Kompetencijos	Pažinimo – sveikųjų skaičių pritaikymas tiesiniuose algoritmuose.

	<p>Skaitmeninė – programavimo aplinkos tinkle arba skaitmeninių įrankių CodeBlocs, DevC++, taikymas, skaitmeninio turinio kūrimas programuojant.</p> <p>Komunikavimo – darbas grupėje, pagalba grupės nariui (-iams).</p> <p>Kūrybiškumo – matematinių veiksmų kūrimas bei sprendimas.</p>
Tikslas	Mokyti veiksmus su sveikaisiais skaičiais naudojant tiesinius algoritmus.
Uždaviniai	<ol style="list-style-type: none"> 1. Išnagrinėsite mokytojos pateiktą uždavinio pavyzdį ir užsirašysite C++ pagrindinius kodavimo elementus: išvedimą į ekraną, kintamojo ir reikšmės užrašymą ir priskyrimą, sveikųjų skaičių aritmetinius veiksmus. 2. Savarankiškai keisdami mokytojos pateiktą programos tekstą, pastebėsite programos keitimosi variantus. 3. Naudojant tiesinius algoritmus bei prisimindami matematikos kurso dalį apie sveikuosius skaičius ir sukursite programos teksto dalį, kuri gali apskaičiuoti matematinius veiksmus (išskyrus dalybą) ir patikrinsite.
Planuojamas rezultatas	<ol style="list-style-type: none"> 1. Gebėsite paaiškinti duomenų išvedimą į ekraną, kintamojo ir reikšmės užrašymą ir priskyrimą, sveikųjų skaičių aritmetinius veiksmus. 2. Mokėsite sukurti programos dalį, kuri gali skaičiuoti matematinius veiksmus su sveikaisiais skaičiais.
Specifinės priemonės / programinė įranga	<p>1. Visual Studio 2022 programa CodeBlocs, DevC++ arba programavimo (aplinkos internete) svetainės:</p> <ol style="list-style-type: none"> 1) https://www.w3schools.com/cpp/cpp_math.asp 2) https://www.onlinegdb.com/online_c++_compiler 3) https://www.programiz.com/cpp-programming/online-compiler/ https://cpp.sh/
Mokymosi metodai	Diskusijos metodas, praktinis tyrimas, darbas grupėmis (rekomenduojama po 2 mokinius, nebent mokinių skaičius yra nelyginis, tai vienoje grupėje bus 3 mokiniai).
Mokinių atlikto darbo vertinimas ir įsivertinimas	<p>Slenkstinis – su mokytojo pagalba užsirašo esminę informaciją apie sveikųjų skaičių veiksmus (išskyrus dalybą), perkopijuoja mokytojos pateiktą pavyzdį į programos langą ir ją įvykdo.</p> <p>Patenkinamas – su mokytojos pagalba. Pakeičia programos kodą, ją įvykdo.</p> <p>Pagrindinis – savarankiškai sukuria matematinius veiksmus (išskyrus dalybą) ir patikrina.</p>

	Aukštesnysis – savarankiškai sukuria uždavinio sąlygą (-as), kuriems reikia pritaikyti veiksmus su sveikaisiais skaičiais (išskyrus dalyba). Geba kitam mokiniui paaiškinti pamokos eigoje iškilusius sunkumus.
Žinios prieš	Moka aritmetinius veiksmus su sveikaisiais skaičiais atlikti, turi darbo C++ programavimo aplinkoje pagrindus, savarankiškai geba išsiversti angliškus žodžius (pvz. naudojantis <i>Google Translate</i>).
Galimybės taikyti spec. poreikių mokiniams	Mažesnis užduočių kiekis ir mokytojo pagalba iškilus sunkumams. Adaptuotos užduotys (pvz. su daliniu programos kodu).
Patarimai kolegoms, kurie naudosis parengta medžiaga	<ol style="list-style-type: none"> 1. Patys atlikite visas scenarijuje numatytas užduotis. 2. Scenarijų adaptuokite skirtingų gebėjimų mokiniams. 3. Per pamoką praktinį tyrimą mokiniai atlieka savarankiškai, mokytojas yra pagalbinkas ir konsultantas.

1 ETAPAS (10 min.). Pamokos plano pristatymas. Įvadas į pamoką.

Mokytojas pasisveikina ir pristato pamokos temą bei uždavinius. Diskusijos metodu mokytojas primina pagrindinius matematinius veiksmus (sudėtis, atimtis, daugyba) ir supažindina su C++ kodavimo pagrindiniais elementais. Diskusijos metu mokytojas kartu su mokiniais suranda C++ ir Scrach programavimo kalbų panašumus, užsipildo 1 lentelę.

1 lentelė. C++ kalbos reikšmių palyginimas su Scrach kalba

C++ kalba	Scrach kalba	Pastabos
#include <iostream>		
using namespace std;		
int		
int main () {}	Green Flag	
cout << „“ << endl;	Ask	
cin >> ;	Ask	
// arba /* */		
return	broadcast	

2 ETAPAS (10 min.). Pavyzdžio aiškinimas.

Mokytoja aiškina uždavinio pavyzdį (*1 pavyzdys*): programos kodo struktūrą, matematinių veiksmų modifikavimą bei pateikia pora pavyzdžių, kaip galima pritaikyti matematiniams veiksmams uždavinio sąlygą (*2 lentelė*). Atsako į iškilusias mokinių klausimus.

1 pavyzdys (veiksmai su sveikaisiais skaičiais)

//Duoti du sveikieji skaičiai. Apskaičiuoti jų sumą.

```
#include <iostream>
using namespace std;
int main()
{
    // Įvedimas
    int x, y, suma; // kintamųjų apibrėžimas
    cout << „Įveskite du skaičius“ << endl; // prašymo paaiškinimas
    cin >> x >> y; //skaičių įvedimas

    // Apskaičiavimai
    suma = x + y; //apskaičiavimas

    // Išvedimas
    cout << „Skaičių suma = “ << suma << endl; //išvedimas į ekraną
    return 0;
```

}

2 lentelė. Matematinų veiksmų ir galimų uždavinių sąlygų pritaikymo palyginimas.

Nr.	Matematiniai veiksmai	Uždavinys, kuriam galima pritaikyti išnagrinėtus matematinius veiksmus.
1.	$x + y$, kai $x = 7$, $y = 5$	Apskaičiuokite kiek yra klasėje mokinių, kai x yra mergaičių skaičius, o y – berniukų skaičius.
2.	$s + p + 2 \cdot u$, kai $s = 2$, $p = 3$, $u = 1$	Valgykloje sriuba kainuoja s eur. (sveikas skaičius), pagrindinis patiekalas – p eur. (sveikas skaičius), o užkandis u eur (sveikas skaičius). Apskaičiuokite, kiek moiniui reikės sumokėti, kai jis pirko sriubą ir pagrindinį patiekalą ir du užkandžius.
3.
4.

3 ETAPAS (20 min.). Praktinis darbas ir darbas grupėmis.

Mokytojas mokinius suskirsto poromis (jeigu klasėje nelyginis mokinių skaičius, tuomet viena grupė sudaryta iš 3 mokinių), paaiškina praktinį darbą. Mokiniai dirbdami poromis savarankiškai persikopijuoja mokytojos programos kodą, jį įvykdo. Modifikuoja programos tekstą, naują programą įvykdo, patikrina ar teisingai skaičiuojami matematiniai veiksmai. Pildo 2 lentelę ir kuria uždavinio (-ių) sąlygą (-as), kuriam (-iems) galima pritaikyti išnagrinėtus matematinius veiksmus

4 ETAPAS (5 min.) Refleksija.

Mokiniai užpildo refleksijos apklausą (pvz. VMA moodle, Google Forms ir pan.)

1. Parašykite po 3 sėkmingiausių ir 3 sunkiausių dalykus atliekant praktinį darbą.

3. Įvertinkite „Praktinio darbo“ užduotį:

Labai įdomi	Įdomi	Neįdomi	Labai neįdomi	Neturiu nuomonės
-------------	-------	---------	---------------	------------------

5. Parašykite, kur praktiškai panaudotumėte įgytas žinias ir įgūdžius.

Tiesiniai algoritmai. Sveikųjų skaičių dalyba (Aldona Alkauskienė)

Pasiekimų sritis	(B) Algoritmai ir programavimas B2 Naudojasi algoritmų kūrimo, programavimo kalbos konstrukcijomis, programavimo aplinkomis. B2.3 Spręsdami problemas naudoja programavimo kalbos konstrukcijas ir aplinką.
Klasė	7-8 klasė
Tema	Tiesiniai algoritmai. Sveikųjų skaičių dalyba.
Integruojami dalykai, pasiekimai	Matematika, anglų kalba, lietuvių kalba.
Kompetencijos	Pažinimo – sveikųjų skaičių pritaikymas tiesiniuose algoritmuose. Skaitmeninė – programavimo aplinkos tinkle arba skaitmeninių įrankių CodeBlocs, DevC++, taikymas, skaitmeninio turinio kūrimas programuojant. Komunikavimo – darbas grupėje, pagalba grupės nariui (-iams). Kūrybiškumo – uždavinių sprendimų kūrimas bei uždavinio sukūrimas.
Tikslas	Mokyti dalybos veiksmą su sveikaisiais skaičiais naudojant tiesinius algoritmus.
Uždaviniai	1. Išnagrinėsite mokytojos pateiktą uždavinio pavyzdį (sveikųjų skaičių dalybą) ir savarankiškai (keisdami programos tekstą) išspręsite mokytojos pateiktą (-us) uždavinį (-ius) 2. Dirbdami grupėmis (2-3 mokiniai) savarankiškai sukursite uždavinį ir išspręsite grupės nario sukurtą uždavinį.
Planuojamas rezultatas	1. Gebėsite paaiškinti sveikųjų skaičių dalybą ir ją taikyti sprendžiant ir kuriant uždavinius. 2. Mokėsite teorines žinias pritaikyti uždavinių sprendimuose ir uždavinio kūrimui.
Specifinės priemonės / programinė įranga	1. CodeBlocs, DevC++ skaitmeniniai įrankiai. arba programavimo (aplinkos internete) svetainės: 4) https://www.w3schools.com/cpp/cpp_math.asp 5) https://www.onlinegdb.com/online_c++_compiler 6) https://www.programiz.com/cpp-programming/online-compiler/ https://cpp.sh/

Mokymosi metodai	Diskusijos metodas, praktinis tyrimas, darbas grupėmis (rekomenduojama po 2 mokinius, nebent mokinių skaičius yra nelyginis, tai vienoje grupėje bus 3 mokiniai).
Mokinių atlikto darbo vertinimas ir įsivertinimas	<p>Slenkstinis – su mokytojo pagalba užsirašo esminę informaciją apie sveikųjų skaičių dalybos veiksmus, perkopijuoja mokytojos pateiktą pavyzdį į programos langą ir ją įvykdo.</p> <p>Patenkinamas – su mokytojos pagalba. Pakeičia programos kodą, ją įvykdo.</p> <p>Pagrindinis – savarankiškai sukuria uždavinių sprendimus.</p> <p>Aukštesnysis – savarankiškai paruošia savo sukurtą uždavinį, jo sprendimą bei išsprendžia kito mokinio uždavinį. Geba kitam mokiniui paaiškinti pamokos eigoje iškilusius sunkumus.</p>
Žinios prieš	Moka aritmetinius veiksmus su sveikaisiais skaičiais atlikti, turi darbo C++ programavimo aplinkoje pagrindus, savarankiškai geba išsiversti angliškus žodžius (pvz. naudojantis <i>Google Translate</i>).
Galimybės taikyti spec. poreikių mokiniams	<p>Pateikiama mokytojo sukurta programa, kurioje reikia įrašyti praleistas kodo vietas (1 priedas).</p> <p>Galima mokiniams skirti paprastus matematinius veiksmus atlikti (1 priedas). Jeigu mokiniui per sunku kurti uždavinį, jam galima pasiūlyti uždavinius spręsti.</p> <p>Mažesnis užduočių kiekis bei paprastesnės užduotys ir mokytojo pagalba iškilus sunkumams.</p> <p>Turėti papildomų ir sunkesnių uždavinių gabiems mokiniams (2 priedas)</p>
Patarimai kolegoms, kurie naudos parengtą medžiagą	<ol style="list-style-type: none"> 1. Patys atlikite visas scenarijuje numatytas užduotis (galima užsirašyti kelis skirtingus programos rezultatus). 2. Pamokos scenarijų adaptuokite skirtingų gebėjimų mokiniams. 3. Turėkite skaitmeninį įrankį, kuriame būtų galima sukelti mokinių kurtus uždavinius, naujus atradimus ir būtų galima pasidalinti patirtimi su visais mokiniais. 4. Per pamoką praktinį tyrimą mokiniai atlieka savarankiškai, mokytojas yra pagalbininkas ir konsultantas. 5. Galima apgalvoti apie patogų skaitmeninį įrankį mokinių kaupiamiesiems rinkti ir refleksijoms (pvz. VMA MOODLE).

1 ETAPAS (15 minučių):

Mokytojas pasisveikina ir pristato pamokos temą bei uždavinius. Diskusijos metodu mokytojas primina pagrindinius matematinius veiksmus: dalyba su sveikaisiais skaičiais ir nuosekliai pristato programos kodo 1 – 3 pavyzdžius.

1 pavyzdys	
Nuo metų pradžios praėjo d dienų. Parašykite programą, kuri apskaičiuotų, kiek savaitių s praėjo nuo metų pradžios.	
Pradiniai duomenys	Gauti rezultatai
Nuo metų pradžios praėjo dienų: $d = 15$	Įveskite kiek praėjo dienų nuo metų pradžios: 15 Nuo metų pradžios praėjo 2 savaitių.
Programos tekstas	
<pre>#include <iostream> #include <iomanip> using namespace std; int main() { setlocale(LC_ALL, "Lithuanian"); // Įvedimas int d, s; cout << "Įveskite kiek praėjo dienų nuo metų pradžios: "; cin >> d; // Apskaičiavimai 53 s = d / 7; // Išvedimas cout << "Nuo metų pradžios praėjo " << s << " savaitių." << endl; return 0; }</pre>	

2 pavyzdys	
Nuo metų pradžios praėjo d dienų. Parašykite programą, kuri apskaičiuotų ir parodytų kiek savaitių s ir dienų dd praėjo nuo metų pradžios.	
Pradiniai duomenys	Gauti rezultatai
Nuo metų pradžios praėjo dienų: $d = 15$	Įveskite kiek praėjo dienų nuo metų pradžios: 15 Nuo metų pradžios praėjo 2 sav. Ir 1 d.
Nuo metų pradžios praėjo dienų: $d = 25$	Įveskite kiek praėjo dienų nuo metų pradžios: 25 Nuo metų pradžios praėjo 3 sav. ir 4 d.
Programos tekstas	
<pre>#include <iostream> #include <iomanip> using namespace std; int main() { setlocale(LC_ALL, "Lithuanian");</pre>	

```

// Įvedimas
int d, s, dd;

cout << "Įveskite kiek praėjo dienų nuo metų pradžios: ";
cin >> d;

// Apskaičiavimai
s = d / 7;

dd = d % 7;
// Išvedimas
cout << "Nuo metų pradžios praėjo " << s << " sav. ir " << dd << " d." << endl;

return 0;
}

```

Mokiniai pabando savarankiškai atlikti 3 pavyzdį keisdami 2 pavyzdžio programos tekstą. Vėliau mokytojas pasidalina sprendimu ir paaiškina pagrindinius akcentus.

3 pavyzdys

Močiutė dalina n obuolių anūkam.

1. Parašykite programą, kuri apskaičiuotų po kiek obuolių nn gaus kiekvienas anūkas, kai anūkų yra a .
2. Papildykite programą: kiek močiutei liko m obuolių.

Pradiniai duomenys

Močiutė turi anūkų: $a = 5$
 Močiutė turi obuolių: $n = 16$

Gauti rezultatai

Įveskite kiek močiutė turi anūkų: 5
 Įveskite kiek močiutė turi obuolių: 16
 Kiekvienas anūkas gavo po 3 obuolių.
 Močiutei liko 1 obuol.

Močiutė turi anūkų: $a = 10$
 Močiutė turi obuolių: $n = 10$

Įveskite kiek močiutė turi anūkų: 10
 Įveskite kiek močiutė turi obuolių: 10
 Kiekvienas anūkas gavo po 1 obuolių.
 Močiutei liko 0 obuol.

Programos tekstas

```

#include <iostream>
#include <iomanip>
using namespace std;

int main() {
    setlocale(LC_ALL, "Lithuanian");
    // Įvedimas
    int a, n, nn, m;

    cout << "Įveskite kiek močiutė turi anūkų: ";
    cin >> a;

    cout << "Įveskite kiek močiutė turi obuolių: ";
    cin >> n;

    // Apskaičiavimai

```

```

nn = n / a;

m = n % a;
// Išvedimas
cout << "Kiekvienas anūkas gavo po " << nn << " obuolių. " << endl;
cout << "Močiutei liko " << m << " obuol." << endl;

return 0;
}

```

2 ETAPAS (25 minučių) Praktinis darbas ir darbas grupėmis.

Mokytojas paaiškina pamokos etapo 1 ir 2 alternatyvas, suskirsto mokinius grupėmis po 2. Jeigu klasėje nelyginis mokinių skaičius, tai vienoje grupėje bus 3 mokiniai.

Mokytojas pamokos etapo metu fiksuoja mokinių kaupiamuosius pasiekimus (pvz. teisingai atliktos užduotys, pagalba kitam mokiniui, išspręsti papildomi uždaviniai, sukurti uždaviniai ir pan.).

1 ALTERNATYVA	2 ALTERNATYVA
Praktinis darbas	Uždavinių sprendimas
<p>Dirbdami grupėse (2-3 mokiniai) sukuria savo uždavinį ir vėliau sprendžia kito klasioko sukurtą uždavinį. Esant poreikiu grupės nariai padeda vienas kitam.</p> <p>Greičiau atlikę mokiniai gali padėti kitiems klasiokams arba savarankiškai atlikti papildomas užduotis iš 2 priedo.</p> <p>Mokiniai fiksuoja savo pamokos kaupiamuosius pasiekimus ir informuoja mokytoją.</p>	<p>Dirbdami grupėse (2-3 mokiniai) sprendžia mokytojos pateiktas užduotis (3 priedas), esant poreikiui grupės nariai padeda vieni kitiems.</p> <p>Greičiau atlikę mokiniai gali padėti kitiems klasiokams arba savarankiškai atlikti papildomas užduotis iš 2 priedo.</p> <p>Mokiniai fiksuoja savo pamokos kaupiamuosius pasiekimus ir informuoja mokytoją.</p>

3 ETAPAS (5 minutės) Refleksija.

Mokiniai užpildo refleksijos apklausą (pvz. VMA moodle, Google Forms ir pan.)

1. Pažymėkite kurią alternatyvą pamokos metu pasirinkote:

1 alternatyva – praktinis darbas (kūriau uždavinį).	2 alternatyva – uždavinių sprendimas.	Kita
---	---------------------------------------	------

1.1. Jeigu pasirinkote „Kita“, parašykite savo atsakymą.

2. Parašykite po 3 sėkmingiausių ir 3 sunkiausių dalykus atliekant praktinį darbą (1 alternatyva) / sprendžiant uždavinius (2 alternatyva).

3. Įvertinkite „Praktinio darbo“ užduotį (1 alternatyva) / uždavinių sprendimą (2 alternatyva) (pasirinkite tinkamiausią):

Labai įdomi	Įdomi	Neįdomi	Labai neįdomi	Neturiu nuomonės
-------------	-------	---------	---------------	------------------

6. Parašykite, kur praktiškai panaudotumėte įgytas žinias ir įgūdžius.

1 priedas. Lengvesnių uždavinių pavyzdžiai.

1 uždavinys	
Miesto informatikos olimpiadoje dalyvavo n septintokų. Mokytoja nupirko m saldainių „Nomeda“ ir išdalijo mokiniams po lygiai. Saldainių neliko arba liko mažiau, negu yra mokinių. Po kiek saldainių s gavo kiekvienas mokinys ir kiek saldainių k liko mokytojai? Parašykite programą šiam uždaviniui spręsti.	
Pradiniai duomenys	Gauti rezultatai
Septintokų skaičius: n = 7 Saldainių skaičius: m = 23	Įveskite devintokų skaičių: 7 Įveskite mokytojos nupirktų saldainių kiekį: 23 Kiekvienas mokinys gavo po 3 saldainius. Mokytojai liko 2 saldainiai.
Programos tekstas	
<pre>#include <iostream> #include <iomanip> using namespace std; int main() { setlocale(LC_ALL, "Lithuanian"); // Įvedimas int n, m, s, k; cout << "Įveskite devintokų skaičių: "; cin >> ____; cout << "Įveskite mokytojos nupirktų saldainių kiekį: "; cin >> ____; // Apskaičiavimai s = ____ / ____; k = ____ % ____; // Išvedimas cout << "Kiekvienas mokinys gavo po " << s << " saldainius." << endl; cout << "Mokytojai liko " << k << " saldainiai." << endl; return 0; }</pre>	

2 uždavinys

Parašykite programą, kuri skaičiuotų šiuos matematinius veiksmus:

```
a = 18 % 3 + 7 / 3 * 2;  
z = (1 + 49) % 5 + 6 ;  
s = 17 % (3 + 7) / 3 * 2;  
d = 1 + 49 % (5 + 7) / 3;  
c = 5 % (3 + 7 / 3);  
p = 19 % 8 + 7 / 3;  
v = 5 + 49 % 5 + 6 / 3;  
f = 9 % (8 + 7 / 3);
```

Pradiniai duomenys

```
a = 18 % 3 + 7 / 3 * 2;  
z = (1 + 49) % 5 + 6 ;  
s = 17 % (3 + 7) / 3 * 2;  
d = 1 + 49 % (5 + 7) / 3;  
c = 5 % (3 + 7 / 3);  
p = 19 % 8 + 7 / 3;  
v = 5 + 49 % 5 + 6 / 3;  
f = 9 % (8 + 7 / 3);
```

Gauti rezultatai

```
a = 4  
z = 6  
s = 4  
d = 1  
c = 0  
p = 5  
v = 11  
f = 9
```

Programos tekstas

```
#include <iostream>  
#include <iomanip>  
using namespace std;  
  
int main() {  
    setlocale(LC_ALL, "Lithuanian");  
    // Įvedimas  
    int a, z, s, d, c, p, v, f, b;  
  
    // Apskaičiavimai ir išvedimai  
    a = 18 % 3 + 7 / 3 * 2;  
    cout << "a = " << a << endl;  
    z = (1 + 49) % 5 + 6 ;  
    cout << "z = " << z << endl;  
    s = 17 % (3 + 7) / 3 * 2;  
    cout << "s = " << s << endl;  
    d = 1 + 49 % (5 + 7) / 3;  
    cout << "d = " << d << endl;  
    c = 5 % (3 + 7 / 3);  
    cout << "c = " << c << endl;  
    p = 19 % 8 + 7 / 3;  
    cout << "p = " << p << endl;  
    v = 5 + 49 % 5 + 6 / 3;  
    cout << "v = " << v << endl;  
    f = 9 % (8 + 7 / 3);  
    cout << "f = " << f << endl;  
    return 0;  
}
```

2 priedas. Sunkių uždavinių pavyzdžiai.

1 Uždavinys*	
Duotas keturženklis skaičius n (n = 1234). Išskaidykite skaičių skaitmenimis ir apskaičiuokite jų sumą.	
Pradiniai duomenys	Gauti rezultatai
n = 1234	Įveskite keturženklį sk.n 1234 Skaitmenų suma = 10
n = 5623	Įveskite keturženklį sk.n 5623 Skaitmenų suma = 16
Programos tekstas	
<pre>#include <iostream> #include <iomanip> using namespace std; int main() { setlocale(LC_ALL, "Lithuanian"); // Įvedimas long n, a, b, c, d, sksum; cout << "Įveskite keturženklį sk.n" << endl; cin >> n; // Apskaičiavimai a = n / 1000; b = n / 100 % 10; c = n / 10 % 10; d = n % 10; sksum = a + b + c + d; // Išvedimas cout << "Skaitmenų suma = " << sksum << endl; return 0; }</pre>	

2 Uždavinys*

Duotas šešiaženklis skaičius n ($n = 123456$). Rasti: skaitmenų sandaugą, išbraukti po du kraštinius skaitmenis, sukeisti skaitmenis taip: 634125. Naudokite „long“ kintamiesiems aprašyti.

Pradiniai duomenys	Gauti rezultatai
$n = 123456$	Įveskite šešiaženklį sk.n 123456 Skaitmenų sandauga = 720 Išbr.du krašt.sk. = 34 Sukeitimas = 634125
$n = 365429$	Įveskite šešiaženklį sk.n 365429 Skaitmenų sandauga = 6480 Išbr.du krašt.sk. = 54 Sukeitimas = 954362

Programos tekstas

```
#include <iostream>
#include <iomanip>
using namespace std;

int main() {
    setlocale(LC_ALL, "Lithuanian");
    // Įvedimas
    long n, a, b, c, d, e, f, san, isb, suk;
    cout << "Įveskite šešiaženklį sk.n" << endl;
    cin >> n;

    // Apskaičiavimai
    a = n / 100000;
    b = n / 10000 % 10;
    c = n / 1000 % 10;
    d = n / 100 % 10;
    e = n / 10 % 10;
    f = n % 10;
    san = a * b * c * d * e * f;
    isb = c * 10 + d;
    suk = f * 100000 + c * 10000 + d * 1000 + a * 100 + b * 10 + e;

    // Išvedimas
    cout << "Skaitmenų sandauga = " << san << endl;
    cout << "Išbr.du krašt.sk. = " << isb << endl;
    cout << "Sukeitimas = " << suk << endl;
    return 0;
}
```

3 priedas. Vidutinio sunkumo uždavinių pavyzdžiai.

1 uždavinys	
<p>Miesto informatikos olimpiadoje dalyvavo n septintokų. Mokytoja nupirko m saldainių „Nomeda“ ir išdalijo mokiniams po lygiai. Saldainių neliko arba liko mažiau, negu yra mokinių. Po kiek saldainių s gavo kiekvienas mokinys ir kiek saldainių k liko mokytojai?</p>	
Pradiniai duomenys	Gauti rezultatai
<p>Septintokų skaičius: n = 7 Saldainių skaičius: m = 23</p>	<p>Įveskite devintokų skaičių: 7 Įveskite mokytojos nupirktų saldainių kiekį: 23 Kiekvienas mokinys gavo po 3 saldainius. Mokytojai liko 2 saldainiai.</p>
<p>Septintokų skaičius: n = 3 Saldainių skaičius: m = 27</p>	<p>Įveskite devintokų skaičių: 3 Įveskite mokytojos nupirktų saldainių kiekį: 27 Kiekvienas mokinys gavo po 9 saldainius. Mokytojai liko 0 saldainiai.</p>
Programos tekstas	
<pre>#include <iostream> #include <iomanip> using namespace std; int main() { setlocale(LC_ALL, "Lithuanian"); // Įvedimas int n, m, s, k; cout << "Įveskite devintokų skaičių: "; cin >> n; cout << "Įveskite mokytojos nupirktų saldainių kiekį: "; cin >> m; // Apskaičiavimai s = m / n; k = m % n; // Išvedimas cout << "Kiekvienas mokinys gavo po " << s << " saldainius." << endl; cout << "Mokytojai liko " << k << " saldainiai." << endl; return 0; }</pre>	

2 uždavinys	
<p>Kalėdų senelis, ruošdamasis lankyti vaikus, nupirko n dovanų. Jis aplankė k šeimų ir joms išdalino po dovaną. Likusias dovanas Kalėdų senelis nusprendė padalinti patiems geriausiems v vaikų po lygiai. Po kiek dovanų d gavo kiekvienas vaikas ir kiek dovanų m liko neišdalinta?</p>	
Pradiniai duomenys	Gauti rezultatai
<p>Dovanų skaičius: n = 20 Šeimų skaičius: k = 3 Gerų vaikų skaičius: v = 4 Geri vaikai gavo po dovanų</p>	<p>Įveskite dovanų skaičių: 20 Įveskite šeimų skaičių: 3 Įveskite gerų vaikų skaičių: 4 Kiekvienas geras vaikas gavo po 4 dovanas.</p>

	Kalėdų seneliui liko 1 dovanų.
Dovanų skaičius: $n = 28$ Šeimų skaičius: $k = 14$ Gerų vaikų skaičius: $v = 5$ Geri vaikai gavo po dovanų	Įveskite dovanų skaičių: 28 Įveskite šeimų skaičių: 14 Įveskite gerų vaikų skaičių: 5 Kiekvienas geras vaikas gavo po 2 dovanas. Kalėdų seneliui liko 4 dovanų.
Programos tekstas	
<pre> #include <iostream> #include <iomanip> using namespace std; int main() { setlocale(LC_ALL, "Lithuanian"); // Įvedimas int n, k, v, d, m; cout << "Įveskite dovanų skaičių: "; cin >> n; cout << "Įveskite šeimų skaičių: "; cin >> k; cout << "Įveskite gerų vaikų skaičių: "; cin >> v; // Apskaičiavimai d = (n-k)/v; m = (n-k) % v; // Išvedimas cout << "Kiekvienas geras vaikas gavo po " << d << " dovanas." << endl; cout << "Kalėdų seneliui liko " << m << " dovanų." << endl; return 0; } </pre>	

3 uždavinys	
Gimtadienis. Andrius septintojo gimtadienio proga gavo n balionų. Su draugais nusprendė balionus paleisti į dangų. Dalis pučiamų k balionų sprogo. Likusius balionus Andrius pasidalijo su d draugais po lygiai. Jeigu po dalybų dar liko balionų, tai juos pasiėmė Andrius. Po kiek balionų m gavo kiekvienas draugas ir kiek balionų a teko Andriui?	
Pradiniai duomenys	Gauti rezultatai
Balionų skaičius: $n = 77$ Susprogusių balionų skaičius: $k = 3$ Draugų skaičius: $d = 7$	Įveskite kiek yra balionų: 77 Įveskite kiek balionų susprogo: 3 Įveskite kiek yra draugų: 7 Kiekvienas draugas gavo po 10 balion. Andriui liko 4 balion.
Balionų skaičius: $n = 50$ Susprogusių balionų skaičius: $k = 8$ Draugų skaičius: $d = 5$	Įveskite kiek yra balionų: 50 Įveskite kiek balionų susprogo: 8 Įveskite kiek yra draugų: 5 Kiekvienas draugas gavo po 8 balion. Andriui liko 2 balion.

Programos tekstas

```
#include <iostream>
#include <iomanip>
using namespace std;

int main() {
    setlocale(LC_ALL, "Lithuanian");
    // Įvedimas
    int n, k, d, m, a;

    cout << "Įveskite kiek yra balionų: ";
    cin >> n;

    cout << "Įveskite kiek balionų susprogo: ";
    cin >> k;

    cout << "Įveskite kiek yra draugų: ";
    cin >> d;

    // Apskaičiavimai
    m = (n-k) / d;

    a = (n-k) % d;
    // Išvedimas
    cout << fixed << setprecision(2);
    cout << "Kiekvienas draugas gavo po " << m << " balion. " << endl;
    cout << "Andriui liko " << a << " balion." << endl;

    return 0;
}
```

Paprastas sąlyginis sakiny (Daiva Vaičiulionienė)

Pasiekimų sritis(remtasi Informatikos programa [1])	(B) Algoritmai ir programavimas 28.2.4. Algoritmų parinkimas. Aptariami klasikinių algoritmų pavyzdžiai (dalybos, paieškos, rikiavimo). Diskutuojama apie žinomų algoritmų naudojimą ir pritaikymą konkrečioms uždaviniais spręsti. Susipažįstama su algoritmų efektyvumo, optimizavimo problemomis.
Klasė	7-8 kl.
Tema	Paprastas sąlyginis sakiny.
Integruojami dalykai, pasiekimai	Anglų kalba, matematika.

Kompetencijos	<p>Pažinimo – įgis žinių apie šakotuosius algoritmus (paprastą sąlyginį sakinį ir jo užrašymą) ir sąlygos tikrinimo operatorių if.</p> <p>Skaitmeninė – savarankiškai ir mokytojo padedamas kuria skaitmeninį turinį programuodamas.</p> <p>Komunikavimo – pagalba draugams, atliktų darbų pristatymai, pasidalinimas idėjomis, diskusijos.</p> <p>Kūrybiškumo - mokiniai nuolat skatinami ieškoti kūrybiškų sprendimų, žinias pritaikyti kitose situacijose sprenddami uždavinius.</p>
Tikslas	Suprasti šakojimo veiksmus ir įtvirtinti įgūdžius sprendžiant uždavinius.
Uždaviniai	<ol style="list-style-type: none"> 1. Išsiaiškinsite, kokie algoritmai vadinami sąlyginiais. 2. Naudodamiesi pateiktu pavyzdžiu, susipažinsite sąlygos tikrinimo operatoriumi if. 3. Susipažinsite su sąlyginio sakinio užrašymui naudojamais palyginimo operatoriais. 4. Pritaikysite paprastą sąlyginį sakinį uždaviniams spręsti; 5. Apibendrinsite ir pristatysite gautus rezultatus.
Planuojamas rezultatas	<ol style="list-style-type: none"> 1. Gebėsite paaiškinti, kokiomis savybėmis pasižymi paprastas veiksmų šakojimas. 2. Gebėsite pritaikyti įgytas žinias sprenddami uždavinius; 3. Pagilinsite atlikto darbo pristatymo įgūdžius.
Specifinės priemonės programinė įranga	<ol style="list-style-type: none"> 1. Techninė įranga: išmanioji lenta, internetas. 2. Skaitmeninės priemonės: programavimo aplinkos Scratch: https://scratch.mit.edu/projects/293536838/ Python: https://www.onlinegdb.com/. Mentimeter. 3. Literatūra ir kiti ištekliai: <ol style="list-style-type: none"> 1. Python programavimo kalbos vadovėlis: https://www.emokykla.lt/titulinis-pythonprogramavimo-kalbos-vadovelis/47343
	<ol style="list-style-type: none"> 2. Programavimas per žaidimą: https://angis.lrt.lt/#?
Mokymosi metodai	<p>Aiškinimasis ir dialogas – analizuojant pavyzdžius.</p> <p>Darbas grupėse.</p> <p>Praktinis tyrimas – algoritmo užrašymas Python kalba.</p> <p>Apverstos klasės metodas</p>
Mokinių atlikto darbo vertinimas ir įsivertinimas (remtasi Informatikos programa [1])	<p>Slenkstinis – su mokytojo pagalba naudojasi programavimo kalbos paprasto sąlyginio sakinio konstrukcijomis, atkartoja mokytojo veiksmus programavimo kalboje. (B2.1)</p> <p>Patenkinamas – užrašo kelių eilučių algoritmus naudodamas programavimo kalbos paprasto sąlyginio sakinio konstrukcijas ir padedamas naudojasi programavimo aplinka. (B2.2)</p> <p>Pagrindinis – sprenddamas problemas naudoja programavimo kalbos paprasto sąlyginio sakinio konstrukcijas ir aplinką. (B2.3)</p> <p>Aukštesnysis – savarankiškai randa, išbando įvairias programavimo aplinkos funkcijas ir eksperimentuoja su jomis. (B2.4)</p>

Žinios prieš	Moka programuoti sąlyginius sakinius Scratch aplinkoje. Žino tiesinio algoritmo sąvoką. Moka naudoti Linoit lenta https://www.onlinegdb.com/ programavimo aplinka.
Galimybės taikyti specialiųjų poreikių mokiniams	Mokiniam galima pateikti iliustruotą kiekvienos sąlyginio sakinio dalies paaiškinimą ir sprendimo pavyzdžius.
Patarimai kolegoms, kurie naudos parengtą medžiagą	<ol style="list-style-type: none"> 1. Koordinuokite mokinių darbą grupėmis, individualų mokinių darbą, 2. Prieš pamoką atlikite visas pamokos plane numatytas užduotis patys. 3. Pamokos scenarijų pakeiskite, kad tiktų skirtingų gebėjimų mokiniams. 4. Galite papildyti mokiniams pateikiamus šaltinius įvairesniais. 5. Stenkitės, kad pamokos metu mokiniai dirbtų savarankiškai, o mokytojas būtų tik pagalbininkas ir konsultantas.

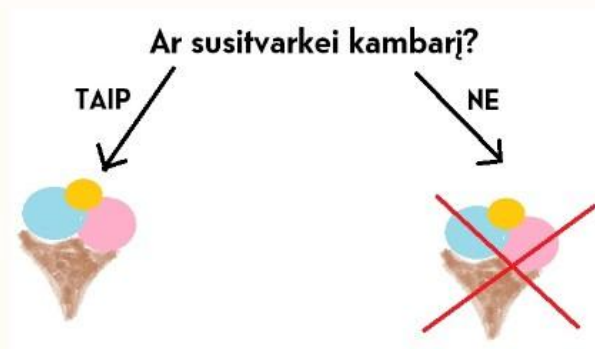
1 ETAPAS (5 minutės). Pamokos temos pristatymas.

Prisimenama „Scratch“ aplinkoje sąlyginio sakinio užrašymas. Mokiniai namuose turėjo prisiminti savo kurtus projektus su sąlygos sakiniais.

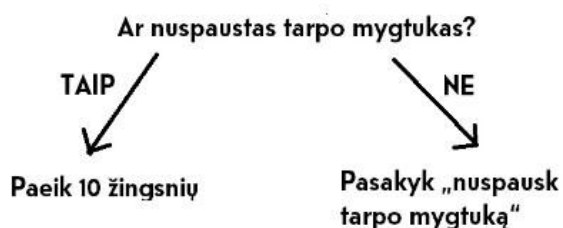
Mokytoja išmaniojoje lentoje pateikia sąlyginio sakinio užrašymą. Trumpai prisimenama, kas yra šakotieji algoritmai.

Kas yra sąlyga?

Sąlyga – tai klausimas, į kurį atsakius teigiamai, bus įvykdytas po jo einantis veiksmas. Visai kaip gyvenime, kai mama sako, kad galėsi suvalgyti ledus viena sąlyga – jei susitvarkysi kambarį.



Pavyzdys *Scratch*:



1 pav. Sąlyginio sakinio užrašymas „Scratch“ aplinkoje

Sąlygų komandos Scratch programoje

Kaip ir ciklo komandas, sąlygų komandas galite rasti toje pačioje *Valdymo (Control)* komandų grupėje. Yra dviejų tipų sąlygos:

- **Jeigu (*if then*)**. Patikrina sąlygą, ir atlieka nurodytą veiksmą tik tuomet, jei sąlyga įvykdyta.
- **Jeigu... kitais atvejais (*if then... else*)**. Jeigu sąlyga įvykdyta, atlieka vieną veiksmą, jeigu ne – kitą nurodytą veiksmą.

Toliau seka projekto pavyzdžio pristatymas.



Sprendimas – kurti dvi programas, veikiančias paraleliai, vienu metu:



2. pav. Projekto pavyzdžio pristatymas

2 ETAPAS (10 minučių). Naujos temos pristatymas, atsakymai į iškilusius klausimus.

Šakotųjų algoritmų (paprasčio sąlyginio sakinio ir jo užrašymo) ir sąlygos tikrinimo operatoriumi if pristatymas.

Sąlyginis sakiny *IF*. Kompiuteris ir jo programos būtų neprotingi ir mažai panaudojami dalykai, jei negalėtų priimti sprendimų. Programos parodė, kad kompiuteriai gali priimti sprendimus. Mes panagrinėsime kaip kompiuteris gali priimti pačius paprasčiausius sprendimus, nulemiančius programos vykdymo tėkmę. Tai yra - kokios operacijos bus vykdomos atsižvelgiant į situaciją.

Norėdami programos veikimą pakreipti tam tikru keliu, turime priimti sprendimą atsižvelgiant į įvairias, programai aktualias, sąlygas. Tai yra, turime palyginti tai, kas aktualu programai, pavyzdžiui:

Kai reikia spręsti uždavinius, kuriuose vieną arba daugiau veiksmų reikia atlikti ne pagal eilę, bet tik tuomet, kai yra įvykdomos tam tikros sąlygos. Pavyzdžiui, reikia nustatyti, ar pavyks laiku atvykti į susitikimą, žinant išvykimo laiką ir kelionės trukmę.

If sakinio struktūra. If sąlygos sakinyse gali susidėti iš kelių dalių:

if - būtinoji dalis, po jos rašomi veiksmai, kurie vydomi patenkinus sąlygą. elif / else if - jeigu reikia tikrinti daugiau sąlygų, galima rašyti tiek else if dalių kiek tik reikia.

else - rašosi pačiame gale, vieną kartą, jei reikia apimti visas sąlygas, kurios nepatenka į kitas dalis.

Paprasčiausia if sakinio forma yra tokia: if sąlyga: veiksmai...

Tai yra, rašomas raktažodis if, nurodoma sąlyga ir gale parašomas dvitaškis. Po to nurodomi veiksmai, kurie atitraukiami nuo krašto.

Sąlygos veiksmai bus vykdomi tik tuo atveju jei sąlygoje gausime teisingą (True tipo) atsakymą.

Užrašant sąlyginius sakinius naudojami palyginimo operatoriai:

<code>==</code>	lygu
<code>!=</code>	nelygu
<code>></code>	daugiau
<code>>=</code>	daugiau arba lygu
<code><</code>	mažiau
<code><=</code>	mažiau arba lygu

Kartu su mokiniais nagrinėjami pavyzdžiai apie paprasčiausią dviejų skaičių palyginimą.

```
m = 6 n = 4
```

```
if (m > n):
```

```
    print (m, ">", n) else:
```

```
    print (n, "> ", m)
```

Ekrane bus rodoma: **6 > 4**

Didesnis, arba lygus

```
m = 6 n = 6
```

```
if (m >= n):
```

```
print (m, ">=", n) else:  
print (m, "<", n)
```

Ekrane bus rodoma: **6 >= 6**

Kai mokiniai išnagrinėja pavyzdžius, dirbdami atskirose grupėse jie užrašo klausimą iš nagrinėtos temos kitai grupei. Tada visi atsakinėja į klausimus. Tam galima naudoti paprastą arba išmaniąją lentą, įvairias aplinkas.

3 ETAPAS (20 minučių). Savarankiškas darbas. Uždavinių sprendimas.

Mokiniai dirba grupėse.

1 užduo­tis. Tyrimas „Buo­žgalviai“. Vaikai nutarė stebėti tvenkinyje buo­žgalvius. Buo­žgalvis sveria a miligramų. Jei buo­žgalviai svers ne mažiau kaip gramą, bus rodoma: „Buo­žgalvių tyrimui pakanka“, jei mažiau – „Buo­žgalvių yra per mažai“.

Pasitikrinkite: jei a = 3, n = 1000000, bus rodoma: Buo­žgalvių tyrimui pakanka. Jei a = 3, n = 1000, bus rodoma: Buo­žgalvių yra per mažai.

Uždavinio sprendimas:

```
a = int(input("Kokia buo­žgalvių masė miligramais? ")) n =  
int(input("Kiek buo­žgalvių nusprendė stebėti vaikai? ")) if m *  
n >= 1000000:
```

```
print('Žuvų tyrimui pakanka') else:
```

```
print('Žuvų tyrimui per mažai')
```

2 užduo­tis. Gepardo greitis. Gepardas yra labai greitas žvėris. Jis per s sekundžių nulekia c metrų. Palyginkite gepardo greitumą su liūto, kuris lekia c m/s, greičiu. Jeigu liūtas greičiau nubėgs už gepardą, bus rodoma: „Liūtas bėga greičiau už gepardą“, jei greitesnis gepardas- „Gepardas bėga greičiau už liūtą“.

Patikrinkite: Kai $s = 2$, $c = 30$, $m = 6$, bus rodoma: “Liūtas bėga greičiau už gepardą”.

Uždavinio sprendimas:

```
# Gepardo bėgimo greitis metrais per sekundę
```

```
Liūto_greitis_cm_per_s = 30
```

```
# Laikas sekundėmis, per kurį tarakonas nubėga nustatytą atstumą
```

```
s = 2
```

```
# Atstumas, kurį nubėga gepardas per s sekundes c =
```

```
gepardo_greitis_cm_per_s * s
```

```
# Liūto skrydžio greitis metrais per sekundę liūto_greitis_m_per_s = 6
```

```
# Palyginame greičius
```

```
if liūto_greitis_m_per_s > (c / 100): # Konvertuojame
```

```
gepardo greitį iš centimetrų į metrus
```

```
print('Liūtas greitesnis už gepardą') else:
```

```
    print('Gepardas greitesnis už liūtą')
```

3 uždotis. Konsultacija. Konsultacija prasideda, kai laikrodis rodo a valandų ir b minučių.

Petras kavinėje valgyti pradėjo, kai laikrodžio rodyklės rodė aa valandų ir ab minučių. Petras pavalgo per bb minučių. Suskaičiuokite, ar Petras suspės užkristi ir laiku ateis į užsiėmimą.

Iš kavinės į klasę Petras nužingsniuoja per ž minučių.

Reikia patikrinti:

kai a = 14, b = 10, bb = 10, ba = 06, aa = 15, ž = 3, parodys „, Petras į konsultaciją nepavėluos“;

kai a = 14, b = 30, bb = 13, ba = 06, aa = 15, ž = 3, parodys „, Petras į konsultaciją pavėluos“;

kai a = 14, b = 10, bb = 14, ba = 06, aa = 10, ž = 5, parodys „, Petras į konsultaciją spės“.

Uždavinio sprendimas:

```
v = input("Kurią valandą prasideda konsultacija? ") m
= input("Kurią minutę prasideda konsultacija? ") vv =
input("kurią valandą Petras pradėjo pietauti? ") mv =
input("Kurią minutę Petras pradėjo pietauti? ") mm =
input("Kelias minutes Petras valgė? ") u = input("Per
kiek laiko atėjo į klasę? ")

if (a * 60 + b >= aa * 60 + ab + bb + ž):
print ("Petras į konsultaciją nevėluos")
else:      print ("Petras į konsultaciją
vėluos")
```

4. Kurmio svarstymai. Kurmis ilgai svarstė, ar jam apsimoka vesti Coliukę. Jis skaičiavo, kiek kainuos Coliukės išlaikymas. Coliukė suvalgo 0,5 grūdo per dieną. Kurmis planuoja gyventi santuokoje m metų, o Coliukės išlaikymui gali skirti n kilogramų grūdų. 1000 grūdų masė yra lygi g gramų. Laikykite, kad metus sudaro 365,25 dienos. Parašykite programą, kuri nustatytų, ar šykštuoliui Kurmiui apsimoka vesti Coliukę.

Pasitikrinkite: kai $g = 48.5$, $m = 10$, $n = 0.10$, tuomet ekrane turi būti rodoma: Kurmiui vesti Coliukę apsimoka.

Kai $g = 48.5$, $m = 10$, $n = 0.005$, tuomet ekrane turi būti rodoma: Kurmiui vesti Coliukės neapsimoka.

Sprendimas

```
g = float( input("Kiek sveria 1000 grūdų? "))

m = float(input("Kelis metus planuoja gyventi santuokoje?
```

```
) )  
  
    n = float(input("kiek kg grūdų skirs Coliukės išlaikymui?  
")) if ((g / 1000) * m * 0.5 * 365.25 <= n * 1000):  
  
    print("Kurmis Coliukę ves") else:  
  
    print("Kurmis Coliukės neves")
```

4 ETAPAS (10 minučių) Įsivertinimas. Refleksija.

Su mokiniais apibendrinamas atliktas darbas. Mokiniai parašo savo įspūdžius Mentimeter atsakydami į klausimus:

Parašykite 3 dalykus, kuriuos per pamoką sekėsi atlikti geriausiai.

Parašykite 3 dalykus, kuriuos per pamoką sekėsi atlikti sunkiausiai.

Parašykite 3 dalykus, kurie per pamoką buvo įdomiausi.

Kur pritaikysite praktikoje tai, ką išmokote?

Kelių sąlygų jungimas loginėmis operacijomis (Daiva Vaičiulionienė)

Pasiekimų sritis (remtasi Informatikos programa [1])	(B) Algoritmai ir programavimas 28.2.4. Algoritmų parinkimas. Aptariami klasikinių algoritmų pavyzdžiai (dalybos, paieškos, rikiavimo). Diskutuojama apie žinomų algoritmų naudojimą ir pritaikymą konkrečioms uždaviniais spręsti. Susipažinama su algoritmų efektyvumo, optimizavimo problemomis.
Klasė	7-8 kl.
Tema	Kelių sąlygų jungimas loginėmis operacijomis.
Integruojami dalykai, pasiekimai	Anglų kalba, matematika.
Kompetencijos	Pažinimo – įgis žinių apie šakotuosius algoritmus (sąlyginį sakinį su keliomis sąlygomis ir jo užrašymą) ir sąlygos tikrinimo operatorių if. Skaitmeninė – savarankiškai ir mokytojo padedamas kuria skaitmeninį turinį programuodamas. Komunikavimo – pagalba draugams, atliktų darbų pristatymai, pasidalinimas idėjomis, diskusijos. Kūrybiškumo - mokiniai nuolat skatinami ieškoti kūrybiškų sprendimų, žinias pritaikyti kitose situacijose spęsdami uždavinius.
Tikslas	Suprasti šakotuosius kelių sąlygų algoritmus ir įtvirtinti įgūdžius sprendžiant uždavinius.
Uždaviniai	1. Mokysitės kelias sąlygas sujungti loginėmis operacijomis. 2. Pritaikysite žinias uždaviniams spręsti. 3. Apibendrinsite ir pristatysite gautus rezultatus.
Planuojamas rezultatas	1. Gebėsite kelias slygas sujungti loginėmis operacijomis. 2. Gebėsite pritaikyti įgytas žinias spęsdami uždavinius; 3. Pagilinsite atlikto darbo pristatymo įgūdžius.
Specifinės priemonės programinė įranga	1. Techninė įranga: išmanioji lenta, internetas. 4. Skaitmeninės priemonės: programavimo aplinka Python: https://www.onlinegdb.com/ , https://scratch.mit.edu/projects/293536838/ / https://mentimeter.com/ Linoit virtuali lenta: https://en.linoit.com/

	<p><u>2. Literatūra ir kiti ištekliai:</u></p> <p>Python programavimo kalbos vadovėlis: https://www.emokykla.lt/titulinis-pythonprogramavimo-kalbos-vadovelis/47343</p>
Mokymosi metodai	<p>Praktinis tyrimas – algoritmo užrašymas Python kalba.</p> <p>Atradimo metodas - tinkamo palyginimo operatoriaus įrašymas.</p>
Mokinių atlikto darbo vertinimas ir išivertinimas (remtasi Informatikos programa [1])	<p>Slenkstinis – su mokytojo pagalba naudojasi programavimo kalbos šakojimo konstrukcijomis, atkartoja mokytojo veiksmus programavimo kalboje. (B2.1)</p> <p>Patenkinamas – užrašo kelių eilučių algoritmus naudodamas programavimo kalbos šakojimo konstrukcijas ir padedamas naudojasi programavimo aplinka. (B2.2)</p> <p>Pagrindinis – sprendamas problemas naudoja programavimo kalbos šakojimo konstrukcijas ir aplinką. (B2.3)</p> <p>Aukštesnysis – savarankiškai randa, išbando įvairias programavimo aplinkos funkcijas ir eksperimentuoja su jomis. (B2.4)</p>
Žinios prieš	<p>Žino paprastąjį sąlyginį sakinį. Moka naudotis https://www.onlinegdb.com/ programavimo aplinka.</p>
Galimybės taikyti spec. poreikių mokiniams	<p>Pateikiamas iliustruotas šakojimo veiksmų su keliomis sąlygomis ir tai iliustruojantys pavyzdžiai.</p>
Rekomendacijos	<ol style="list-style-type: none"> 1. Koordinuokite mokinių darbą grupėmis, individualų mokinių darbą, 2. Prieš pamoką peržiūrėkite visas pamokos plane numatytas užduotis patys. 3. Pamokos planą pakeiskite, kad tiktų skirtingų gebėjimų mokiniams. 4. Galite papildyti mokiniams pateikiamus šaltinius įvairesniais. 5. Stenkitės, kad pamokos metu mokiniai dirbtų savarankiškai, o mokytojas būtų tik pagalbininkas ir konsultantas.

1 ETAPAS (5 minutės). Praeitą pamokos medžiagą kartojimas.

Mokytojas pateikia uždavinį, kuriame mokiniai turi įrašyti tinkamą palyginimo operatorių:

santaupos = 10

išlaidos = 5

```

#Patikrinti, ar santaupos didesnės už išlaidas
if (santaupos >
išlaidos):
    print ("Santaupos didesnės už išlaidas") else:
    print ("Santaupos mažesnės už išlaidas")

```

1. Vietoj klaustuko įrašykite tinkamą palyginimo operatorių:

```

santaupos = 10
išlaidos = 5

#Patikrinti, ar santaupos didesnės už išlaidas
if (santaupos ?
išlaidos):
    print ("Santaupos didesnės už išlaidas")
else:
    print ("Santaupos mažesnės už
išlaidas")
*****

santaupos = 10
išlaidos = 5

#Patikrinti, ar santaupos didesnės už išlaidas
if (santaupos ? išlaidos):
    print
("Santaupos mažesnės už išlaidas") else:
    print ("Santaupos didesnės už išlaidas")

```

2 ETAPAS (15 minučių). Naujos temos pristatymas, atsakymai į iškilusius klausimus.

Supažindinama su loginėmis operacijomis, jų reikšmėmis ir pavyzdžiais.

Mokiniamis pristatomos loginės operacijos ir jų skirtumai nuo aritmetinių operacijų. Yra 3 loginiai operatoriai: **IR**, **ARBA**, **NE**. Šių operatorių dėka galima formuluoti sudėtingesnes sąlygas. Loginiai

operatoriai naudojami kasdien ir realiame gyvenime, tačiau jie gali pasirodyti tokie natūralūs, jog apie tai net nesusimąstome. Pavyzdžiui:

Norėčiau šalto gėrimo ARBA ledų

Jei atliksi namų darbus IR susitvarkysi kambarį, galėsi [...] NEtriukšmauk!

Loginiai operatoriai dažniausiai naudojami norint patikrinti kelias sąlygas vienu metu. Tuo būdu siekiama sumažinti kodo kiekį ir jį supaprastinti.

Prisimename ir palyginame su Scratch:

Komandų derinių pavyzdžiai su paaiškinimais:



Veikėjas eis tik tuomet, kai bus nuspaustas tarpo mygtukas klaviatūroje IR lies žalią spalvą.



Veikėjas eis, jei jis žiūri į dešinę ARBA jei yra dešinėje ekrano pusėje.

3 pav. Komandų derinių pavyzdžiai Scratch aplinkoje

Loginiai operatoriai ir, ar, ne (medžiaga mokytojui)

Operatorius	Aprašymas	Pavyzdys
and	True , jeigu abi sąlygos yra True	x < 5 and x < 10
or	True , jeigu bent viena iš sąlygų yra True	x < 5 or x < 4
not	atvirkštinė reikšmė	not 1 < 2

and (ir) operatorius

and Pitono loginis operatorius atlieka dviejų loginių reikšmių, kintamųjų arba išraiškų loginį palyginimą.

Jei abiejų operatoriaus pusių reikšmės yra True , operatorius and grąžina True . Jei kurios nors iš pusių (arba abiejų pusių) reikšmė False , operatorius and grąžina False .

Jei naudojama ne loginio tipo reikšmė (arba kintamasis, kuriame saugoma reikšmė), Pitone ji visada bus vertinama kaip True .

Teisingumo lentelė

X	Y	X and Y
0	0	0
0	1	0
1	0	0
1	1	1

4 pav. Operatorius IR (AND)

Operacija ir rezultatas

Operacija	Rezultatas
False and False	False
False and True	False
True and False	False
True and True	True

Keli pavyzdžiai

```
In [ ]: True and True # Bus True
True and False # Bus False
False and False # Bus False
1 == 1 and 1 < 2 # Bus True
1 < 2 and 3 < 1 # Bus False
"Taip" and 100 # Bus True
```

5 pav. Operatoriaus IR (AND) pavyzdžiai

or (arba) operatorius

or Pitono loginis operatorius sudeda dvi logines išraiškas ir įvertina True, jei bent viena iš išraiškų įvertinama True. Priešingu atveju, jei abi išraiškos yra False, visa išraiška įvertinama kaip False.

Teisingumo lentelė

X	Y	X or Y
0	0	0
0	1	1
1	0	1
1	1	1

Operacija ir rezultatas

Operacija	Rezultatas
False or False	False
False or True	True
True or False	True
True or True	True

6 pav. Operatorius ARBA (OR)

Keli pavyzdžiai

```
In [ ]: True or True # Bus True
True or False # Bus True
False or False # Bus False
1 < 2 or 3 < 1 # Bus True
3 < 1 or 1 > 6 # Bus False
1 == 1 or 1 < 2 # Bus True
```

7 pav. Operatoriaus ARBA (OR) pavyzdžiai

not (ne) operatorius

ne Pitono loginis operatorius naudojamas loginėje išraiškoje tam, kad būtų gauta išraiškos atvirkštinė reikšmė. Jei pradinė išraiška buvo True, panaudojus operatorių not išraiška taptų False, ir atvirkščiai.

Teisingumo lentelė

X	not X
0	1
1	0

Operacija ir rezultatas

Operacija	Rezultatas
not False	True
not True	False

Keli pavyzdžiai

```
In [ ]: not True # Bus False
not False # Bus True
1 > 2 # Bus False
not 1 > 2 # Bus True
1 == 1 # Bus True
not 1 == 1 # Bus False
```

8 pav. Operatorius NE (NOT) ir pavyzdžiai

Kartu su mokiniais nagrinėjami pavyzdžiai su loginėmis operacijomis:

1.

```
skaicius = int( input('Įveskite skaičių: ') )
if skaicius > 0 and skaicius <= 100:
    print('Skaičius patenka į intervalą [1-100].')
else:
    print('Skaičius nepatenka į intervalą [1-100].')
```

2.

```
skaicius = int( input('Įveskite skaičių: ') )
if skaicius == 3 or skaicius > 100:
    print('Skaičius lygus 3 arba didesnis nei 100.')
```

3.

```
skaicius = int( input('Įveskite skaičių: ') )  
  
if not skaicius % 2 == 0:  
    print('Skaičius nelyginis.') else:  
    print('Skaičius lyginis.')
```

3 ETAPAS (20 minučių). Savarankiškas darbas-praktinis tyrimas.

1 užduotis. Sarafanas. Rinkdamasi audinį mokykliniam sarafanui Roma mano, kad jai labiausiai tiks gelsvo, rudo ar žalio atspalvio audinys. Internete ji ieško audinių pavyzdžių ($h_1 = 1$, tai gelsvo atspalvio audinio yra, $h_1 = 0$ – gelsvo atspalvio audinio internete nėra; $h_2 = 1$, tai rudo atspalvio audinio yra, $h_2 = 0$ – rudo atspalvio audinio internete nėra; $h_3 = 1$, tai žalio atspalvio audinio yra, $h_3 = 0$ – žalio atspalvio audinio internete nėra). Parašykite, ar pavyks Romai nusipirkti audinio mokykliniam sarafanui.

Išspręskite:

kada $h_1 = 1$, $h_2 = 1$, $h_3 = 1$, tada bus rašoma: Roma audinio mokykliniam sarafanui įsigis.

Kada $h_1 = 1$, $h_2 = 0$, $h_3 = 0$, tada bus rašoma: Roma medžiagą mokykliniam sarafanui įsigis.

Kada $h_1 = 0$, $h_2 = 0$, $h_3 = 0$, tada bus rašoma: Roma audinio mokykliniam sarafanui neįsigis. Roma medžiagą mokykliniam sarafanui įsigis, jeigu internete bus bent vieno iš norimo atspalvių medžiagos audinio. Šiuo atveju sąlygos sakiniai sujungiami logine operacija **arba**.

Sprendimas:

```
h1 = int(input("Ar yra gelsvos spalvos audinio (įveskite 1 jei  
taip, 0 jei ne): "))  
  
h2 = int(input("Ar yra rudos spalvos audinio (įveskite 1 jei taip,  
0 jei ne): "))  
h3 = int(input("Ar yra žalios spalvos audinio (įveskite 1 jei taip, 0  
jei ne): ")) if h1 == 1 or h2 == 1 or h3 == 1:  
  
    print("Roma medžiagos mokykliniam sarafanui nusipirks.") elif h1  
== 1:  
  
    print("Roma medžiagą mokykliniam sarafanui įsigis.") else:
```

```
print("Roma audinio mokykliniam sarafanui neįsigis.")
```

2 užduotis. Paskatinimas. Norėdamas paskatinti penktokus kantriai dirbti, mokytojas nutarė trimestro pabaigoje pridėti po 10 tiems:

- kurie visada atliko užduotis namuose ($r_1 = 1$, kai penktokas visada atliko užduotis namuose, $r_1 = 0$, kai penktokas neatliko užduočių namuose),
- kurių sąsiuviniai yra sutvarkyti ($r_2 = 1$, kai penktokas dirbo tvarkingai, $r_2 = 0$, kai penktoko užrašai yra nesutvarkyti),
- kurie visada gerai lankė ($r_3 = 1$, kai mokinys gerai lankė, $r_3 = 0$, kai mokinys yra praleidęs ir nepateisinęs pamokų).

Suskaičiuokite, ar penktokas gaus 10 už trimestro darbą.

Išspręskite:

kai $r_1 = 0$, $r_2 = 0$, $r_3 = 0$, tada bus rodoma: Penktokas 10 negaus, kai

$r_1 = 0$, $r_2 = 1$, $r_3 = 0$, tada bus rodoma: Penktokas 10 negaus, kai

$r_1 = 1$, $r_2 = 1$, $r_3 = 1$, tada bus rodoma: - Penktokas 10 gaus.

Tik įvykdžius visus reikalavimus, bus galima gauti 10.

Šiame uždavinyje sąlygos sakiniai sujungti logine operacija **ir**.

Sprendimas:

```
r1 = int(input("Ar penktokas visada sprendė namų darbus? (1 - taip, 0 - ne): "))
```

```
r2 = int(input("Ar penktoko sąsiuviniai tvarkingi ir pilni? (1 - taip, 0 - ne): "))
```

```
r3 = int(input("Ar penktokas nepraleido nė vienos pamokos be pateisinamos priežasties? (1 - taip, 0 - ne): "))
```

```
if r1 == 1 and r2 == 1 and r3 == 1:
```

```
    print("Sveikiname! Jūs gausite dešimtuką už gerą pusmečio darbą.") else:
```

```
print("Dešimtuko šį kartą negausite. Tęskite stengtis!")
```

3 uždutis. Siurprizas.

Kastytis eis į Jono vakarėlį. Jonas patinka konstruoti, todėl Kastytis nupirko konstruktorių. Jo skersmuo f centimetrų ir nutarė jį supakuoti į margą stačiakampio gretasienio formos dėželę. Jos aukštis yra e , ilgis – $è$, o plotis – b cm.

Išspręskite: „Konstruktorius į dėželę sutilps“, jei konstruktorius į dėželę sutilps, arba „Konstruktorius“ į dėželę nesutilps“, jei konstruktorius į dėželę nesutilps. Duomenys sveikieji skaičiai. Išspręskite: kada $e = 20$, $è = 35$, $b = 120$, $f = 80$, bus rodoma: Kamuolys į dėželę nesutilps.

Kada $e = 110$, $b = 85$, $c = 90$, $f = 70$, bus rodoma: Kamuolys į dėželę sutilps.

Uždavinio sprendimas:

```
ilgis = int(input("Įveskite dėželės ilgį: ")) plotis =
int(input("Įveskite dėželės plotį: ")) aukštis =
int(input("Įveskite dėželės aukštį: ")) skersmuo =
int(input("Įveskite kamuolio skersmenį: ")) if ilgis >=
skersmuo and plotis >= skersmuo and aukštis >=
skersmuo:          print ("Kamuolys į dėžute
tilps") else:
    print ("Kamuolys į dėželę netilps")
```

4 uždutis. Trupininis pyragas

Norint iškepti skanų „trupininį“ reikia vaisių ir tešlos. Ona ir Ūla labiausiai mėgsta „trupininį“ su kriaušėmis arba gervuogėmis, o Kazys ir Vytas – su obuoliais arba abrikosais. Išspręskite, kad įvedant sveikąjį skaičių t , kuris žymi uogų rūšį (1 – kriaušės, 2 – gervuogės, 3 – obuoliai, 4 – abrikosai), būtų rodoma, kam trupininis patiktų labiausiai.

Pasitikrinkite:

- Jeigu $t = 1$, rodoma: Trupininis patiktų Onai ir Ūlai.
- Jeigu $t = 2$, rodoma: Trupininis patiktų Onai ir Ūlai.
- Jeigu $t = 3$, rodoma: Trupininis patiktų Kaziui ir Vytui.
- Jeigu $t = 4$, rodoma: Trupininis patiktų Kaziui ir Vytui.

Uždavinio sprendimas:

```
# Uogų rūšis (1 - kriaušės, 2 - gervuogės, 3 - obuoliai, 4 - slyvos) t
= int(input("Įveskite vaisių rūšį (1-4): "))

if t == 1 or t == 2:

    print("Trupininis labiausiai patiks Onai ir Ūlai.") elif
t == 3 or t == 4:

    print("Trupininis labiausiai patiks Kaziui ir Vytui.")
else:

    print("Neteisinga uogų rūšis įvesta.")
```

5 uždotis. Spintos tvarkymas.

Dvi sesės nesusitaria, kuriai reikia sutvarkyti spintą. Jos meta žaidimo kubelį. Jei iškris skaičiai 3, 5 arba 1 – spintą sutvarko jaunėlė, jeigu 4, 6, 2 – vyresnėlė.

Išspręskite, kad įvedant skaičių d , būtų rodoma: „Spintą sutvarko jaunėlė“, arba „Spintą sutvarko vyresnėlė“. Pasitikrinkite:

kada $d = 1$, tada būtų rodoma: Spintą sutvarko jaunėlė.

Kada $d = 4$, tada būtų rodoma: Spintą sutvarko vyresnėlė .

Uždavinio sprendimas:

```
#skaičiai (1,2,3,4,5,6)
d=int(input("Įveskite skaičių?")) if
d==1 or d==3 or d==5:
    print("Spintą sutvarko jaunėlė.") elif
d==2 or d==4 or d==6:
    print("Spintą sutvarko vyresnėlė ") else:
    print("Netinkamas skaičius, įveskite 1, 2, 3, 4, 5 arba 6")
```

4 ETAPAS (5 minutės). Pamokos aptarimas. Įsivertinimas.

Mokiniai ant skirtingų spalvų lapelių parašo, kas jiems pasisekė geriausiai ir kas nepavyko, buvo sunku. Tam galima naudoti paprastą arba išmaniają lentą, įvairias virtualias aplinkas.

Sudėtingas sąlyginis sakiny (Daiva Vaičiulionienė)

5 lentelė. Trečios pamokos planas

Pasiekimų sritis (remtasi Informatikos programa [1])	(B) Algoritmai ir programavimas 28.2.4. Algoritmų parinkimas. Aptariami klasikinių algoritmų pavyzdžiai (dalybos, paieškos, rikiavimo). Diskutuojama apie žinomų algoritmų naudojimą ir pritaikymą konkreitiems uždaviniams spręsti. Susipažįstama su algoritmų efektyvumo, optimizavimo problemomis.
Klasė	7-8 kl.
Tema	Sudėtingas sąlyginis sakiny.
Integruojami dalykai, pasiekimai	Anglų kalba, matematika.
Kompetencijos	Pažinimo – įgis žinių apie šakotuosius algoritmus (sąlyginį sakiny su keliomis sąlygomis ir jo užrašymą) ir sąlygos tikrinimo operatorių if. Skaitmeninė – savarankiškai ir mokytojo padedamas kuria skaitmeninį turinį programuodamas. Komunikavimo – pagalba draugams, atliktų darbų pristatymai, pasidalinimas idėjomis, diskusijos. Kūrybiškumo - mokiniai nuolat skatinami ieškoti kūrybiškų sprendimų, žinias pritaikyti kitose situacijose spręsdami uždavinius.
Tikslas	Supažindinti su sudėtingu sąlyginiu sakiniu.
Uždaviniai	1. Mokysitės tinkamai užrašyti sudėtingą sąlyginį sakiny. 2. Įgytas žinias pritaikysite spręsdami uždavinius. 3. Apibendrinsite ir pristatysite gautus rezultatus.
Planuojamas rezultatas	1. Gebėsite tinkamai užrašyti sudėtingą sąlyginį sakiny. 2. Gebėsite pritaikyti įgytas žinias spręsdami uždavinius; 3. Pagilinsite atlikto darbo pristatymo įgūdžius.
Specifinės priemonės programinė įranga /	1. Techninė įranga: išmanioji lenta. 2. Skaitmeninės priemonės: Python programavimo aplinka: Linoit virtuali lenta: https://www.onlinegdb.com/ , Mentimeter, https://en.linoit.com/ 3. <u>Literatūra ir kiti ištekliai:</u> https://www.emokykla.lt/titulinis-python- Python programavimo kalbos vadovėlis: programavimo-kalbos-vadovelis/47343

Mokymosi metodai	„Spaudos konferencija“– klausimai mokytojui arba mokiniui iš praeitos temos pamokos pradžioje. Grupės darbas- varžytuvės.
	Praktinis tyrimas– algoritmo užrašymas Python kalba.
Mokinių atlikto darbo vertinimas ir įsivertinimas (remtasi Informatikos programa [1])	Slenkstinis – su mokytojo pagalba naudojasi programavimo kalbos šakojimo konstrukcijomis, atkartoja mokytojo veiksmus programavimo kalboje. (B2.1) Patenkinamas – užrašo kelių eilučių algoritmus naudodamas programavimo kalbos šakojimo konstrukcijas ir padedamas naudojasi programavimo aplinka. (B2.2) Pagrindinis – sprendamas problemas naudoja programavimo kalbos šakojimo konstrukcijas ir aplinką. (B2.3) Aukštesnysis – savarankiškai randa, išbando įvairias programavimo aplinkos funkcijas ir eksperimentuoja su jomis. (B2.4)
Žinios prieš	Žino, ką vadiname paprastu sąlyginiu, geba sujungti kelias sąlygas loginėmis operacijomis. Moka naudotis https://www.onlinegdb.com/ programavimo aplinka.
Galimybės taikyti specialiųjų poreikių mokiniams	Pateikiamas kiekvienos komandos funkcijų (ką jos atlieka) iliustruotas paaiškinimas ir paruošiamas pavyzdys, kaip komandoms priskirti įvykius.
Rekomendacijos	<ol style="list-style-type: none"> 1. Koordinuokite mokinių darbą grupėmis, individualų mokinių darbą, 2. Prieš pamoką peržiūrėkite visas pamokos plane numatytas užduotis patys. 3. Pamokos planą pakeiskite, kad tiktų skirtingų gebėjimų mokiniams. 4. Galite papildyti mokiniams pateikiamus šaltinius įvairesniais. 5. Stenkitės, kad pamokos metu mokiniai dirbtų savarankiškai, o mokytojas būtų tik pagalbininkas ir konsultantas.

1 ETAPAS (15 minučių). Sudėtingo sąlyginio sakinio pristatymas.

Naudojant tik **if** dalį tikriausiai pastebėjote, kad tikriname tik vieną sąlygą. **elif / else if** leidžia tikrinti kelias sąlygas (vieną po kitos). Pavyzdžiui:

Parinkti tokį pavyzdį, kad būtų daugiau negu 2 sąlygos. Pvz.: jei iki pamokos galo liko mažiau nei 8 minutės, tuomet pamoka baigiasi, jei 7-15 – liko nedaug laiko, jei 16-30 – liko nemažai laiko, jei daugiau negu 30 minučių – liko labai daug laiko.

```
t = input("Kiek laiko liko iki pertraukos pabaigos? ") t =
```

```
int (m) if (m < 8) :
```

```

    print ("Pertrauka greitai baigsis")
elif (m < 16):

    print("Liko mažai laiko") elif (m
<= 30):

    print("Liko daug laiko") else:

    print("Liko dar labai daug laiko")

```

Galima naudoti tiek **elif** blokų kiek tik reikia.

Kiekviena **elif** dalis atsakinga už naujos sąlygos apdorojimą.

Jei įvertinant skirtingas sąlygas yra randama teisinga, tuomet bloke esantis kodas yra įvykdomas ir tolesnės sąlygos nebetikrinamos. T.y. yra vykdoma tik pirma rasta teisinga sąlyga. **elif /else if sintaksė**

```

if salyga_1:

    # šio bloko kodas vykdomas jeigu salyga_1 reikšmė yra True

elif salyga_2:

    # šio bloko kodas vykdomas jeigu salyga_2 reikšmė yra True

elif salyga_3:

    # šio bloko kodas vykdomas jeigu salyga_3 reikšmė yra
True

```

Kartu su mokiniais sprendžiamas pavyzdys „Žaidimo meniu pasirinkimas“:

```

print("1.  žaisti")    print("2.
Lyderiai")  print("3. Nustatymai")
print("4. Išėiti")

pasirinkimas = int( input('Iveskite meniu (1-4): ') )

if pasirinkimas == 1:

```

```

    print('Pasiruoškite stipriai žaisti!') elif
pasirinkimas == 2:
    print('Toliau pateiksime lyderių lentelę...') elif
pasirinkimas == 3:
    print('Nustatykite grafikos ir garso nustatymus...') elif
pasirinkimas == 4:
    print('Programa išjungiama...')

```

1. Žaisti
2. Lyderiai
3. Nustatymai
4. Išeiti

Iveskite meniu (1-4): 1

Pasiruoškite stipriai žaisti!

2 ETAPAS (20 minutės). Savarankiškas darbas. Darbas grupėse – varžybos „Kas greičiau išspręs“.

1 užduois. Skaičių palyginimas. Parašykite programą, kuri palygintų du skaičius (skaičiai gali ir sutapti).

Sprendimas:

```

skaicius_1 = int(input("Įveskite 1 skaičių: ")) skaicius_2 =
int(input("Įveskite 2 skaičių: ")) if skaicius_1 > skaicius_2:
print("Skaičius 1 yra didesnis už skaičių 2.") elif skaicius_1 <
skaicius_2:
    print("Skaičius 1 yra mažesnis už skaičių 2.") else:
    print("Skaičius 1 lygus skaičiui 2.")

```

2 uždutis. Intervalo suradimas. Vaikai žaidžia žaidimą ir skaičiuoja taškus. Jeigu surinkote ne mažiau kaip 50 taškų, tai esate genijus. Jeigu surinkote 49-30 taškų, tai esate puikus žaidėjas. Jeigu 29-10 taškų, esate geras žaidėjas. Jeigu surinkote 9-5 taškus, esate vidutinis žaidėjas.

Jeigu mažiau nei 5 taškus, jūs esate pradedantysis žaidėjas. Parašykite programą, kuri surastų koks žaidėjas esate.

Sprendimas:

```
taškai = int (input ("Kiek taškų surinkote?"))
if taškai >= 50:
    print('Jūs esate genijus!') elif taškai >=
30:
    print('Jūs esate puikus žaidėjas') elif taškai >=
10:
    print('Jūs esate geras žaidėjas') elif taškai >=
5:
    print('Jūs esate vidutinis žaidėjas') else:
    print('Jūs esate pradedantysis žaidėjas')
```

3 uždutis. Testo rezultatai. Mokinys turi įvesti skaitinį testo rezultatą (t.y. pažymį nuo 1 iki 10) ir gauti žodinį įvertinimą.

Parašykite programą ir išveskite atitinkamą tekstą:

- Jei pažymys yra lygus 10, išvesti "puiku".
- Jei pažymys yra lygus 9, išvesti "labai gerai".
- Jei pažymys yra 8 arba 7, išvesti "gerai".
- Jei pažymys yra lygus 6 arba 5, išvesti "patenkinamai".
- Jei pažymys mažesnis nei 5, išvesti "Testas neišlaikytas".

Sprendimas:

```
pažymys=int(input('įveskite pažymį'))
```

```

if pažymys == 10:
print('Puiku!') elif
pažymys >= 9:
    print('Labai gerai') elif
pažymys >= 7:    print('Gerai')
elif pažymys >= 5:
    print('Patenkinamai') else:
    print ('Testas neišlaikytas')

```

4 uždutis. Troleibusas. Jeigu norite nužingsniuoti nuo troleibuso stotelės iki laivų priplaukos maždaug reikia r_1 minučių. Nulėkti iki laivų priplaukos ir spėti į troleibusą galima per r_2 minučių.

Išspręskite, ar iš troleibuso išėjęs keliautojas suspės nužingsniuoti arba nulėkti į laivą, išplauksiantį po 1 min.

Pasitikrinkite: kai $l_1 = 2, l_2 = 1, l = 5$, tada keliautojas suspės nužingsniuoti į laivą; kai $l_1 = 5, l_2 = 2, l = 3$, tada keliautojas spės nulėkti į laivą; kai $l_1 = 3, l_2 = 2, l = 1$, tada keliautojas nespės į laivą.

Uždavinio sprendimas:

```

laikas_nueiti = int(input("Kiek laiko reikia nueiti iki
stoties?"))

laikas_nubėgti = int(input("Kiek laiko reikia nubėgti iki
stoties?"))

laikas = int(input("Po kelių minučių atvažiuoja troleibusas?"))

if(laikas_nueiti <= laikas):    print
("Spės nueiti") elif(laikas_nubėgti <=
laikas):

```

```
print("Spės nubėgti") else:  
print ("Nespės")
```

5 uždutis. Dovanėlė. Moksleivis per pusmetį gavo 6 matematikos modulio įvertinimus.

Vadovė nutarė padaryti moksleiviams dovanėlę: moksleiviai, kurių įvertinimų vidurkis yra daugiau už 9, gaus 3 taškus, o moksleiviams, su vidurkiu tarp 7 ir 9, įskaičiuojant intervalo pabaigas, bus įrašyti 2 taškai. Kitiems bus įrašyta po 1 tašką. Suskaičiuokite, kiek taškų jiems bus įrašyta.

Uždavinio sprendimas:

```
p1 = int(input("Koks pirmas pažymys? ")) p2 =  
int(input("Koks antras pažymys? ")) p3 =  
int(input("Koks trečias pažymys? ")) p4 =  
int(input("Koks ketvirtas pažymys? ")) p5 =  
int(input("Koks penktas pažymys? ")) vidurkis = (p1  
+ p2 + p3 + p4 + p5) / 5  
if (vidurkis > 9):  
    print ("Gaus 3 taškus") elif  
(vidurkis >= 7):  
    print ("Gaus 2 taškus") else:  
    print("Gaus 1 tašką")
```

3 ETAPAS (10 minutės). Pamokos rezultatų aptarimas. Įsivertinimas. Refleksija.

1. **Varžybų rezultatų skaičiavimas.** Mokiniai išmaniojoje lentoje užrašo savo vardą į tą lentelės dalį, pagal tai, kiek uždavinių išsprendė.

6 lentelė. Rezultatų skaičiavimas

Grupės /mokiniai	4 uždaviniai	3 uždaviniai	2 uždaviniai	1 uždavinys

2. **Refleksija.** Pabaikite sakinius:

- Per pamoką sekėsi atlikti geriausiai....
- Per pamoką sekėsi atlikti sunkiausiai....
- Per pamoką buvo įdomiausia....
- Praktiškai tai, ką išmokau, galėčiau pritaikyti

Komentarų svarba programuojant (Dovilė Žurauskienė)

Pamokų planas pateiktas lentelėje

4 lentelė. Pamokos planas

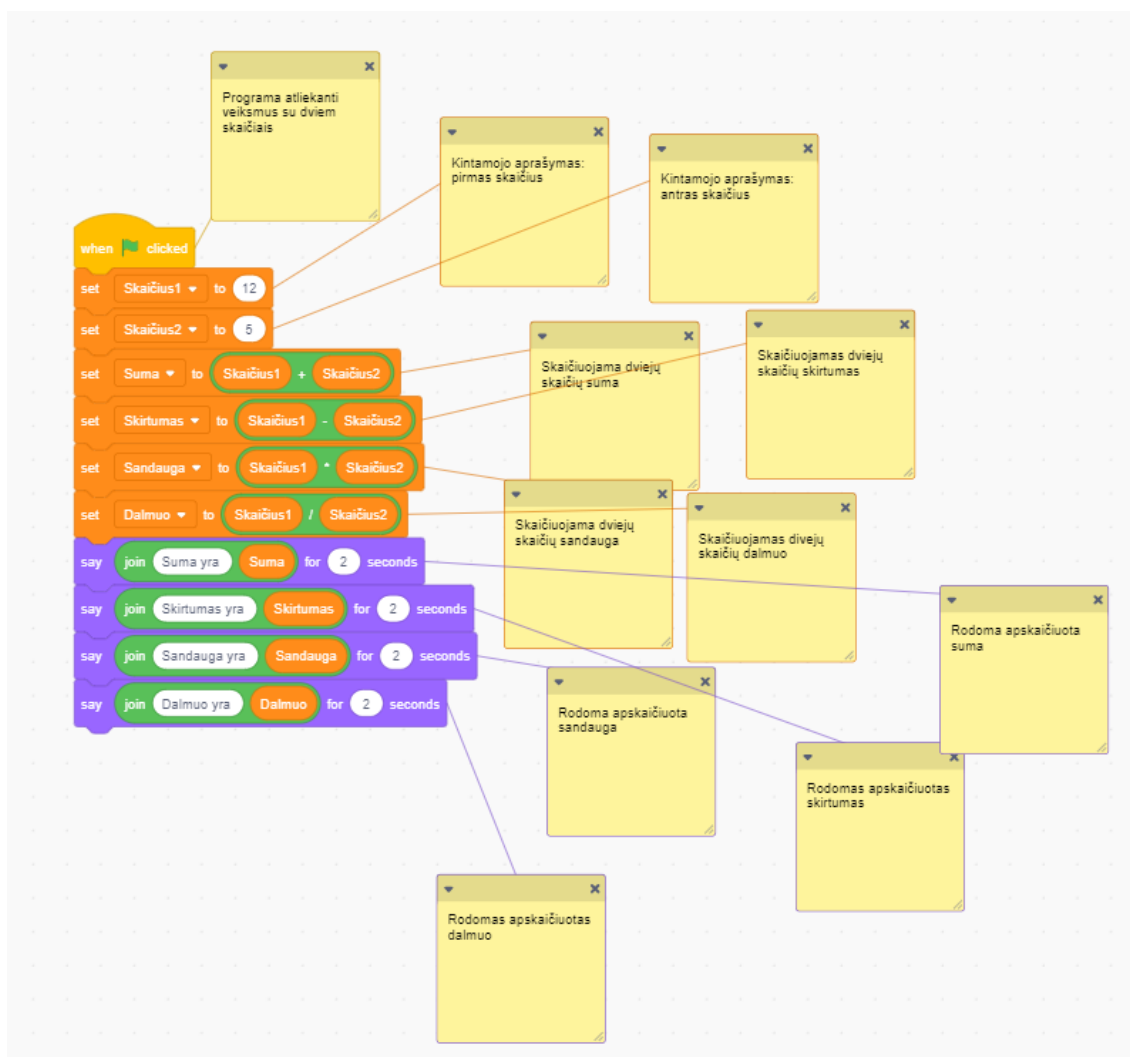
Pasiekimų sritis	(B) Algoritmai ir programavimo mokymasis. 28.2.5. Programos derinimas
Klasė	7-8 kl.
Tema	Komentarų svarba programuojant.

Integruojami dalykai, pasiekimai	Technologijos, užsienio kalba (anglų)
Kompetencijos	Pažinimo – supranta komentarų paskirtį ir svarbą, ugdomi kritinio mąstymo įgūdžius. Skaitmeninė – kuria skaitmeninį turinį programuodami.
Tikslas	Susipažinti su komentarų svarba programuojant
Uždaviniai	1. Išsiaiškinti kuriam tikslui naudojami komentarai programuojant. 2. Mokysitės tinkamai komentuoti kuriamas programas.
Planuojamas rezultatas	1. Gebėsite paaiškinti kuo svarbūs komentarai programavime. 2. Gebėsite tinkamai komentuoti kuriamas programas.
Specifinės priemonės / programinė įranga	Multimedia, projektorius, išmanioji lenta. Scratch aplinka: https://scratch.mit.edu/ Python aplinka: https://www.onlinegdb.com/ Literatūra ir kiti šaltiniai: ✓ https://wiki.angis.net/w/Python_Vadovėlis ✓ https://github.com/blaztoma/pitono-pamokos
Mokymosi metodai	Aiškinimas, praktinis tyrimas, darbas grupėmis, pateikčių ir/ar skaitmeninių mokymosi objektų naudojimas.
Mokinių atlikto darbo vertinimas ir įsivertinimas	Slenkstinis pasiekimų lygis – su pagalba rašo komentarus programoje. Patenkinamas pasiekimų lygis – savarankiškai komentuoja kintamuosius, kitus komentarus rašo su pagalba. Pagrindinis pasiekimų lygis – savarankiškai komentuoja programos kintamuosius ir veiksmus. Aukštesnysis lygis – analizuoja ir vertina bendraklasių atliktus darbus, suteikia pagalbą.
Žinios prieš	Geba dirbti Scratch ir Python aplinkose. Geba kurti nesudėtingas Python programas.

Galimybės taikyti spec. poreikių mokiniams	Spec. poreikių mokiniams galima pateikti sukurta projektą, kuriame jiems reikėtų tik pakeisti kažkokį komentarą, Pvz., pakeisti vieną komentaro žodį kitu.
Patarimai kolegoms, kurie naudos parengtą medžiagą	<ol style="list-style-type: none"> Užduotis ir pamokos pateikimą derinkite prie skirtingų gebėjimų turinčių mokinių. Jei klasėje mokiniai dirbo pagal nurodymus, parenkite praktines užduotis, kurios padėtų įvertinti mokinių pasiekimų lygius. Jei pažangesni mokiniai t.y. aukštesniojo lygio pasirinko alternatyvų būdą, nustatykite pristatymo vertinimo kriterijus ir įvertinkite mokinių darbus remiantis šiais kriterijais.

1 ETAPAS 🕒 10-15 minučių. Aiškinimas

Pradžioje išsiaiškinsime kas yra komentarai. Išnagrinėkime paprastą programą Scratch aplinkoje.



1 pav. Programos pavyzdys Scratch aplinkoje

Kiekvienoje programos dalyje surašytas aiškinamasis tekstas apie programos paskirtį, kintamuosius, atliekamus veiksmus. Programos naudotojas komentarų nemato. Komentarai ypač naudingi, kai skaitysite kodą vėliau, arba kiti žmonės peržiūrės jūsų programą.

Komentarai yra tekstinės pastabos, pridedamos prie programos, siekiant pateikti aiškinančią informaciją apie programos kodą. Jie naudojami programavimo kalboje, kad dokumentuotų programą ir primintų programuotojams, kokius sudėtingus dalykus jie ką tik padarė su kodu, taip pat padeda vėlesnei kartai suprasti, prižiūrėti ir modifikuoti programą.

Ta pati programa Python programavimo kalba atrodytų taip:

```
# Aritmetiniai veiksmai su dviem skaičiais

skaicius1 = 15 #Kintamojo skaicius1 reikšmė
skaicius2 = 12 #Kintamojo skaicius2 reikšmė

suma = skaicius1 + skaicius2 #Skaičiuojama dviejų skaičių suma
skirtumas = skaicius1 - skaicius2 #Skaičiuojamas dviejų skaičių skirtumas
sandauga = skaicius1 * skaicius2 #Skaičiuojama dviejų skaičių sandauga
dalmuo = skaicius1 / skaicius2 #Skaičiuojamas dviejų skaičių dalmuo

print(skaicius1, " + ", skaicius2, " = ", suma) # Ekrane rodoma apskaičiuota suma
print(skaicius1, " - ", skaicius2, " = ", skirtumas) # Ekrane rodomas apskaičiuotas skirtumas
print(skaicius1, " * ", skaicius2, " = ", sandauga) # Ekrane rodoma apskaičiuota sandauga
print(skaicius1, " / ", skaicius2, " = ", dalmuo) # Ekrane rodomas apskaičiuotas dalmuo
```

Naudotojas matys:

$$15 + 12 = 27$$

$$15 - 12 = 3$$

$$15 * 12 = 180$$

$$15 / 12 = 1.25$$

2 ETAPAS 🕒 25-30 minučių. Praktinis tyrimas

1 uždutis. Sukurta programa uždaviniui spręsti. Įrašykite komentarus, paaiškinančius programą.

Uždutis slenkstiniam lygiui. Pasidaryti konspektą apie komentarų svarbą.

Uždutis patenkinamam lygiui.

Stačiakampis. Stačiakampio ilgis yra lygus i centimetrų, plotis – p centimetrų. Parašykite programą, skaičiuojančią stačiakampio perimetrą ir plotą.

Pradiniai duomenys įvedami klaviatūra	Rezultatai, rodomi ekrane
---------------------------------------	---------------------------

Koks stačiakampio ilgis? 5	Stačiakampio perimetras: 14
Koks stačiakampio plotis? 2	Stačiakampio plotas: 10

Užduotis pagrindiniam lygiui.

Skola. Jurgis iš turimų j eurų Romui paskolino r eurų, o Simui – s eurų. Parašykite programą, skaičiuojančią kiek eurų k liko Jurgio piniginėje.

Pradiniai duomenys įvedami klaviatūra	Rezultatai, rodomi ekrane
Kelis eurus turėjo Jurgis? 5	Jurgiui liko 2 eurai (eurų).
Kelis eurus paskolino Romui? 2	
Kelis eurus paskolino Simui? 1	

Užduotis aukštesniajam lygiui.

Pietų kaina. Trys draugės Ieva, Saulė ir Jonė valgė valgykloje. Ievos maistas kainavo j eurų, Saulės – r , o Jonės – f eurų. Reikia parašyti programą, kuri suskaičiuotų, kiek eurų v iš viso sumokėta.

Pradiniai duomenys įvedami klaviatūra	Rezultatai, rodomi ekrane
Kiek eurų sumokėjo Ieva? 5	Iš viso sumokėta 13 eurų (eurai)
Kiek eurų sumokėjo Saulė? 4	
Kiek eurų sumokėjo Jonė? 4	

2 užduotis. Sugalvokite uždavinį, jį išspręskite, komentuokite programos dalis. Su draugu apsikeiskite uždaviniais, įvertinkite vienas kito darbą.

Pvz.: Rita ir Rima sprendžia uždavinius. Rita uždavinį išsprendė per rt minučių, Rima – per rm minučių. Programa turi nustatyti, kuri mergaitė uždavinį išsprendė greičiau.

```
# Rita ir Rima sprendžia uždavinius.
```

```
# Rita uždavinį išsprendė per rt minučių, Rima - per rm minučių.
```

```
# Programa turi nustatyti, kuri mergaitė uždavinį išsprendė greičiau.
```

```
rt = input("Per kiek laiko uždavinį išsprendė Rita? ") # Įvedamas Ritos laikas
```

```
rm = input("Per kiek laiko uždavinį išsprendė Rima? ") # Įvedamas Rimos laikas
```

```
if rt < rm: #Jei Ritos sprendimo laikas trumpesnis už Rimos
```

```
    print ("Greičiau išsprendė Rita") # Tuomet greičiau išsprendė Rita
```

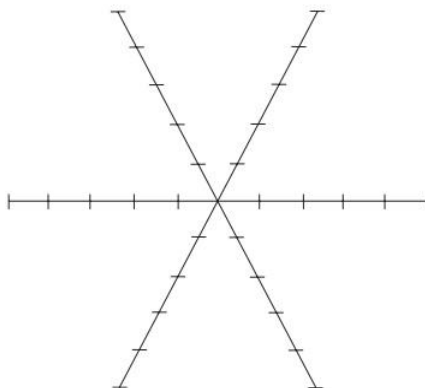
```
elif (rm < rt): # Jei Rimos sprendimo laikas trumpesnis už Ritos
    print ("Greičiau išsprendė Rima") # Tuomet greičiau išsprendė Rima
else:
    print ("Abi išsprendė per vienodą laiką") # Visais kitais atvejais merginos išsprendė
per vienodą laiką
```

3 ETAPAS 🕒 5-10 minutės. Apibendrinimas

Pamokai apibendrinti ir įsivertinti, naudosime voratinklio, žvaigždės formos šabloną. Figūrų ašyse mokiniai turėtų pažymėti kaip jiems sekėsi: suprato ar nesuprato temą. Mokytojas pasirenka ašių pavadinimus, atitinkantį pamokos tikslus. Mokiniai gali įvertinti tiek dalyko įgūdžius, tiek kompetencijas, jei nori.

Mokiniai, kurie su draugu apsikeitė darbais, įvertina vienas kito darbą.

ĮSIVERTINIMO METODAS „ŽVAIGŽDĖ“



Programos testavimas pažingsniui (Dovilė Žurauskienė)

Pasiekimų sritis	(B) Algoritmai ir programavimo mokymasis. 28.2.5. Programos testavimas pažingsniui.
Klasė	7-8 kl.
Tema	Programos testavimas pažingsniui.
Integruojami dalykai, pasiekimai	Užsienio kalba (anglų), matematika, gamtos mokslai.
Kompetencijos	Pažinimo – supranta programavimo paskirtį ir svarbą, ugdomi kritinio mąstymo įgūdžius. Skaitmeninė – mokiniai sugeba su pasitikėjimu naudoti skaitmenines priemones, siekdami produktyvaus ir efektyvaus mokymosi, supranta, kaip vyksta informacijos apdorojimas ir suvokia jo reikšmę.
Tikslas	Išsiaiškinti programos testavimą pažingsniui
Uždaviniai	1. Išsiaiškinti kaip vyksta programos testavimas pažingsniui. 2. Mokysitės testuoti programas pažingsniui.
Planuojamas rezultatas	1. Gebėsite paaiškinti programos testavimą pažingsniui. 2. Gebėsite testuoti programas pažingsniui.
Specifinės priemonės / programinė įranga	Multimedia projektorius, išmanioji lenta. Programinė įranga. Scratch aplinka: https://scratch.mit.edu/ Python aplinka: https://www.onlinegdb.com/ Internetiniai ir kiti šaltiniai: https://vaizdopamokos.lt/programavimas/c-programavimas/c-pamoka-debug-programos-vykdydas-pazingsniui-trasavimas-angl-debugging/ https://www.youtube.com/watch?v=b7pOcU1xMks
Mokymosi metodai	Diskusija, praktinis tyrimas, abipusis mokymas
Mokinių atlikto darbo vertinimas ir	Slenkstinis pasiekimų lygis – padedamas naudojasi programavimo aplinka, randa nurodytas komandas, atpažįsta pradinius duomenis ir rezultatus.

įsivertinimas (remiamasi [5])	<p>Patenkinamas pasiekimų lygis – padedamas atlieka nurodytus veiksmus programavimo aplinkoje, randa reikiamas komandas, atpažįsta rezultatus, vykdo pateiktą programą, patikrina ar pradiniai duomenys įvesti tinkamai.</p> <p>Pagrindinis pasiekimų lygis – programavimo aplinkoje randa reikiamas komandas, paaiškina programos vykdymo eigą, patikrina ar rezultatai išvesti tinkamai.</p> <p>Aukštesnysis lygis – savarankiškai naudojami programavimo aplinka, randa ir taiko uždaviniams spręsti tinkamas komandas, paaiškina gautus rezultatus ir programos vykdymo eigą, geba testuoti programą pažingsniui savarankiškai, sukuria testinius rinkinius.</p>
Žinios prieš	Geba surasti patikimą informaciją įvairiuose šaltiniuose, geba dirbti bendrojo naudojimo programine įranga.
Galimybės taikyti spec. poreikių mokiniams	Mokytojas scenarijų pritaiko priklausomai nuo vaiko specialiųjų poreikių. Jei sudėtingiau su pateikta informacija, mokytojas suteikia galimybę informaciją išnagrinėti pagal suprantamumą, o poto aiškinamsi kaip turi būti.
Patarimai kolegoms, kurie naudos parengtą medžiagą	<ol style="list-style-type: none"> 1. Įsitikinkite, kad kiekvienas mokinys supranta, kokia yra jų atliekamų užduočių prasmė ir tikslas, kad jie galėtų sėkmingai įgyvendinti užduotis. 2. Pritaikykite scenarijų, kad atitiktų skirtingų gebėjimų mokinių poreikius. Pasiūlykite papildomą pagalbą ar instrukcijas tiems mokiniams, kuriems gali reikėti papildomo palaikymo. 3. Parenkite klausimyną, kuriame būtų įtraukti įvairūs klausimai, pritaikyti skirtingų gebėjimų mokiniams. Suteikite mokiniams galimybę pasirinkti klausimus, kurie atitinka jų mokymosi lygį ir patirtį. 4. Sudarykite kriterijus, kuriuos mokiniai turės atitikti, pristatydami savo darbus. Įvertinkite jų darbus remiantis šiais kriterijais, suteikdami jiems aiškų supratimą, ką turėtų pasiekti. 5. Pateikite mokytojams patarimus, kaip skatinant mokinius dirbti savarankiškai. Suteikite jiems pagalbą, kuri padėtų mokiniams pasiekti savo tikslus.

1 ETAPAS 🕒 15-20 minučių. Aiškinimas, pavyzdys ir diskusija

Testavimas yra proceso, kai ieškoma klaidų programoje. Susiduriame su trimis pagrindinėmis klaidų kategorijomis: sintaksės, vykdymo metu kylančiomis ir loginėmis. Sintaksės klaidos atsiranda, kai programa nepasikliauja griežtomis taisyklėmis, kurių reikia laikytis programavime. Ištaisius visas sintaksės klaidas, gauname techniškai teisingą, tačiau ne visada logiškai teisingą programą. Loginės klaidos yra nustatomos, palyginus rankiniu būdu apskaičiuotus rezultatus su programos sukurtų rezultatų. Vykdymo klaidos dažniausiai kyla dėl blogo algoritmo, o jų rezultatas yra programos nutraukimas.

Vienas iš testavimo tikslų yra rasti klaidas, kurios gali turėti įtakos programos veikimui, tokioms kaip logikos klaidos, nenumatytos situacijos, klaidingi rezultatai ar programos užstrigimas. Taip pat testavimas padeda užtikrinti, kad programa atitinka numatytą elgseną pagal specifikacijas ar reikalavimus.

Testavimo metodai gali būti įvairūs, įskaitant rankinius ir automatinės testavimo procedūras. Automatizuotas testavimas yra vis populiariesnis, nes jis leidžia greičiau ir efektyviau testuoti įvairias programos funkcijas, o taip pat sumažina klaidų atsiradimo riziką.

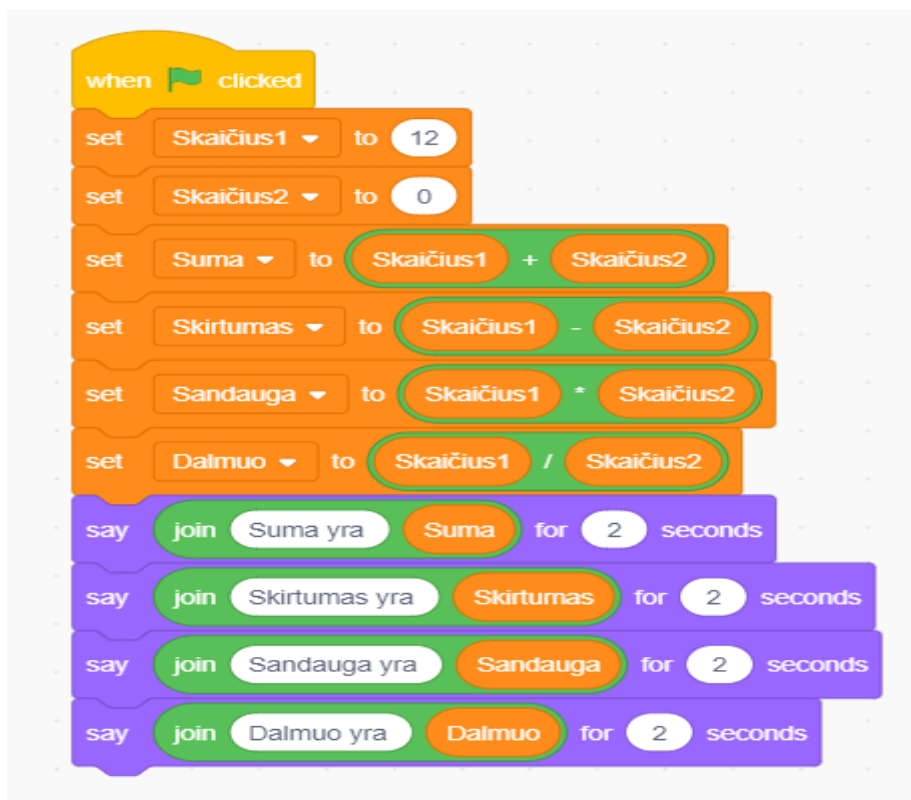
Be to, testavimas apima ne tik pagrindinių programos funkcijų tikrinimą, bet ir atsparumo įvairioms apkrovoms, atkūrimo galimybių patikrinimą po klaidų bei saugumo aspektų vertinimą. Taigi, tai yra būtina dalis kiekvieno programos kūrimo proceso, siekiant užtikrinti kokybę ir patikimumą.

Išsiaiškinsime kokie yra programos testavimo žingsniai:

- 1. Planavimas:** Prieš pradėdant testavimą, sukurti testavimo planą, kuriame nurodomi įvairūs testai, kuriuos ketinate atlikti.
- 2. Vieningas testavimas:** Užtikrinkite, kad kiekvieną programos dalį būtų galima atskirai testuoti. Tai padės greičiau nustatyti problemas.
- 3. Klaidų dokumentavimas:** Kai randate klaidą, įrašykite ją į klaidų žurnalą. Nurodykite, kaip atkartoti klaidą ir kokia yra tikimybė, kad ji pasitaikys.
- 4. Automatiniai testai:** Jei įmanoma, sukurti automatinius testus, kurie gali automatiškai tikrinti jūsų programos veikimą.
- 5. Rankinis testavimas:** Testuokite programą rankiniu būdu, įvedant įvairius duomenis ir tikrindami jos reakciją į skirtingas situacijas.
- 6. Atsiliepimai:** Paprašykite draugų ar mokytojų testuoti jūsų programą ir pateikti atsiliepimus. Tai gali padėti rasti klaidas, kurias galite nepastebėti.
- 7. Klaidų taisymas:** Jei atradote klaidų, taisykite jas ir testuokite programą iš naujo. Kartokite šį procesą, kol programa veiks be klaidų.

Pavyzdys: Ištestuosime Scratch aplinkoje sukurtą skaičiuotuvą. Kintamojo Skaičius2 reikšmę įvesime lygią nuliui. Kadangi dalyba iš nulio negalima, dalmuo bus neapibrėžtas (Infinity).

Paryškinta geltonai, nes testuosime rankiniu būdu.

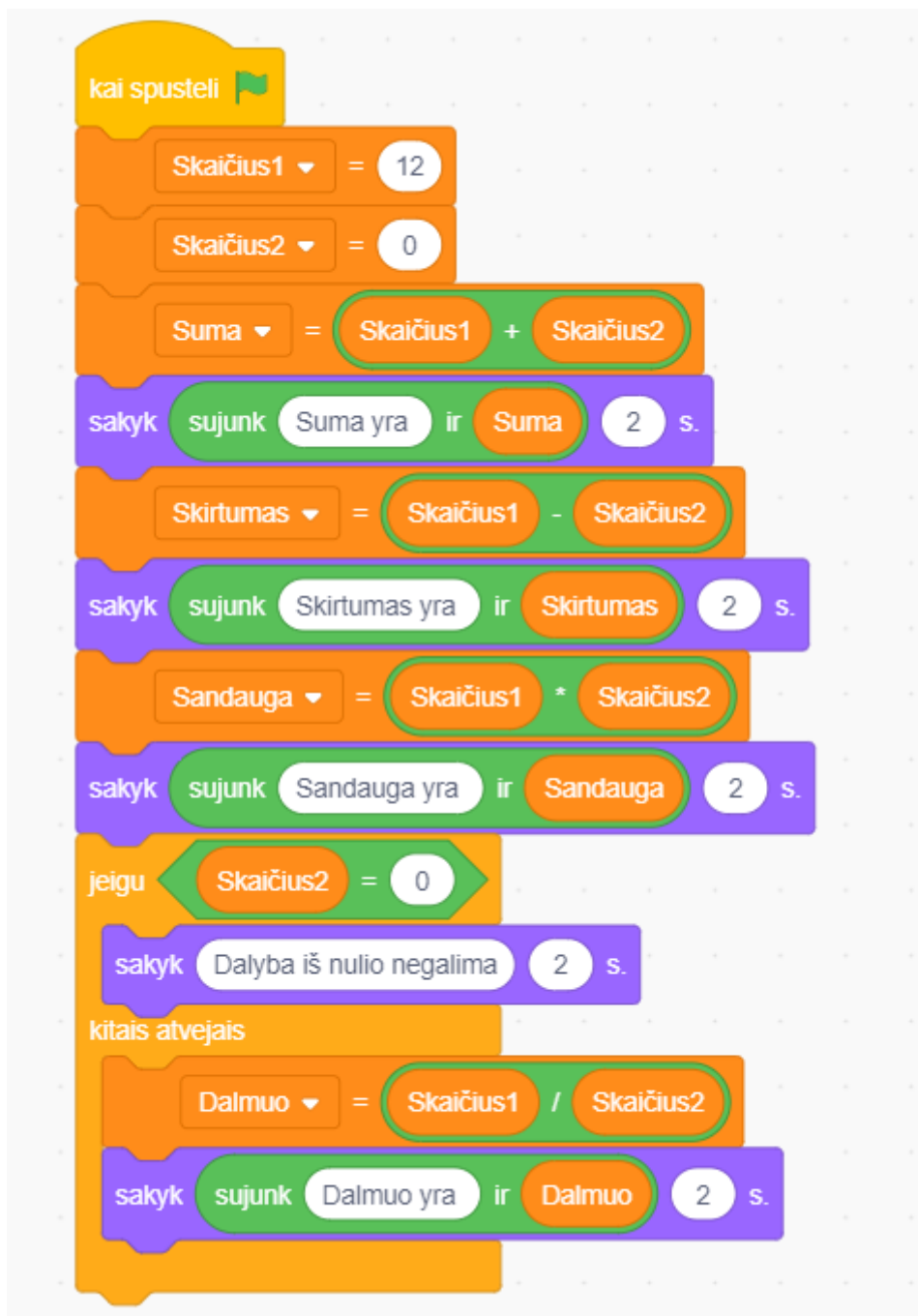


2 pav. Programos pavyzdys Scratch aplinkoje



3 pav. Programos pavyzdys Scratch aplinkoje

Kaip ištaisyti šią klaidą? Dalinant reikia patikrinti, kad vardiklis nebūtų lygus nuliui.



4 pav. Programos pavyzdys Scratch aplinkoje

Tą patį veiksmą atliksime ir Python programoje:

Aritmetiniai veiksmai su dviem skaičiais

```

skaicius1 = 15 #Kintamojo skaicius1 reikšmė
skaicius2 = 0 #Kintamojo skaicius2 reikšmė

suma = skaicius1 + skaicius2 #aičiuojama dviejų skaičių suma
print(skaicius1, " + ", skaicius2, " = ", suma) # Ekrane rodoma apskaičiuota suma
skirtumas = skaicius1 - skaicius2 #Skaičiuojamas dviejų skaičių skirtumas
print(skaicius1, " - ", skaicius2, " = ", skirtumas) # Ekrane rodomas apskaičiuotas skirtumas
sandauga = skaicius1 * skaicius2 #Skaičiuojama dviejų skaičių sandauga

```

```

print(skaicius1, " * ", skaicius2, " = ", sandauga) # Ekrane rodoma apskaičiuota sandauga
if (skaicius2 != 0):
    dalmuo = skaicius1 / skaicius2 #Skaičiuojamas dviejų skaičių dalmuo
    print(skaicius1, " / ", skaicius2, " = ", dalmuo) # Ekrane rodomas apskaičiuotas dalmuo
else:
    print("Dalyba iš nulio negalima")

```

Rezultatai ekrane

15 + 0 = 15

15 – 0 = 15

15 * 0 = 0

Dalyba iš nulio negalima

Pvz.: Rita ir Rima sprendžia uždavinius. Rita uždavinį išsprendė per r_t minučių, Rima – per r_m minučių. Programa turi nustatyti, kuri mergaitė uždavinį išsprendė greičiau.

```

# Rita ir Rima sprendžia uždavinius.
# Rita uždavinį išsprendė per  $r_t$  minučių, Rima - per  $r_m$  minučių.
# Programa turi nustatyti, kuri mergaitė uždavinį išsprendė greičiau.

rt = input("Per kiek laiko uždavinį išsprendė Rita? ") # Įvedamas Ritos laikas
rm = input("Per kiek laiko uždavinį išsprendė Rima? ") # Įvedamas Rimos laikas

if rt < rm: #Jei Ritos sprendimo laikas trumpesnis už Rimos
    print ("Greičiau išsprendė Rita") # Tuomet greičiau išsprendė Rita
elif (rm < rt): # Jei Rimos sprendimo laikas trumpesnis už Ritos
    print ("Greičiau išsprendė Rima") # Tuomet greičiau išsprendė Rima
else:
    print ("Abi išsprendė per vienodą laiką") # Visais kitais atvejais merginos išsprendė
per vienodą laiką)

```

Reikia patikrinti, ar programa tinkamai veikia, kai: 1) Rita išsprendė uždavinį greičiau: $r_t = 5$, $r_m = 7$; 2) Rima išsprendė uždavinį greičiau: $r_t = 10$, $r_m = 5$; 3) abi merginos išsprendė per vienodą laiką: $r_t = 7$, $r_m = 7$.

Rezultatai ekrane:

1 testas

Per kiek laiko uždavinį išsprendė Rita? 5

Per kiek laiko uždavinį išsprendė Rima? 7

Greičiau išsprendė Rita

2 testas

Per kiek laiko uždavinį išsprendė Rita? 10

Per kiek laiko uždavinį išsprendė Rima? 5

Greičiau išsprendė Rima

3 testas

Per kiek laiko uždavinį išsprendė Rita? 7

Per kiek laiko uždavinį išsprendė Rima? 7

Abi išsprendė per vienodą laiką

2 ETAPAS 🕒 15-20 minučių. Praktinis tyrimas

1 uždavinys. Lietuvių liaudies pasakose karžygys Jonas, jojantis išvaduoti karalaitės, dažnai turi rinktis vieną iš 4 kelių. Jei Jonas pasirinks pirmą kelią, jo laukia neižengiama giria, jei antrą – kautynės su piktomis raganomis, jei trečią – tiesus kelias pas karalaitę, jei ketvirtą – kautynės su piktuoju slibinu. Parašysime programą, kuri, nurodžius, kokį iš kelių pasirinko Jonas, pasakys, kur kelias veda. Kelias įvedamas kaip sveikasis skaičius.

```
def nustatyti_kelias(kelias):  
    if kelias == 1:  
        return "Neižengiama giria"  
    elif kelias == 2:  
        return "Kautynės su piktomis raganomis"  
    elif kelias == 3:  
        return "Tiesus kelias pas karalaitę"  
    elif kelias == 4:  
        return "Kautynės su piktuoju slibinu"  
    else:  
        return "Tokio kelio nėra"  
  
# Vartotojo įvesties gavimas  
kelias = int(input("Įveskite pasirinktą kelią (1, 2, 3 arba 4): "))  
  
# Kvieskite funkciją nustatyti_kelias ir spausdinkite rezultatą  
kelio_aprasymas = nustatyti_kelias(kelias)
```

```
print(f"Jonas pasirinko kelią: {kelias}, kuris veda į {kelio_aprasymas}")
```

Reikia patikrinti, ar programa tinkamai veikia, kai: 1) Jei pasirinko 1 kelią, laukia neįžengiama giria 2) Jei pasirinko 2 kelią, laukia kautynės su piktomis raganomis 3) Jei pasirinko 3 kelią, tiesus kelias pas karalaitę 4) Jei pasirinko 4 kelią, kautynės su piktuoju slibinu.

Rezultatai ekrane:

1 bandymas

Įveskite pasirinktą kelią (1, 2, 3 arba 4): 1

Jonas pasirinko kelią: 1, kuris veda į neįžengiamą girią

2 bandymas

Įveskite pasirinktą kelią (1, 2, 3 arba 4): 2

Jonas pasirinko kelią: 2, kuris veda į kautynes su piktomis raganomis

3 bandymas

Įveskite pasirinktą kelią (1, 2, 3 arba 4): 1

Jonas pasirinko kelią: 3, kuris veda Tiesiai pas karalaitę

4 bandymas

Įveskite pasirinktą kelią (1, 2, 3 arba 4): 4

Jonas pasirinko kelią: 1, kuris veda į kautynes su piktuoju slibinu

2 uždavinys. Augustė parengė draugams kalėdines dovanėles ir supakavo jas į mažas dėžutes. Iš viso n dėžučių. Ji sugalvojo dovanėles paslėpti didelėse dėžėse, į kiekvieną kurių telpa po a mažų dėžučių. Parašykite programą, skaičiuojančią, kelių didelių dėžių k reikės Augustei, kad paslėptų kalėdines dovanas. Visi duomenys yra sveikieji skaičiai. Pasitikrinkite: kai $n = 10$, $a = 5$, $k = 2$. Kai $n = 10$, $a = 3$, $k = 4$.

```
# Įvedami duomenys
```

```
n = int(input("Įveskite kiekvienų dėžučių skaičių n: "))
```

```
a = int(input("Įveskite, kiek mažų dėžučių telpa didelėje dėžėje a: "))
```

```
# Skaičiuojame, kiek reikės didelių dėžių
```

```
k = n // a
```

```

# Tikriname, ar reikės papildomos dėžutės dėžėse
if n % a != 0:
    k += 1

print(f"Reikės {k} didelių dėžių Augustei, kad paslėptų kalėdines dovanas.")

```

Reikia patikrinti, ar programa tinkamai veikia, kai: 1) Pasitikrinkite: kai $n = 10$, $a = 5$, $k = 2$ 2) Kai $n = 10$, $a = 3$, $k = 4$.

Rezultatai ekrane:

1 rezultatas:

kai $n = 10$, $a = 5$, $k = 2$

Įveskite kiekvienų dėžučių skaičių $n = 10$

Įveskite, kiek mažų dėžučių telpa didelėje dėžėje $a = 5$

Reikės 2 didelių dėžių Augustei, kad paslėptų kalėdines dovanas.

2 rezultatas:

$n = 10$, $a = 3$, $k = 4$.

Įveskite kiekvienų dėžučių skaičių $n = 10$

Įveskite, kiek mažų dėžučių telpa didelėje dėžėje $a = 3$

Reikės 4 didelių dėžių Augustei, kad paslėptų kalėdines dovanas.

3 ETAPAS 🕒 5-10 minučių. Apibendrinimas

Su mokiniais apibendrinamas atliktas darbas.

Nuotaikinga, saldi refleksija

Pamokos apibendrinimas išmoktą medžiagą arba atlikus grupėse darbus, galima mokiniams pristatyti įvairių skonių skirtingų rūšių užkandžių. Kiekvienas užkandis turi simbolinę reikšmę:

- Obuoliukai – kas jus sužavėjo?
- Bananai – kuo abejojate arba ką norėtumėte geriau suprasti?
- Persikai – kas jus nuliūdino arba sukėlė nerimą?
- Braškės – su kuo visiškai sutinkate arba ko palaikote?

- Ananasai – kas jus nustebino arba išplėtė jūsų žinias?
- Kriaušės – kas jus nudžiugino arba paliko teigiamą įspūdį?

Tai ne tik suteikia malonią užkandžių patirtį, bet ir skatina mokinius išreikšti savo mintis ir jausmus, plėtoja bendravimo ir refleksijos įgūdžius.

Programos testavimas naudojant pasirinktos aplinkos testavimo priemones (Dovilė Žurauskienė)

Pasiekimų sritis	(B) Algoritmai ir programavimo mokymasis. 28.2.5. Programos testavimas
Klasė	7-8 kl.
Tema	Programos testavimas naudojant pasirinktos aplinkos testavimo priemones.
Integruojami dalykai, pasiekimai	Lietuvių kalba, technologijos, užsienio kalba (anglų), dailė
Kompetencijos	„Pažinimo – motyvuojami gilintis į problemų sprendimo automatizavimo galimybes, išvelgti, kaip panaudoti algoritmavimą ir programavimą sprendžiant realaus gyvenimo uždavinius, pritaikyti įgytas dalyko žinias ir išugdytus gebėjimus idėjoms formuluoti; mokomi duomenų ir informacijos analizei pasitelkti skaitmenines technologijas, aptarti sprendimo procesą, argumentuoti sprendimų pasirinkimą, daryti pagrįstas išvadas. Skaitmeninė – užtikrintai ir sumaniai naudojasi skaitmeninėmis priemonėmis siekdami veiksmingo ir konstruktyvaus mokymo(si), suvokia informacijos apdorojimo procesus, jų svarbą“.[5]
Tikslas	Susipažinti su programos testavimu naudojant pasirinktos aplinkos testavimo priemones
Uždaviniai	<ol style="list-style-type: none"> 1. Išsiaiškinsite kaip vyksta programos testavimas su pasirinktomis aplinkos testavimo priemonėmis. 2. Mokysitės testuoti programas naudojant pasirinktas testavimo priemones.
Planuojamas rezultatas	<ol style="list-style-type: none"> 1. Gebėsite paaiškinti programos testavimą naudojant pasirinktos aplinkos testavimo priemones. 2. Gebėsite programas testuoti.

<p>Specifinės priemonės / programinė įranga</p>	<p>Multimedia projektorius, išmanioji lenta.</p> <p>Programinė įranga.</p> <p>Python aplinka: https://www.onlinegdb.com/</p> <p>Internetiniai ir kiti šaltiniai:</p> <p>https://www.onlinegdb.com/blog/brief-guide-on-how-to-use-onlinegdb-debugger/</p> <p>https://www.gdbtutorial.com/tutorial/how-use-gdb-example</p> <p>https://vaizdopamokos.lt/programavimas/c-programavimas/c-pamoka-debug-programos-vykdydas-pazingsniui-trasavimas-angl-debugging/</p> <p>https://www.youtube.com/watch?v=b7pOcU1xMks</p>
<p>Mokymosi metodai</p>	<p>Diskusija, praktinis tyrimas, mokinių darbų aplankas, mokinių patirties išsiaiškinimas.</p>
<p>Mokinių atlikto darbo vertinimas ir įsivertinimas (remiamasi [5])</p>	<p>Slenkstinis pasiekimų lygis – testuoja programą su mokytojo pateiktais duomenimis ir vertina programos rezultatų teisingumą.</p> <p>Patenkinamas pasiekimų lygis – padedamas sudaro testų rinkinius ir testuoja programą. Vertina programos patogumą naudotojui.</p> <p>Pagrindinis pasiekimų lygis – kritiškai vertina programos rezultatus, jų pateikimą, patogumą naudotoju.</p> <p>Aukštesnysis lygis – vertina ir tobulina programos sąsają su naudotoju.</p>
<p>Žinios prieš</p>	<p>Geba surasti patikimą informaciją įvairiuose šaltiniuose, geba dirbti bendrojo naudojimo programine įranga.</p>
<p>Galimybės taikyti spec. poreikių mokiniams</p>	<p>Mokytojas scenarijų pritaiko priklausomai nuo vaiko specialiųjų poreikių. Jei sudėtingiau su pateikta informacija, mokytojas suteikia galimybę informaciją išnagrinėti pagal suprantamumą, o poto aiškinamasi kaip turi būti.</p>
<p>Patarimai kolegoms, kurie naudos parengtą medžiagą</p>	<p>1. Pirmiausia, išbandykite visas užduotis patys, kad gerai suprastumėte procesą ir galimus iššūkius. 2. Pritaikykite užduotis taip, kad jos būtų tinkamos įvairių lygių mokiniams, atsižvelgiant į jų individualius gebėjimus. 3. Paruoškite klausimyną, kuris padės įvertinti, kaip gerai mokiniai suprato ir atliko užduotis pagal scenarijų. 4. Nustatykite vertinimo kriterijus aukštesnio lygio užduotims ir naudokite juos vertindami mokinių darbus, kai jie pasirinks sudėtingesnes užduotis. 5. Suteikite mokiniams galimybę dirbti savarankiškai, o jūs būkite pasiekiami, kad galėtumėte jiems patarti ir padėti, kai to prireiks.</p>

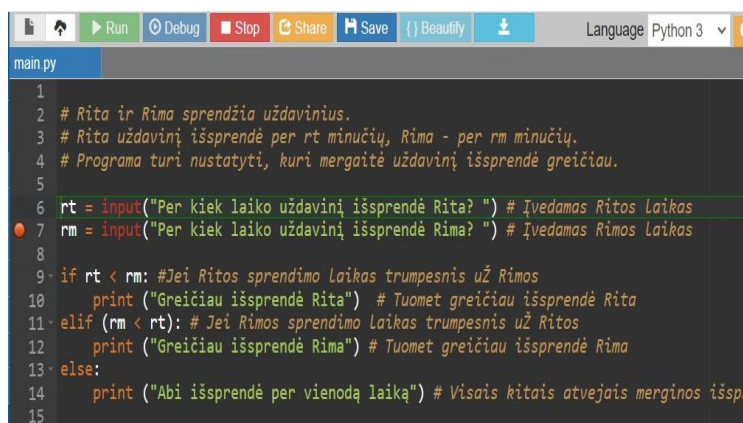
1 ETAPAS 🕒 15-20 minučių. Aiškinimas, pavyzdys ir diskusija

Programavimas – tai uždavinio sprendimo eigos aprašymas kompiuteriui suprantama forma. Sudėtingi uždaviniai reikalauja papildomų etapų: uždavinio analizės, sprendimo būdo ir kelio parinkimo, sprendimo algoritmo aprašymo, jo pavertimo programa, programos testavimo ir derinimo. Parašius ir paleidus porą kodo eilučių, vietoje norimo rezultato gali pasirodyti tuščias ekranas arba naršyklėje iškils klaida. Išdėstysiu planą, kaip sutvarkyti kodą ir atrasti netikėtas klaidas. Programavime labai dažnai susiduriame su sintaksės arba rašybos klaidomis. Jų rasti paprastai nėra sudėtinga. Pavyzdžiui, kai naudojame simbolius, kurie neturėtų būti panaudoti kode, gali iškilti sintaksės klaida. Taip pat neretai klaidų priežastis yra netaisyklingai parašyti funkcijų ar kintamųjų pavadinimai, dėl to gauname klaidą, nurodančią, jog toks kintamasis neegzistuoja. Kita, sudėtingesnė klaidų kategorija, yra loginės taip pat ir mąstymo klaidos. Dažnai įsivaizduojame, kad kodas veikia vienaip ar kitaip, tačiau realybėje gali būti visiškai ne taip. Išsiaiškinti loginių klaidų priežastis gali būti sunku, ypač naujiems programuotojams. Suvokimas, kad kažką darai netinkamai, dažnai ateina tik su patirties gausa. Be to, loginės klaidos gali atsirasti, kai veikianti programa negali tinkamai apdoroti parametru, kurių negalime numatyti. Pavyzdžiui, bandydami išsaugoti vartotojo vardą sistemoje, galime nepagalvoti, kad įvedant duomenis į tam skirtą laukelį, juos reikia perrašyti ne tik pagal raidžių, bet ir kitus simbolius. Pradžioje išsiaiškinsime kas yra Debugger?

Apskritai, derintuvas yra programa, įrankis, kuris leidžia paleisti programą ir kontroliuoti jos vykdymą bei stebėti būseną, ypač kai programa yra pristabdyta. GDB yra toks derintuvas, naudojamas C/C++ programoms derinti. „OnlineGDB“ suteikia sąsają, leidžiančią naudoti GDB debesies aplinkoje iš jūsų naršyklės.

Kaip galiu kontroliuoti programos vykdymą?

Debuggeriui galime pasakyti, kada pristabdyti programą, nustatydami lūžio taškus. Norėdami nustatyti lūžio tašką, spustelėkite tuščią sritį, esančią redaktoriaus eilutės numerio kairėje. Kai spustelėsite jį, turėtų būti rodomas raudonas apskritimas; o tai reiškia, kad toje eilutėje yra nustatytas lūžio taškas. Čia yra vaizdas, kaip jis atrodo.

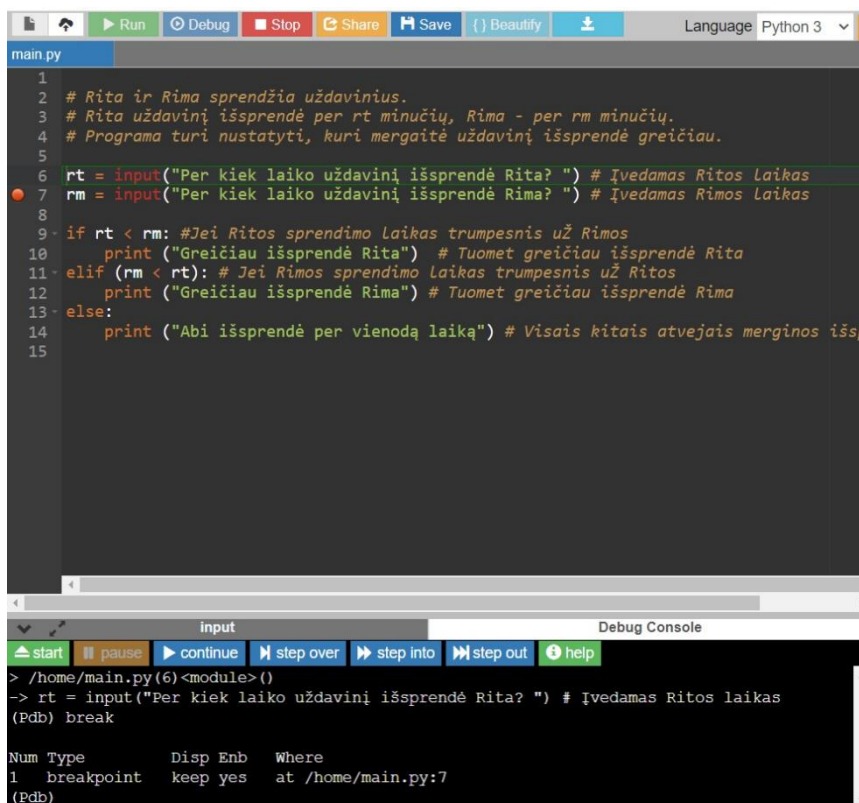


```
main.py
1
2 # Rita ir Rima sprendžia uždavinius.
3 # Rita uždavinį išsprendė per rt minučių, Rima - per rm minučių.
4 # Programa turi nustatyti, kuri mergaitė uždavinį išsprendė greičiau.
5
6 rt = input("Per kiek laiko uždavinį išsprendė Rita? ") # Įvedamas Ritos laikas
7 rm = input("Per kiek laiko uždavinį išsprendė Rima? ") # Įvedamas Rimos laikas
8
9 if rt < rm: #Jei Ritos sprendimo laikas trumpesnis už Rimos
10     print("Greičiau išsprendė Rita") # Tuomet greičiau išsprendė Rita
11 elif (rm < rt): # Jei Rimos sprendimo laikas trumpesnis už Ritos
12     print("Greičiau išsprendė Rima") # Tuomet greičiau išsprendė Rima
13 else:
14     print("Abi išsprendė per vienodą laiką") # Visais kitais atvejais merginos išsp
15
```

5 pav. Programos

pavyzdys

Kai nustatote lūžio tašką, kai paleisite programą derinimo režimu, ji pristabdys vykdymą, kai programa pasieks eilutę, kurioje nustatytas lūžio taškas.



6 pav. Programos pavyzdys

Dabar galime naudoti žingsnių komandas, kad programa būtų vykdoma eilutė po eilutės.



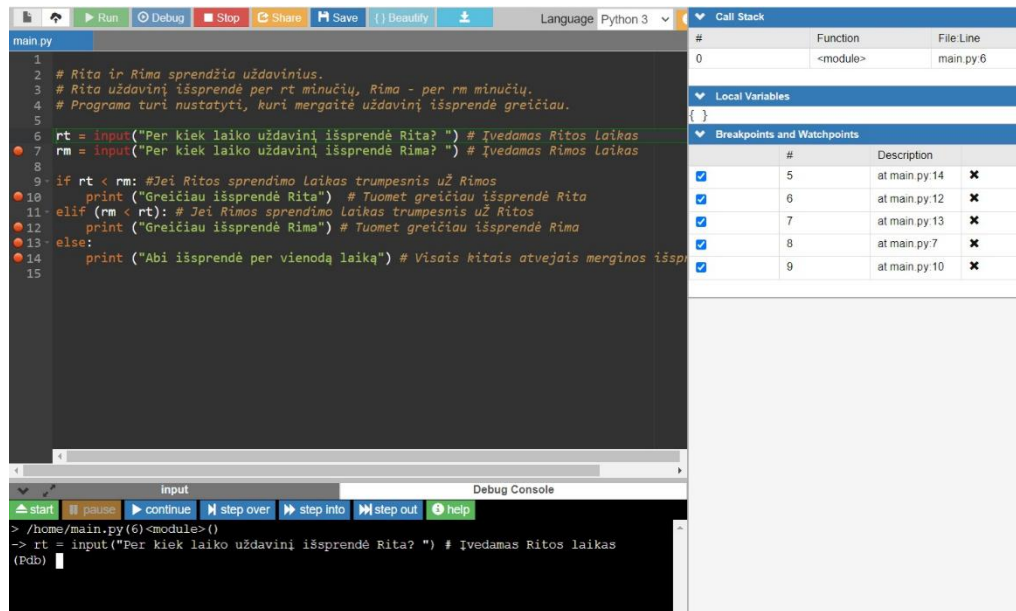
7 pav. Programos pavyzdys

1. Tęsti – Tęsti programos vykdymą, kol bus pasiektas kitas lūžio taškas
2. Įeiti – Vykdyti programą eilutė po eilutės, pereinant prie funkcijos
3. Peržengti – Vykdyti programą eilutė po eilutės, bet neįeiti į funkcijos iškvietimą
4. Išeiti – Tęsti programą vykdydamas, kol baigsis esama funkcija.

Ką galiu pamatyti, kai programa pristabdyta? Galite matyti iškvietimų krūvą ir dabartinės funkcijos vietinių kintamųjų reikšmes. Iškvietimų kaminas rodo funkcijų iškvietimų grandinę. Kaip matyti toliau pateiktame paveikslėlyje, „pagrindinė“ funkcija

Call Stack		
#	Function	File:Line
Local Variables		
Breakpoints and Watchpoints		
#	Description	
<input checked="" type="checkbox"/>	1	at main.py:12 ✘
<input checked="" type="checkbox"/>	2	at main.py:11 ✘
<input checked="" type="checkbox"/>	3	at main.py:10 ✘

išskviečia funkciją „factorial“, o „factorial“ išskviečia funkciją „dauginti“.



8 pav.
Programos
pavyzdys

Čia yra vienas vaizdas, vaizduojantis „OnlineGDB“ derinimo priemonės naudojimą.

Dabar mes kontroliuojame ir stebime programos vykdymą, todėl galime patikrinti programos vykdymo srautą. Lūžio taškai, žingsniavimo komandos ir kintamųjų rodiniai yra pagrindinė derinimo priemonės savybė.

2 ETAPAS 🕒 15-20 minučių. Praktinis tyrimas

Visiems mokiniams su mokytojo (slenkstinio pasiekimo lygio mokiniams pagalba) bandyti atlikti praktinę užduotį.

Čia yra programa, kurios elgesys yra klaidingas.

Ši programa parašyta skaičiavimui Nr. metų, per kuriuos buvo pasiektas tikslinis balansas, esant dabartiniam likučiui ir metinei palūkanų normai.

Tačiau tai suteikia 1 metų rezultatą bet kokiai balanso ar tikslo vertei.

```
/* File: buggy.c */
#include <stdio.h>
int main()
{
    int balance=100;
```

```

int target=1000;

float rate = 0.1;

int year = 0;

do
{
    float interest = balance * rate;

    balance = balance + interest;

    year++;

} while ( balance >= target );

printf("%d No. of years to achieve target balance.\n", year);

return 0;

}

```

Čia pateikiami šios programos derinimo veiksmai.

1 veiksmas: sukompiliuokite ir sukurkite programą su derinimo simboliais `$ gcc -g buggy.c`

2 veiksmas: paleiskite programą su GDB `$ gdb a.out`

3 veiksmas: nustatykite pagrindinės funkcijos pertraukos tašką. `(gdb) b main`
 Breakpoint 1 at 0x400535: file buggy.c, line 5.

4 veiksmas: paleiskite programą Kadangi pagrindinėje funkcijoje buvo nustatytas pertraukimas, programa sustos ties pagrindine funkcija ir lauks gdb komandos. `(gdb) run`
 Starting program: a.out

Breakpoint 1, main () at buggy.c:5
 5 int balance=100;

5 veiksmas: žingsniuokite, kol pasieksime 13 eilutę `(gdb) s`
 6 int target=1000; `s`
`(gdb)` `s`
 7 float rate = 0.1; `s`
`(gdb)` `s`
 8 int year = 0; `s`
`(gdb)` `s`
 11 float interest = balance * rate; `s`
`(gdb)` `s`
 12 balance = balance + interest; `s`
`(gdb)` `s`
 13 year++;

```

6 veiksmas: spausdinkite likučio vertę, normą, palūkanas (gdb) p balance
$1 = 110
(gdb) p rate
$2 = 0.100000001
(gdb) p interest
$3 = 10

```

```

7 veiksmas: žingsniuokite tol, kol pasieksime 15 eilutę (gdb) s
14 } while ( balance >= target );
(gdb) s
15 printf("%d No. of years to achieve target balance.\n", year);

```

8 veiksmas: spausdinkite tikslo, metų, balanso vertę Likutis yra 110, o tikslas yra 1000. Tai atrodo gerai. Sąlygos, ty balansas >= tikslas, bus false. Ir kilpa nutrūks. Taigi čia situacija netinkama. Tai turėtų būti balansas < tikslas. Sveikiname! Suderinote savo pirmąją klaidingą programą. (gdb) p

```

target
$5 = 1000
(gdb) p year
$6 = 1
(gdb) p balance
$7 = 110

```

3 ETAPAS 🕒 5-8 minutės. Apibendrinimas

Su mokiniais apibendrinamas atliktas darbas.

1. Jei buvo dirbta pagal numatytą planą, galite paprašyti juos įvertinti visus eigos etapus užpildant mokytojo paruoštą formą arba dalinant kiekvienam mokiniui lapelius su nebaigtais sakiniais, kuriuos jie tęsia ir pasidalina su kitais draugais arba klasių draugais.

Refleksijai pateikiami klausimai:

- Kaip jautiesi, vykdant užduotį? Kodėl?
- Kaip sekėsi užduoties atlikimas? Kodėl?
- Kodėl užduotis tau patiko?
- Ką žinojai iš anksto, prieš atliekant užduotį?
- Ką naujo sužinojai, atlikęs užduotį?
- Kuri užduotis buvo sunkiausia? Kodėl?
- Ar prireikė pagalbos?
- Koks pažymys atitiktų tavo darbą? Kodėl?
- Ką galėtum patobulinti?

- Kaip galėtum veikti kitaip?
 - Kur pritaikytum įgytas žinias praktikoje?
2. Jei patys mokiniai atliko užduotis, tada trumpai apibūdina atliktus darbus. Darbai vertinami pagal iš anksto numatytus tam tikrus kriterijus.

Programos rezultatų pateikimas diagramomis (Ina Liniova)

Pasiekimų sritis	28.2 Algoritmų ir programavimo mokymas(is) 28.2.6. Programos rezultatų pateikimas. Mokomasi numatyti uždavinio sprendimo rezultatus ir pasirinkti jų pateikimo formą diagramomis.
Klasė	7-8 kl.
Tema	Programos rezultatų pateikimas diagramomis
Integruojami dalykai, pasiekimai	Lietuvių kalba, Anglų kalba, Matematika.
Kompetencijos [2]	Pažinimo – mokiniai ugdomi informatinį mąstymą, motyvuojami gilintis į problemų sprendimo automatizavimo galimybes, išvelgti, kaip panaudoti algoritmavimą ir programavimą sprendžiant realaus gyvenimo uždavinius, pritaikyti įgytas dalyko žinias ir išugdytus gebėjimus idėjoms formuluoti; mokomi duomenų ir informacijos analizei pasitelkti skaitmenines technologijas, aptarti sprendimo procesą, argumentuoti sprendimų pasirinkimą, daryti pagrįstas išvadas. Mokiniai skatinami nuolat reflektuoti savo mokymąsi, į(si)vertinti patirtį ir pažangą, mokytis iš klaidų, išsikelti naujus mokymo(si) tikslus. Skaitmeninė – mokiniai mokosi kurti skaitmeninį turinį programuodami. Komunikavimo – bendravimas grupėje, atliktų darbų pristatymas, pasidalinimas idėjomis, diskusijos.
Tikslas	Mokytis pateikti programos darbo rezultatus diagramomis.
Uždaviniai	<ol style="list-style-type: none"> 1. Išsiaiškinsite, kokiais atvejais rezultatus patogiau pateikti diagramomis. 2. Mokysitės rezultatus pateikti diagramomis Python programavimo kalba. 3. Įgūdžiams įtvirtinti spręsite realaus pasaulio uždavinius.
Planuojamas rezultatas	<ol style="list-style-type: none"> 1. Gebėsite paaiškinti, kokiais atvejais rezultatus patogiau pateikti diagramomis. 2. Gebėsite pateikti rezultatus diagramomis Python programavimo kalba.
Specifinės priemonės / programinė įranga	<p>Minčių žemėlapių kūrimui skirti įrankiai, pvz. Canva, MindMap, https://www.iklase.lt/minciu_zemelapiai/ ir pan.</p> <p>Python programavimo aplinka, pvz. Visual Studio 2022, NŠA siūloma aplinka Google Colab: https://colab.research.google.com/notebooks/charts.ipynb https://matplotlib.org/stable/tutorials/pyplot.html</p> <p>Kompiuterių klasė. Interaktyvi lenta.</p>

Mokymosi metodai	Metodas „Minčių žemėlapis“ naudojamas įgyvendinant pirmąjį pamokos uždavinį. Praktinis tyrimas mokantis tinkamai pavaizduoti rezultatus diagramomis (įgyvendinant 2 ir 3 pamokos uždavinius).
Mokinių atlikto darbo vertinimas ir įsivertinimas	Slenkstinis lygis – padedamas naudojasi programavimo aplinka, randa nurodytas komandas, atpažįsta rezultatus (B2.1). [3] Patenkiamas lygis – padedamas atlieka nurodytus veiksmus programavimo aplinkoje, randa reikiamas komandas, atpažįsta rezultatus, vykdo pateiktą programą (B2.2). [3]
	Pagrindinis lygis – programavimo aplinkoje randa reikiamas komandas, paaiškina programos vykdymo eigą, parodo rezultatus (B2.3). [3] Aukštesnysis lygis – savarankiškai naudojami programavimo aplinka, randa ir taiko uždaviniams spręsti tinkamas komandas, paaiškina gautus rezultatus ir programos vykdymo eigą (B2.4). [3]
Žinios prieš	Gebėti naudotis minčių žemėlapių kūrimo programomis, turėti darbo Python aplinkoje pagrindus.
Galimybės taikyti spec. poreikių mokiniams	Užduotys adaptuojamos pagal SUP poreikius. Mokiniais galima skirti papildyti minčių žemėlapi, kuriamas programos, sumažinti užduočių skaičių.
Patarimai kolegoms	Užduotis rekomenduojama papildyti pačių sukurtomis užduotimis. Mokiniai turi būti skatinami dirbti savarankiškai, mokytojas yra tik pagalbininkas ir konsultantas. Nurodytus šaltinius papildyti įvairesniais. Skatinti mokinius suteikti pagalbą klasės draugams.

1 ETAPAS ☐ 10 min Praktinis darbas „Nubraižyti minčių žemėlapi“

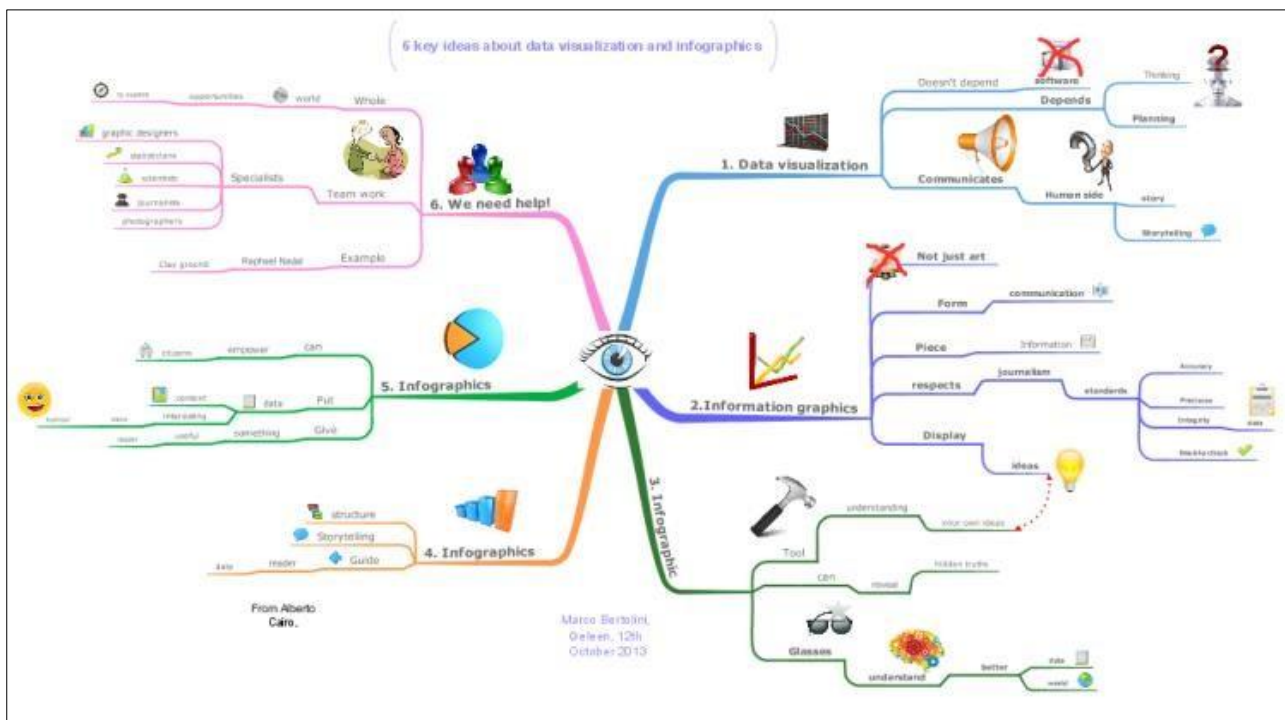
Mokiniais formuluojama užduotis: sukurkite minčių žemėlapi, kuriame būtų pavaizduota, kokiais atvejais patogiau atvaizduoti rezultatus diagramomis.

Mokiniais gali būti pateikiami jau sukurtų minčių žemėlapių pavyzdžiai 14 lentelė.



4 pav. Minčių žemėlapis „Geogeometry“

<https://gogeometry.com/data-visualization/data-visualization-mind-mapping-software.htm>



5 pav. Minčių žemėlapis „Biggerplate“

<https://www.biggerplate.com/mindmaps/nip/CdnR/6-key-ideas-about-data-visualization>

2 ETAPAS (30 minučių). Praktinis tyrimas kuriant diagramas Python

1 pavyzdys. Sukursime tiesinę diagramą, vaizduojančią, kaip kito mokyklų skaičius Lietuvoje 2011-2020 m. Duomenis iš 14 lentelės (šaltinis: <https://osp.stat.gov.lt/lietuvos-svietimas-kultura-irsportas-2021/bendrasis-mokyklinis-ugdymas>) perkelsime į Python programą.

14 lentelė. Duomenys Python programai

Metai	Mokyklų skaičius	Mokyklų skaičiaus mažėjimas
2011	1309	0
2012	1242	67
2013	1208	34
2014	1200	8
2015	1193	7
2016	1151	42
2017	1125	26
2018	1089	36
2019	1056	33

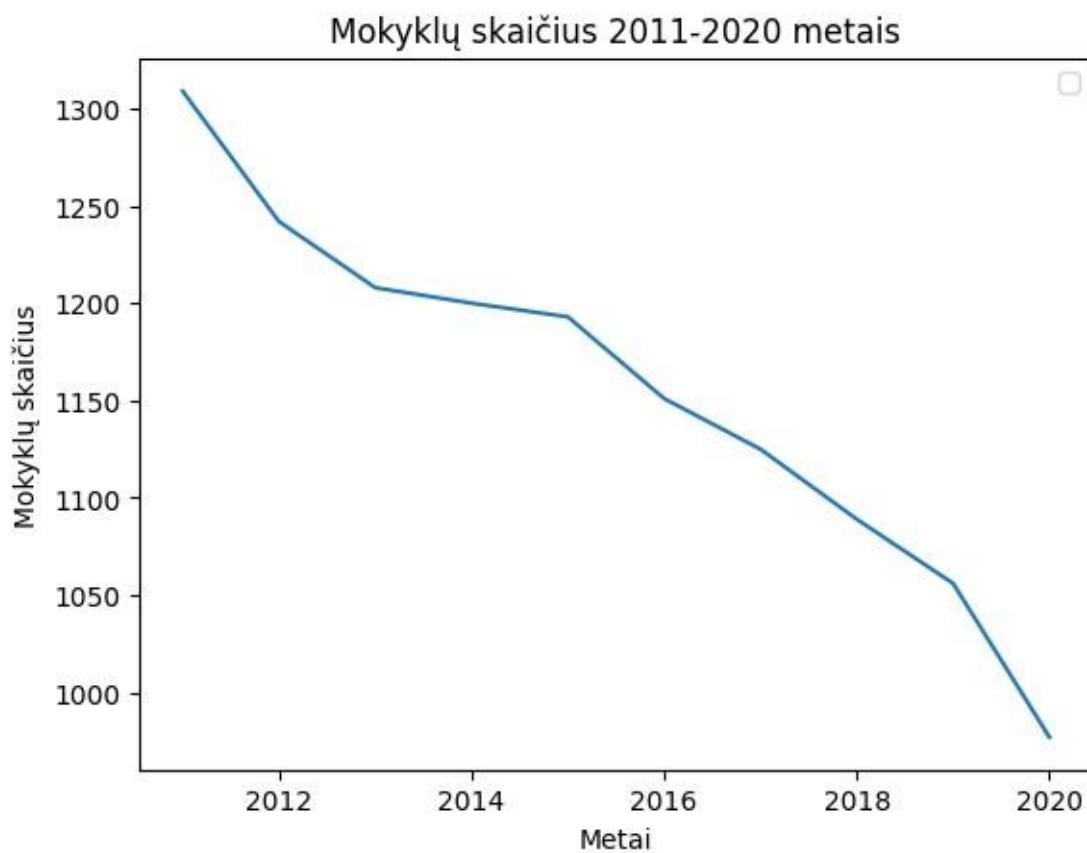
2020	977	79
------	-----	----

```

import
matplotlib.pyplot
as plt
x = [2011, 2012, 2013, 2014, 2015, 2016, 2017,
2018, 2019, 2020] y = [1309, 1242, 1208, 1200, 1193,
1151, 1125, 1089, 1056, 977] plt.plot(x, y,
label="") plt.plot()
plt.xlabel("Metai")
plt.ylabel("Mokyklų skaičius")
plt.title("Mokyklų skaičius
2011-2020 metais") plt.legend()
plt.show()

```

6 pav. Python programos kodas

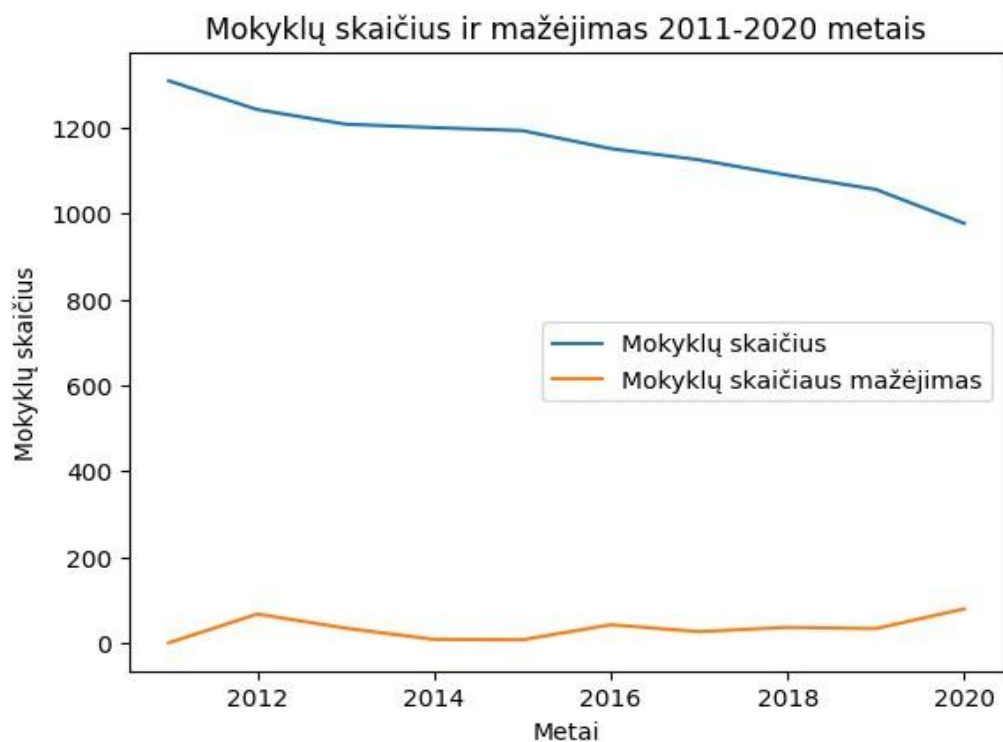


7 pav. Tiesinė diagrama

Papildykime programos kodą dar vienu duomeniu, parodančiu, keliomis mokyklomis sumažėjo per metus.

```
import matplotlib.pyplot as plt
x = [2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020]
y1 = [1309, 1242, 1208, 1200, 1193, 1151, 1125, 1089, 1056, 977]
y2 = [0, 67, 34, 8, 7, 42, 26, 36, 33, 79] plt.plot(x, y1,
label="Mokyklų skaičius") plt.plot(x, y2, label="Mokyklų skaičiaus
mažėjimas") plt.plot()
plt.xlabel("Metai") plt.ylabel("Mokyklų skaičius")
plt.title("Mokyklų skaičius ir mažėjimas 2011-2020 metais")
plt.legend() plt.show()
```

8 pav. Papildytas Python programos kodas



9 pav. Papildyta tiesinė diagrama

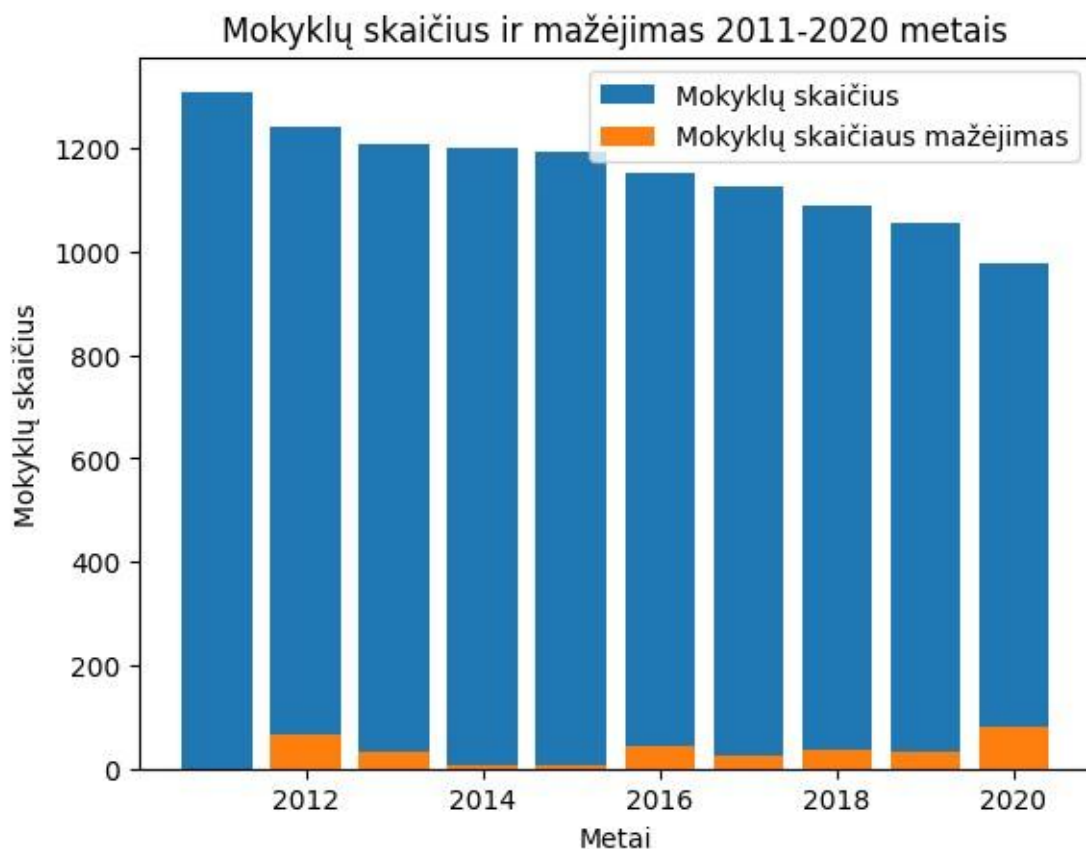
2 pavyzdys. Pakeisime programos kodą, kad duomenys būtų atvaizduojami stulpeline diagrama.

```
import matplotlib.pyplot as plt

x = [2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020]
y1 = [1309, 1242, 1208, 1200, 1193, 1151, 1125, 1089, 1056, 977]
y2 = [0, 67, 34, 8, 7, 42, 26, 36, 33, 79] plt.bar(x, y1,
label="Mokyklų skaičius") plt.bar(x, y2, label="Mokyklų skaičiaus
mažėjimas") plt.plot()

plt.xlabel("Metai") plt.ylabel("Mokyklų skaičius")
plt.title("Mokyklų skaičius ir mažėjimas 2011-2020 metais")
plt.legend() plt.show()
```

10 pav. . Python programos kodas stulpelinei diagramai



11 pav. Stulpelinė diagrama

Gabesniems mokiniams mokytojas gali pasiūlyti išnagrinėti ir išbandyti daugiau diagramų kūrimo galimybių, nurodydamas papildomus išteklius:

<https://colab.research.google.com/notebooks/charts.ipynb>

<https://matplotlib.org/stable/tutorials/pyplot.html>

Jei mokiniai suspėjo išnagrinėti pavyzdines programas, jiems galima skirti papildomas užduotis, pateikiant lenteles su duomenimis, arba pasiūlyti mokiniams diagramomis atvaizduoti pačių pasirinktus duomenis.

Užduočių pavyzdžiai savarankiškam darbui

1. Nubraižyti dviejų tiesių grafikų (žr. 15 lentelę).

15 lentelė. Duomenys dviejų tiesių grafikams nubraižyti

x	$y_1 = 1 + 2x$	$y_2 = 2 + x$
-10	-19	-8
-9	-17	-7
-8	-15	-6
-7	-13	-5
-6	-11	-4
-5	-9	-3
-4	-7	-2
-3	-5	-1
-2	-3	0
-1	-1	1
0	1	2
1	3	3
2	5	4
3	7	5
4	9	6
5	11	7
6	13	8
7	15	9
8	17	10

9	19	11
10	21	12

2. Nubraižyti keturių funkcijų grafikus.

16 lentelė. Duomenys keturių funkcijų grafikams nubraižyti

x	$y=2x^2$	$y=-2x^2$	$y=4x^2$	$y=-4x^2$
-10	200	-200	400	-400
-9	162	-162	324	-324
-8	128	-128	256	-256
-7	98	-98	196	-196
-6	72	-72	144	-144
-5	50	-50	100	-100
-4	32	-32	64	-64
-3	18	-18	36	-36
-2	8	-8	16	-16
-1	2	-2	4	-4
0	0	0	0	0
1	2	-2	4	-4
2	8	-8	16	-16
3	18	-18	36	-36
4	32	-32	64	-64
5	50	-50	100	-100
6	72	-72	144	-144
7	98	-98	196	-196
8	128	-128	256	-256
9	162	-162	324	-324
10	200	-200	400	-400

3 Stulpeline diagrama atvaizduoti pateiktus duomenis (nepamirškite mokiniams priminti, kad tekstiniai duomenys rašomi viengubose kabutėse).

17 lentelė. Duomenys stulpelinei diagramai atvaizduoti

Mokyklos tipas	Mokslo metai	
	2007–2008	2008–2009

Mokykla-darželis	110	104
Pradinė	101	95
Pagrindinė	562	530
Jaunimo	21	20
Vidurinė	387	357
Gimnazija	152	166
Suaugusiųjų	31	32
Vaikų socializacijos centras	4	6
Specialioji	67	65
Sanatorinė	9	10

3 ETAPAS ☺ (5 minutės). Darbų vertinimas, įsivertinimas, refleksija.

Mokinių prašoma užpildyti pamokos dalis atsakant į klausimus (klausimai gali būti įdėti į Google Forms, apklausa.lt, VMA Moodle ir kt. sistemas).

1. Parašykite po 3 sėkmingiausių ir 3 sunkiausių dalykus kuriant minčių žemėlapius.
2. Įvertinkite užduotį „Minčių žemėlapio kūrimas“ (pasirinkite tinkamą):
 - labai įdomi;
 - įdomi;
 - neturiu nuomonės; neįdomi; labai neįdomi.
3. Parašykite po 3 sėkmingiausių ir 3 sunkiausių dalykus kuriant diagramas Python programavimo kalba.
4. Įvertinkite užduotį „Diagramų kūrimas“ (pasirinkite tinkamą):
 - labai įdomi;
 - įdomi;
 - neturiu nuomonės; neįdomi;
 - labai neįdomi.

2. Parašykite, kur praktiškai panaudosite įgytas žinias ir įgūdžius.

Programavimo aplinka, jos pagrindiniai elementai (Renata Ališauskienė)

Pasiekimų sritis	Algoritmai ir programavimas. B(2.3)
Klasė	7-8 klasė
Tema	Programavimo aplinka, jos pagrindiniai elementai

Integruojami dalykai, pasiekimai	Lietuvių k., matematika, anglų k.
Kompetencijos	Pažinimo – susipažįsta su nauja programavimo aplinka. Skaitmeninė – skaitmeninių įrankių https://www.codeblocks.org/ taikymas. Subkompetencija skaitmeninis raštingumas – tobulina skaitmeninį raštingumą. Subkompetencija individualizuotas skaitmeninis mokymas – savarankiškai mokosi skaitmeninio įrankių taikymo. Komunikavimo – pagalba draugams, atliktų darbų pristatymai, pasidalinimas sprendimais.
Tikslas	Susipažinti su CodeBlocks aplinka.
Uždaviniai	Sukurs darbo katalogą ir programos failą. Pakeis programos pavadinimą, programą įrašys į darbo katalogą. Sukompiliuos ir įvykdys paprasčiausią programą. Išmoks taisyti programą. Išmoks išvesti duomenis į ekraną naudodami išvesties srautą cout.
Planuojamas rezultatas	Sekdami nurodymais: gebės sukurti darbo katalogą ir programos failą: gebės pakeisti programos pavadinimą, programą įrašydami į darbo katalogą: gebės sukompiliuoti ir įvykdyti paprasčiausią programą: mokės taisyti programą: mokės išvesti duomenis į ekraną naudodami išvesties srautą cout.
Specifinės priemonės / programinė įranga	Išmanioji lenta arba multimedija internetinė svetainė: https://www.codeblocks.org/
Mokymosi metodai	Naujos temos pristatymas Individualus darbas Įsivertinimas?
Mokinių atlikto darbo vertinimas ir įsivertinimas[4]	Slenkstinis – padedamas naudojasi programavimo kalbos konstrukcijomis , atkartoja mokytojo veiksmus programavimo aplinkoje. Patenkinamas – užrašo kelių eilučių algoritmus naudodamas programavimo konstrukcijas ir padedamas naudojasi programavimo aplinka. Pagrindinis – sprendamas problemas naudoja programavimo kalbos konstrukcijas ir aplinką. Aukštesnysis – randa ir išbando įvairias programavimo aplinkos funkcijas, eksperimentuoja su programavimo kalbos konstrukcijomis.
Žinios prieš	Mokėti naudotis kompiuterinėmis taikomosiomis programomis.
Galimybės taikyti spec. poreikių mokiniams	Pritaikyti pamokos dėstymą pagal spec. Mokinio ar mokinių gebėjimus.

<p>Patarimai kolegoms, kurie naudos parengtą medžiagą</p>	<ol style="list-style-type: none"> 1. Scenarijuje numatytas užduotis atlikite patys. 2. Skirtingų gebėjimų mokiniams adaptuokite scenarijų. 3. Galite praturtinti mokiniams teikiamus šaltinius įvairesniais. 4. Pamokos metu mokiniai dirba savarankiškai, mokytojas atlieka pagalbininko ir konsultanto vaidmenį.
---	---

1.ETAPAS (5 minutės) Mokytojas pristato temą:

Šiandien susipažinsite su CodeBlocks aplinka ir sukursite savo pirmąjį CodeBlocks. Lentoje

<https://www.codeblocks.org>

2. ETAPAS (35 minutės) Mokiniai pagal duodamus mokytojos nurodymus susipažins ir kurs savo pirmąjį CodeBlocks:

Norint pradėti dirbti, reikia, kad kompiuteryje būtų įdiegta CodeBlocks versija 1712



Code::Blocks

Code::Blocks

The free C/C++ and Fortran IDE.

Code::Blocks is a free C/C++ and Fortran IDE built to meet the most demanding needs of its users. It is designed to be very extensible and fully configurable.

Built around a plugin framework, Code::Blocks can be extended with plugins. Any kind of functionality can be added by installing/coding a plugin. For instance, event compiling and debugging functionality is provided by plugins!

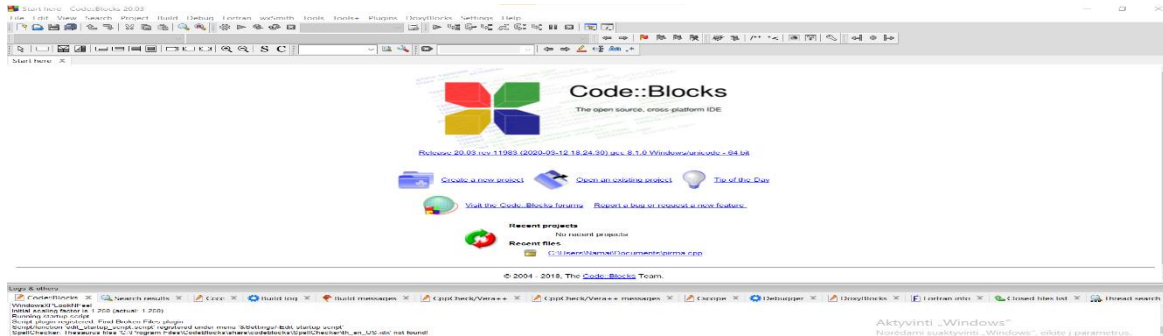


6 pav. Code Blocks

Įdiegus CodeBlocks dukart spragtelėkite darbalaukyje šaukinį:

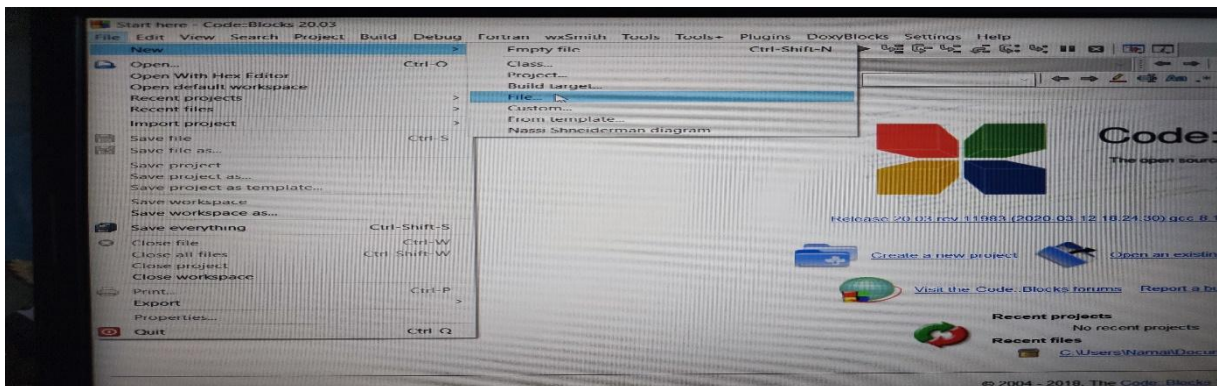


7 pav. Code Blocks šaukinys

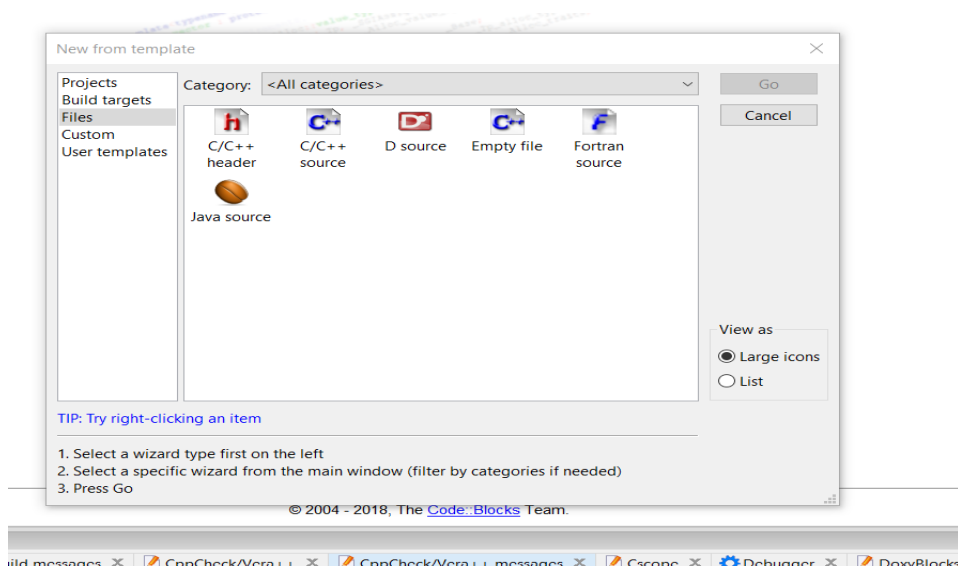


8 pav. Code Blocks atidarymas

Meniu juostoje file atsidarome new file:



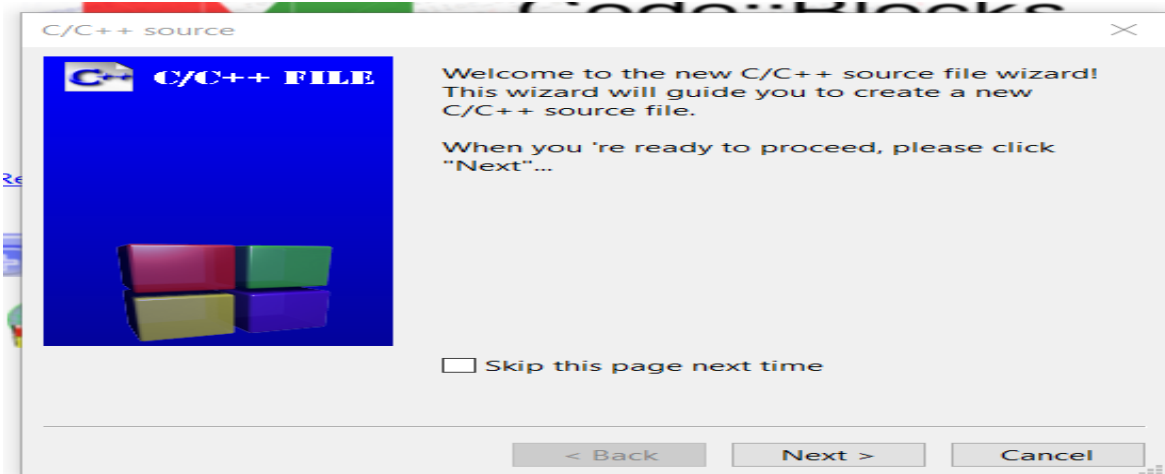
9 pav. Failo atidarymas
Atsivers



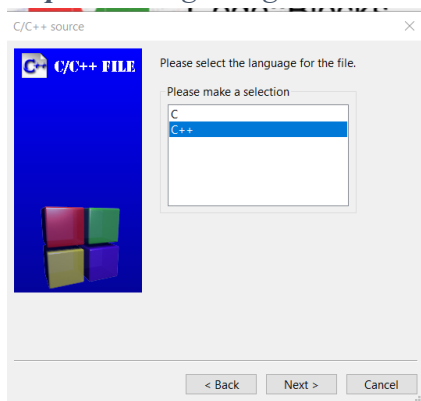
10 pav. Spausti c++

Spaudžiame mėlyną c++ ir mygtuką GO

Atsivėrusiame dialogo lange pasirinkite programavimo kalbą C++ ir spragtelėkite mygtuką Next.

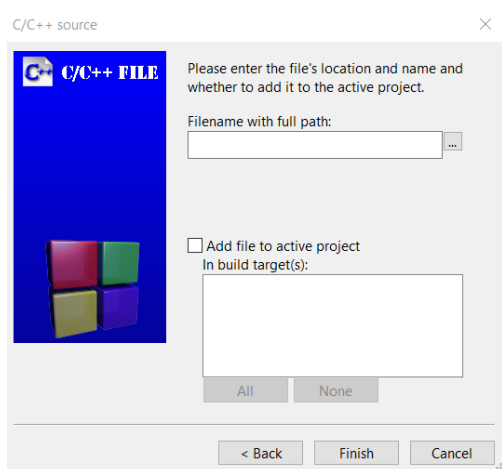


11 pav. Dialogo langas

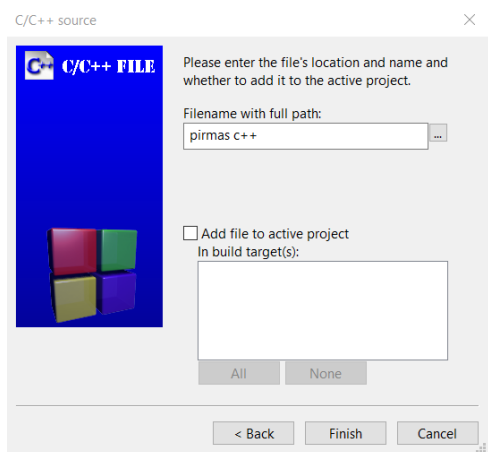


12 pav. Kalbos nurodymas

Pasirinkite kur įrašyti failą: spragtelėkite mygtuką , atsivėrusiame dialogo lange nurodykite katalogą pvz.: pirmas c++, o laukelyje File name įrašykite failo, kuriame bus sukurta programa vardą pirmas c++. Spragtelėkite mygtuką Save.

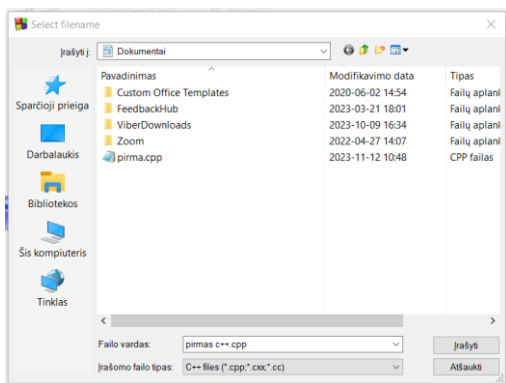


13 pav. Failo vardo užrašymas

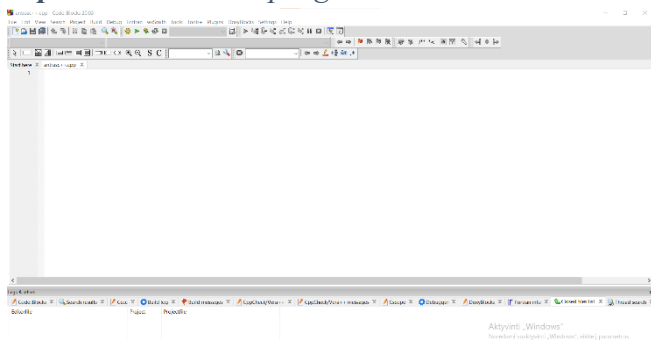


14 pav. Failo vardo išsaugojimas

Dialogo lange matysite visą programos failo adresą. Spragtelėjus mugtuką Finich, aplinkoje CodeBlocks atsivers kuriamos programos šablonas, kurį galėsite redaguoti CodeBlocks rengyklėje.



15 pav. Code Blocks programos šablonas



16 pav. Code Blocks programos langas

Kai mokiniai atsidaro savo pirmojo darbo langą mokytoja aiškina toliau:

Pristato c++ programos sintaksę:

Programos pradžioje surašytos instrukcijos parengiamajai doroklei (angl. preprocessor), kurios žymimos simboliu #. Įterpimo instrukcijomis include nurodoma, kokių failų tekstai turi būti įterpti instrukcijų pažymėtose vietose pirminio apdorojimo metu. Įterpiamų failų vardai rašomi tarp simbolių <>. Šablone įrašyti pirmajame darbe naudojami failai.

```

1 #include <iostream>
2 #include <string>
3 #include <math>
4 using namespace std;
5
6 int main()
7 {
8     int a = 5, b = 7;
9     cout << "a + b = " << a + b << endl;
10
11     return 0;
12 }

```

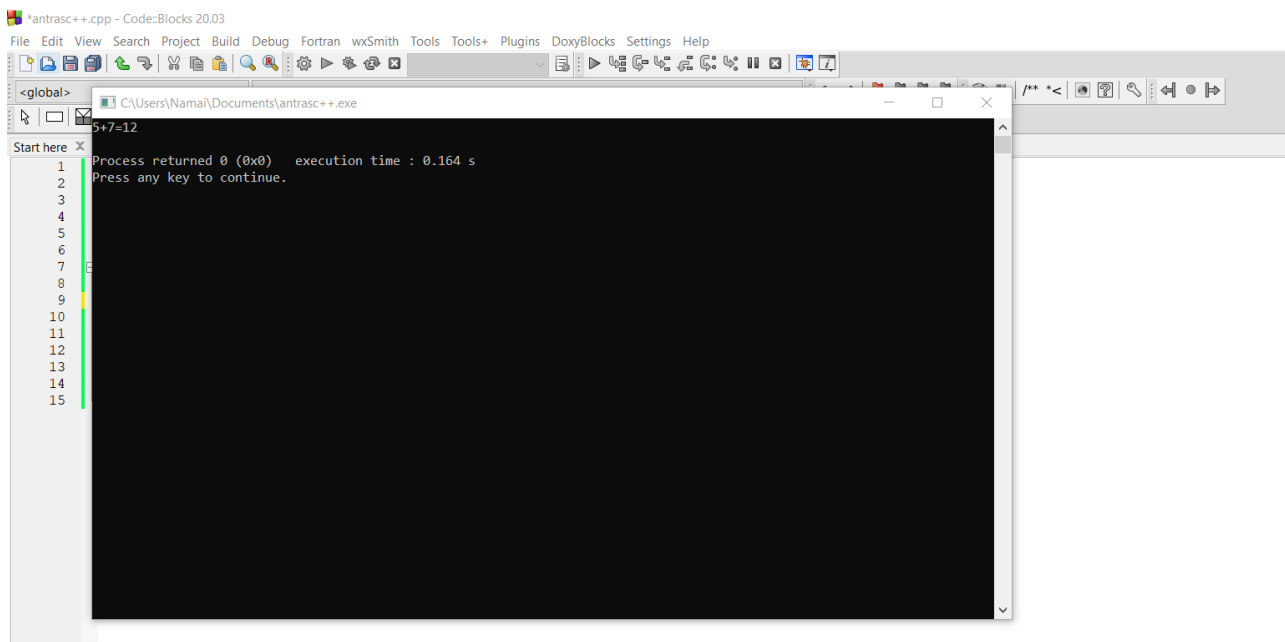
17 pav. Naudojamų failų įrašymas

Sakinys `using namespace std;` rašomas visada, jei programoje įterpiamas bent vienas antraštinis failas (pvz., `iostream`). Toliau rašoma programos pagrindinės funkcijos antraštė `int main ()`. Pagrindinės funkcijos kamienas (veiksmų sritis) pradedamas ženklų `{`, baigiamas – ženklų `}`.

Skaičiavimas `cout << << endl;` į ekraną išveda atsakymą 12. Sakinys `return 0;` nurodo programai baigti funkcijos `main()` darbą.

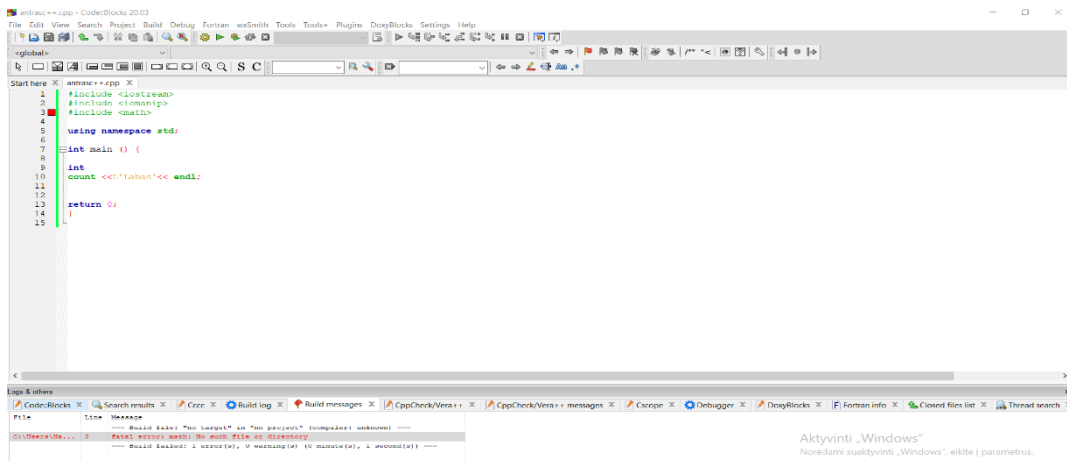
Mokiniai nusirašo mokytojos duotą programą. Išanalizuoja.

Programos kompiliavimas: Jeigu programoje nėra sintaksės klaidų, programos priemonių juostoje paspaudus krumpliaračio mygtuką, apatinėje darbo lango dalyje bus rodomas pranešimas apie sėkmingai sukompiliuotą programą. O paspaudus krumpliaračio mygtuką su žaliu trikampiu rodys suskaičiuotą rezultatą:



18 pav. Uždavinio rezultatas

Mokytoja tyčia gali padaryti klaidą, kad mokiniai suprastų ir pasimokytų kaip ją ištaisyti:



19 pav. Klaidos radimas ir taisymas

Darbą galima pabaigti spragtelėjus CodeBlocks lango užvėrimo mygtuką X.

3 DALIS (5 minutės) Refleksija:

1. Ką pirmiausia rekomenduojama sukurti prieš pradėdant rašyti programą?
2. Kaip paleisti CodeBlocks?
3. Kaip sukurti naują programą?
4. Kur įvedamas programos pavadinimas?
5. Kokius veiksmus reikia atlikti norint išsaugoti programą laikmenoje?
6. Kokie yra programos kompiliavimo būdai?
7. Kaip galima paleisti programą?
8. Kokią funkciją atlieka manipulatorius endl?

Veiksmų šakojimas, šakotieji algoritmai (Renata Ališauskienė)

Pasiekimų sritis	(B) Algoritmai ir programavimas. B 2.3
Klasė	7 kl. – 8 kl.
Tema	Veiksmų šakojimas - šakotieji algoritmai.
Integruojami dalykai, pasiekimai	Anglų kalba, lietuvių kalba, matematika.
Kompetencijos	Pažinimo – šakotųjų algoritmų pristatymas ir paaiškinimas. Skaitmeninė – skaitmeninių įrankių https://www.onlinegdb.com/# taikymas. Subkompetencija skaitmeninis raštingumas – tobulina skaitmeninį raštingumą. Subkompetencija individualizuotas skaitmeninis mokymas – savarankiškai mokosi skaitmeninio įrankių taikymo. Komunikavimo – pagalba draugams, atliktų darbų pristatymai, pasidalinimas sprendimais.
Tikslas	Spręsdami uždavinius mokyšis kurti paprastuosius šakotuosius algoritmus.
Uždaviniai	1. Susipažins su mokytojo pateikta informacine medžiaga. 2. Spręsdami uždavinius išmoks kurti šakotuosius algoritmus. 3. Apibendrins ir pristatys sukurtą algoritmą.
Planuojamas rezultatas	1. Gebės naudotis pateikta informacija; 2. Mokės kurti šakotuosius algoritmus 3. Pagilins atlikto darbo pristatymo įgūdžius.
Specifinės priemonės / programinė įranga	<u>Išmanioji lenta</u> <u>arba multimedija</u> <u>internetinė svetainė:</u> http://skaidres.puslapiai.lt/images/2_medziaga/%C5%A0akotieji_algoritmai.pdf https://www.onlinegdb.com/#

Mokymosi metodai	Naujos temos pristatymas Diskusijos Individualus darbas Sukurta algoritma pristato klasei. Išvertinimo metodas „Šviesoforas“.
Mokinių atlikto darbo vertinimas ir išvertinimas[4]	Slenkstinis – Padedamas naudojami programavimo kalbos konstrukcijomis, atkartoja mokytojo veiksmus programavimo aplinkoje. Patenkiamas - Užrašo kelių eilučių algoritmus naudodamas programavimo konstrukcijas ir padedamas naudojami programavimo aplinka. Pagrindinis - Spręsdamas uždavinius naudoja programavimo kalbos konstrukcijas ir aplinką. Aukštesnysis – Sprendžia sudėtingesnius uždavinius su sąlyginu sakiniu ir išbando įvairias programavimo aplinkos funkcijas, eksperimentuoja su programavimo kalbos konstrukcijomis.
Žinios prieš	Mokėti atlikti sudėti, atimti, dauginti, dalinti su sveikaisiais ir realiaisiais skaičiais.
Galimybės taikyti spec. poreikių mokiniams	Pateikiamas vienas šaltinis iš kurio reikia nukopijuoti jau parašytą uždavinio sprendimą ir įklijuoti į word lapą.
Patarimai kolegoms, kurie naudos parengtą medžiagą	1. Scenarijuje numatytas užduotis atlikite patys. 2. Skirtingų gebėjimų mokiniams adaptuokite scenarijų. 3. Galite praturtinti mokiniams teikiamus šaltinius įvairesniais. 4. Pamokos metu mokiniai dirba savarankiškai, mokytojas atlieka pagalbininko ir konsultanto vaidmenį.

1 ETAPAS (15 minučių). Pamokos temos pristatymas. Mokytojo rodomos skaidrės.

Mokytojas pateikia skaidres. Kartu su mokiniais jas aptaria.

1. Mokytojas primena kaip atliekami veiksmai su sveikaisiais ir realiaisiais skaičiais:

```

#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;
int main()
{
    // Kintamųjų aprašymas: int - sveikųjų skaičių tipas, po tipo seka vardai
    int pirmas, antras; //Pradiniai duomenys
    int suma, skirtumas, sandauga, sv_dalis, liekana; //Rezultatas
    double dalmuo; //Realiojo tipo
    cout << "Įveskite pirmą ir antrą skaičius: " << endl;
    cin >> pirmas >> antras;
    suma = pirmas + antras; //Priskyrimo sakiny
    skirtumas = pirmas - antras;
    sandauga = pirmas * antras;
    sv_dalis = pirmas / antras;
    liekana = pirmas % antras;
    dalmuo = (double) pirmas / antras;
    cout << pirmas << " + " << antras << " = " << suma << endl;
    cout << pirmas << " - " << antras << " = " << skirtumas << endl;
    cout << pirmas << " * " << antras << " = " << sandauga << endl;
    cout << pirmas << " / " << antras << " = " << sv_dalis << endl;
    cout << pirmas << " % " << antras << " = " << liekana << endl;
    cout << pirmas << " / " << antras << " = " << dalmuo << endl;
    return 0;
}

```

2. Mokytojas pateikia programą palyginančius du sveikuosius skirtingus skaičius:

```

//Programa palyginanti du sveikuosius skirtingus skaičius
#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;
int main()
{
    int pirmas, antras;
    cout << "Įveskite du sveikuosius skaičius: " << endl;
    cin >> pirmas >> antras;
    // if (pirmas > antras) cout << pirmas << " > " << antras << endl;
    // else cout << pirmas << " < " << antras << endl;
    if (pirmas < antras) cout << pirmas << " < " << antras << endl;
    else cout << pirmas << " > " << antras << endl;
}

```

```

return 0;
}

```

3. Pateikiama sudėtingesnė programa, kai vienu atveju reiškinio reikšmę galime suskaičiuoti, o kitu atveju turime dalybą iš 0.

// Programa, skaičiuojanti reiškinio $y = (x + 1) / x$ reikšmę. x ir y - realieji skaičiai.

```

#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;
int main()
{
    double x; // Pradinis duomuo
    double y; // Rezultatas
    cout << "Įveskite x reikšmę: " << endl;
    cin >> x;
    if (x != 0) {
        y = (x + 1) / x;
        cout << "Kai x = " << x << " y = " << y << endl;
    }
    else cout << "Vardiklis lygus nuliui." << endl;
    return 0;
}

```

```

using namespace std;
int main()
{
    double x; // Pradinis duomuo
    double y; // Rezultatas
    cout << "Įveskite x reikšmę: " << endl;
    cin >> x;
    // if (x != 0) {
    //     y = (x + 1) / x;
    //     cout << "Kai x = " << x << " y = " << y << endl;
    // }
    // else cout << "Vardiklis lygus nuliui." << endl;
    if (x == 0) cout << "Vardiklis lygus nuliui." << endl;
    else {
        y = (x + 1) / x;
        cout << "Kai x = " << x << " y = " << y << endl;
    }
    return 0;
}

```

2 ETAPAS (20 minučių). Savarankiškas darbas.

Užduotis – savarankiškai atlieka duotus uždavinius:

1. **Sprendimas.** Urtė sprendžia informatikos uždavinį per b minutes, o Ugnė – per a minučių. Parašykite programą, kuri nustato, kuri iš merginų užduotį išsprendė greičiau. Visi duomenys yra sveikieji skaičiai ir abi merginos užduotį atliko per skirtingą laiką.

Patikrinkite programą: jei $b = 5$, $a = 10$, išveskite pranešimą į ekraną: „Išspręs Urtė uždavinį greičiau“.¹

Jei $b = 10$, $a = 5$, išveskite pranešimą į ekraną: „Ugnė uždavinį išsprendė greičiau“.

```
//Programa, įveskite Urtės laiką (b)
```

```
#include <iostream>
```

```
#include <iomanip>
```

```
#include <cmath>
```

```
using namespace std;
```

```
int main() {
```

```
    int b, a;
```

```
    cout << "Įveskite Urtės laiką (b): ";
```

```
    cin >> b;
```

```
    cout << "Įveskite Ugnės laiką (a): ";
```

```
    cin >> a;
```

```
    if (b < a) {
```

```
        cout << "Išspręs Urtė uždavinį greičiau." << endl;
```

```
    } else if (a < b) {
```

```
        cout << "Ugnė uždavinį išsprendė greičiau." << endl;
```

```
    } else {
```

```
        cout << "Merginos išsprendė uždavinį per tą patį laiką." << endl;
```

```
    }
```

```
    return 0;
```

¹ Laisvai platinamas uždavinynas sukurtas doc. R. Burbaitės

```
}
```

2. **Žuvyčių tyrimas.** Turime mažą žuvytę sveriančią m miligramų. Mokslininkai, tyriantys mažas žuvytes, nusprendė stebėti n žuvyčių. Jei bendra stebimų žuvyčių masė yra ne mažesnė kaip vienas kilogramas, išveskite pranešimą į ekraną „Žuvų užtenka“, jei mažesnė – „Žuvų per mažai“.

Patikrinkime programą: jei $m = 3$, $n = 1000000$, išveskite pranešimą į ekraną: „Žuvų užtenka“. Jei $m = 3$, $n = 1000$, išveskite pranešimą į ekraną: „Žuvų per mažai“.²

```
//Programa, įveskite žuvytės svorį m miligramais ir stebėtojų skaičių n
#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;
int main()
{
    int m, n;

    // Įveskite žuvytės svorį m ir stebėtojų skaičių n
    cout << "Įveskite žuvytės svorį (miligramais): ";
    cin >> m;
    cout << "Įveskite stebėtojų skaičių: ";
    cin >> n;
    // Tikriname sąlygą: jei žuvytė sveria bent 1000 gramų, tai pakanka stebėtojų
    if (m >= 1000)
    {
        cout << "Žuvų užtenka" << endl;
    }
    else
    {
        cout << "Žuvų per mažai" << endl;
    }
    return 0;
}
```

3. **Leopardo greitis.** Leopardas greičiausias gyvūnas per s sekundžių nubėga c kilometrų. Sukurkite programą, kuri palygintų leopardo greitį su žirafos, kuri bėga v km/s greičiu. Jei žirafa greitesnė, išveskite pranešimą į ekraną: „Žirafa greitesnė už leopardą“, jei greitesnis yra leopardas ekrane turi būti rodoma: „Leopardas greitesnis už žirafą“.

Patikrinkite programą: jei $s = 2$, $c = 20$, $v = 4$, išveskite pranešimą į ekraną: „Žirafa greitesnė už leopardą“.³

```
//Programa, įveskite laiką s (sekundėmis), atstumą c (kilometrais) ir žirafos greitį v (km/s)
#include <iostream>
```

² Laisvai platinamas uždavinynas sukurtas doc. R. Burbaitės

³ Laisvai platinamas uždavinynas sukurtas doc. R. Burbaitės

```

#include <iomanip>
#include <cmath>
using namespace std;
int main()
{
    double s, c, v;
    // Įveskite laiką s (sekundėmis), atstumą c (kilometrais) ir žirafos greitį v (km/s)
    cout << "Įveskite laiką s (sekundėmis): ";
    cin >> s;
    cout << "Įveskite atstumą c (kilometrais): ";
    cin >> c;
    cout << "Įveskite žirafos greitį v (km/s): ";
    cin >> v;
    // Konvertuojame žirafos greitį į km/h (1 km/s = 3600 km/h)
    double zirafos_greitis_kmh = v * 3600;
    // Skaičiuojame leopardui reikalingą laiką bėgant atstumą c greičiu zirafos
    double leopardas_laikas = c / zirafos_greitis_kmh;
    // Tikriname sąlygą: jei leopardas bėga ilgiau nei s sekundžių, tai spausdiname atitinkamą
pranešimą
    cout << "Leopardas greitesnis už žirafą" << endl;
    }
    else
    {
    cout << "Žirafa greitesnė už leopardą" << endl;
    }
    return 0;
}

```

4. **Bauda.** Vairuotojui už greičio viršijimą atėjo bauda 1000 eurų. Kaltininkas 1 euro monetomis sumokėjo baudą. Ar tilps pinigai į priekabą, kuri gali vežti t tonų? Jei monetos telpa ekrane turi būti rodomas pranešimas „Telpa“, jei netelpa: „Netelpa“. Viena 1 euro moneta sveria 7,5 gramo.

Patikrinkite programą: jei $k = 15000$, $t = 1$, išveskite pranešimą į ekraną: „Netelpa“.

Jei $k = 15000$, $t = 1.5$, išveskite pranešimą į ekraną: „Telpa“.⁴

// Programa, įveskite iš viso money skaičius(eurais)

```
#include <iostream>
```

```
#include <iomanip>
```

```
#include <cmath>
```

⁴ Laisvai platinamas uždavinynas sukurtas doc. R. Burbaitės

```

using namespace std;

int main() {
    int k; // Iš viso monetų skaičius (eurais)

    double t; // T tonų priekabos talpa

    const double gram_per_euro = 7.5; // Gramų skaičius vienai 1 euro monetai

    cout << "Įveskite monetų skaičių (eurais): ";

    cin >> k;

    cout << "Įveskite priekabos talpą tonomis: ";

    cin >> t;

    double total_weight = k * gram_per_euro / 1000.0; // Bendras monetų svoris tonomis

    if (total_weight <= t) {
        cout << "Telpa" << endl;
    } else {
        cout << "Netelpa" << endl;
    }

    return 0;
}

```

5. Močiutės daržas. Močiutė nusprendė virve pasižymėti stačiakampio plotą, kuriame sodins agurkus. Virvės ilgis yra a metrų (sveikais skaičiais). Kokį didžiausią plotą močiutė galės pažymėti virvės ilgio s pagal plotą a ? Rezultatas pateikiamas sveikuoju skaičiumi (galimas virvės galas gali likti nepanaudotas).

Patikrinkite:programą: jei $a = 12$, išveskite pranešimą į ekraną: $s = 36$. Jei $a = 23$, išveskite pranešimą į ekraną: $s = 121$.⁵

```

//Programa,      įveskite      virvės      ilgį      a      metrais
#include          <iostream>
#include          <iomanip>
#include          <cmath>
using            namespace    std;
int              main()      {
    int          a;

```

⁵ Laisvai platinamas uždavinynas sukurtas doc. R. Burbaitės

```

//      Įveskite      virvės      ilgį      a      metrais
cout      <<      "Įveskite      virvės      ilgį      (metrais):      ";
cin      >>      a;
//      Skaičiuojame      stačiakampio      plotą,      kurį      galima      pažymėti      virve
//      Plotas      lygus      a/2      *      a/2,      kadangi      stačiakampio      pusrutulio      ilgis      yra      lygus      a/2
int      s      =      (a      /      2)      *      (a      /      2);
//      Spausdiname      rezultata
cout      <<      "s      =      "      <<      s      <<      endl;
return      0;
}

```

Mokytojas padeda mokiniams, kuriems reikia pagalbos.

Papildomos užduotys – 1. Blusa. Blusos nušoka b cm. Sukurkite programą, nustatančią ar nušokusi s šuoliais, blusa nušoks m metrų atstumą. Duomenys pateikti sveikaisiais skaičiais. Patikrinkite programą: jei $b = 40$, $s = 4$, $m = 4$, išveskite pranešimą į ekraną: „Blusa 4 šuoliais 4 metrų nenušoks“. Jei $b = 40$, $s = 4$, $m = 1$, išveskite pranešimą į ekraną: „Blusas 4 šuoliais 1 metrą nušoks“.⁶

```

//Programa,      įveskite      blusos      šuolio      ilgį      b      (cm)
#include      <iostream>
#include      <iomanip>
#include      <cmath>
using      namespace      std;
int      main()
{
    int      b,      s,      m;
    //      Įveskite      blusos      šuolio      ilgį      b      (cm)
    cout      <<      "Įveskite      blusos      šuolio      ilgį      (cm):      ";
    cin      >>      b;
    //      Įveskite      šuolių      skaičių      s
    cout      <<      "Įveskite      šuolių      skaičių:      ";
    cin      >>      s;
    //      Įveskite      norimą      nušokti      atstumą      m      (metrais)
    cout      <<      "Įveskite      norimą      nušokti      atstumą      (metrais):      ";
    cin      >>      m;
    //      Konvertuojame      blusos      šuolio      ilgį      iš      cm      į      metrus
    double      b_m      =      static_cast<double>(b)      /      100.0;
    //      Skaičiuojame      bendrą      nušoktą      atstumą
    double      nušoktas_atstumas      =      b_m      *      s;
    //      Tikriname      sąlygą:      ar      nušoktas      atstumas      atitinka      norimą      atstumą      m
    if      (nušoktas_atstumas      >=      m)      {
        cout      <<      "Blusa      "      <<      s      <<      "      šuoliais      "      <<      m      <<      "      metro      atstumą      nušoks."      <<      endl;
    }
    else      {

```

⁶ Laisvai platinamas uždavinynas sukurtas doc. R. Burbaitės

```

    cout << "Blusa " << s << " šuoliais " << m << " metrų atstumo nenušoks." << endl;
}
return 0;
}

```

2. **Vakarienė.** Tėtė virė vakarienę, kai laikrodis rodė v valandų ir m minučių. Jis vakarienę paruošė per p minučių. Sūnūs iš treniruotės grįš, kai bus vv valandų ir vm minučių. Parašykite programą, kuri nustatytų, ar tėtis spės paruošti vakarienę, kol sūnūs grįš iš treniruotės. Visi duomenys yra sveikieji skaičiai.

Patikrinkite programą: jei $v = 16$, $m = 50$, $p = 60$, $vv = 18$, $vm = 50$, išveskite pranešimą į ekraną: „Tėtis vakarienę pagaminti spės“.

Jei $v = 16$, $m = 50$, $p = 30$, $vv = 17$, $vm = 10$, išveskite pranešimą į ekraną: „Tėtis vakarienės pagaminti nespės“.⁷

```

//Programa, įveskite kurią valandą tėtis pradėjo gaminti vakarienę (v)
#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;
int main() {
    int v, m, p, vv, vm;
    // Įveskite, kiek minučių tėtis pradėjo gaminti vakarienę (m)
    cout << "Įveskite minutes, kai tėtis pradėjo gaminti vakarienę: ";
    cin >> m;
    // Įveskite, kiek minučių reikia tėčiui pagaminti vakarienę (p)
    cout << "Įveskite, kiek minučių reikia pagaminti vakarienę: ";
    cin >> p;
    // Įveskite, kiek valandų sūnūs grįš iš treniruotės (vv)
    cout << "Įveskite valandas, kai sūnūs grįš iš treniruotės: ";
    cin >> vv;
    // Įveskite, kiek minučių sūnūs grįš iš treniruotės (vm)
    cout << "Įveskite minutes, kai sūnūs grįš iš treniruotės: ";
    cin >> vm;
    // Skaičiuojame laiko skirtumą tarp vakarienės gamybos ir sūnų grįžimo
    int skirtumas = (vv - v) * 60 + (vm - m);
    // Tikriname sąlygą: ar tėtis spės pagaminti vakarienę per duotą laiką
    if (skirtumas >= p) {
        cout << "Tėtis vakarienę pagaminti spės." << endl;
    }
}

```

⁷ Laisvai platinamas uždavinynas sukurtas doc. R. Burbaitės

```

    }
    cout << "Tėtis vakarienės pagaminti nespės." << endl;
}
return 0;
}

```

3. **Planšetės.** Su p procentų (p – sveikasis skaičius) nuolaida mobiliųjų paslaugų teikėjas parduoda planšetes. Aurėja turi a eurų (realusis skaičius) ir pirks planšetę, kuri be nuolaidos kainuoja t eurų (realusis skaičius). Parašykite programą, kuri apskaičiuotų, kiek kainuos planšetė su nuolaida tn ir ar Aurėjai užteks pinigų planšetei pirkti.

Patikrinkite programą: jei $p = 10$, $a = 200.00$, $t = 300.00$, išveskite pranešimą į ekraną: „Planšetė su

nuolaida kainuos 20.00 eurų. Aurėjai pinigų neužteks pirkti“.

Jei $p = 10$, $a = 400.00$, $t = 300.00$, išveskite pranešimą į ekraną: „Planšetė kainuos su nuolaida 270.00 eurų. Aurėjai pinigų užteks planšetei pirkti“.⁸

```

//Programa, įveskite nuolaidos procentą p
#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;
int main()
{
    double p, a, t;
    // Įveskite nuolaidos procentą p
    cout << "Įveskite nuolaidos procentą (%): ";
    cin >> p;
    // Įveskite Aurėjos sutaupytus pinigus a eurai
    cout << "Įveskite Aurėjos sutaupytus pinigus (eurai): ";
    cin >> a;
    // Įveskite planšetės kainą be nuolaidos t eurai
    cout << "Įveskite planšetės kainą be nuolaidos (eurai): ";
    cin >> t;
    // Skaičiuojame kainą su nuolaida
    double nuolaida = (p / 100.0) * t;
    double kaina_su_nuolaida = t - nuolaida;
    // Tikriname, ar Aurėjai užtenka pinigų planšetei įsigyti
    if (a >= kaina_su_nuolaida)
        cout << "Planšetė su nuolaida kainuos " << kaina_su_nuolaida << " eurų. Aurėjai pinigų

```

⁸ Laisvai platinamas uždavinynas sukurtas doc. R. Burbaitės

```

neužteks                pirkti."                <<                endl;
    }
    else                {
        cout << "Planšetė su nuolaida kainuos " << kaina_su_nuolaida << " eurų. Aurėjai pinigų
užteks                planšetei                pirkti."                <<                endl;
    }
    return                0;
}

```

3 ETAPAS (10 minučių). Darbų vertinimas, įsivertinimas, refleksija.

Tie mokiniai, kurie nori pristato savo pavykusius sprendimus. Aptariame kam nepavyko, kodėl, stipresnieji mokiniai pataiso nepavykusius darbus. Mokiniai įsivertina „Šviesoforo“ metodu.

Veiksmų kartojimas – ciklai (Renata Ališauskienė)

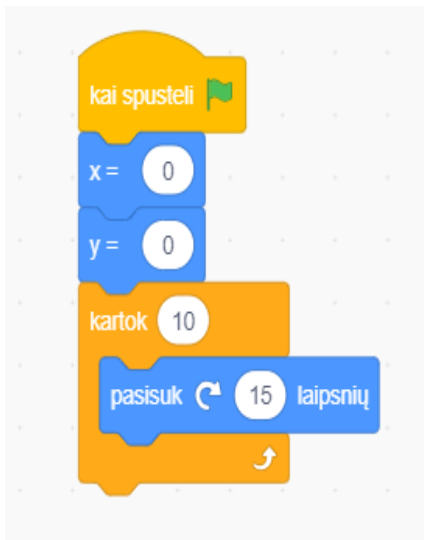
Pasiekimų sritis	(B) Algoritmai ir programavimas. B 2.3
Klasė	7kl. – 8kl.
Tema	Veiksmų kartojimas – ciklai.
Integruojami dalykai, pasiekimai	Anglų k., lietuvių k., matematika
Kompetencijos	Pažinimo – veiksmų kartojimas- ciklų pristatymas ir paaiškinimas. Skaitmeninė – skaitmeninių įrankių https://www.onlinegdb.com/# taikymas. Skaitmeninis raštingumas subkompetencija – tobulina skaitmeninį raštingumą. Individualizuotas skaitmeninis mokymas subkompetencija – savarankiškai mokosi skaitmeninio įrankių taikymo. Komunikavimo – pagalba draugams, atliktų darbų pristatymai, pasidalinimas sprendimais.
Tikslas	Susipažins su veiksmų kartojimo – ciklų uždaviniais ir jų sprendimais.

Uždaviniai	<p>1. Remdamiesi mokytojo pateikta informacine medžiaga susipažins su veiksmų kartojimu – ciklišku.</p> <p>2. Atlikdami https://www.onlinegdb.com/# aplinkoje pratybas, kurs reikiamas komandas.</p> <p>3. Apibendrins įgytą patirtį, atlikdami refleksiją elektroninio pašto ir Š. De Bono kepurėlių pagalba.</p>
Planuojamas rezultatas	<p>1. Gebės naudotis mokytojo pateikta informacija.</p> <p>2. Išmoks spręsti ciklinius uždavinius: žinomo ir nežinomo skaičiaus.</p> <p>3. Apibendrins įgytą patirtį.</p>
Specifinės priemonės / programinė įranga	<p><u>Išmanioji lenta</u> <u>arba multimedija</u> <u>internetinės svetainės:</u> https://www.onlinegdb.com/# https://scratch.mit.edu/</p>
Mokymosi metodai	<p>Naujos temos pristatymas panaudojant Š. De Bono kepurėles.</p> <p>Diskusijos.</p> <p>Individualus darbas panaudojant Š. De Bono kepurėles.</p> <p>Įsivertinimo metodas – Š. De Bono kepurėlių ir elektroninis paštas.</p>
Mokinių atlikto darbo vertinimas ir įsivertinimas[4]	<p>Slenkstinis – Padedamas naudojami programavimo kalbos konstrukcijomis, atkartoja mokytojo veiksmus programavimo aplinkoje.</p> <p>Patenkinamas - Užrašo kelių eilučių algoritmus naudodamas programavimo konstrukcijas ir padedamas naudojami programavimo aplinka.</p> <p>Pagrindinis - Spręsdamas uždavinius naudoja programavimo kalbos konstrukcijas ir aplinką.</p> <p>Aukštesnysis – Sprendžia sudėtingesnius uždavinius su sąlyginiu sakiniu ir išbando įvairias programavimo aplinkos funkcijas, eksperimentuoja su programavimo kalbos konstrukcijomis.</p>
Žinios prieš	<p>Mokiniai turi žinoti tiesinius ir šakotuosius algoritmus ir turi būti susipažinę su E. De Bono kepurėlių spalvų reikšmėmis.</p>

Galimybės taikyti spec. poreikių mokiniams	Pateikiamas vienas šaltinis iš kurio reikia nukopijuoti jau parašytą uždavinio sprendimą ir įklijuoti į word lapą.
Patarimai kolegoms, kurie naudos parengtą medžiagą	<ol style="list-style-type: none"> 1. Scenarijuje numatytas užduotis atlikite patys. 2. Skirtingų gebėjimų mokiniams adaptuokite scenarijų. 3. Galite praturtinti mokiniams teikiamus šaltinius įvairesniais. 4. Pamokos metu mokiniai dirba savarankiškai, mokytojas atlieka pagalbininko ir konsultanto vaidmenį.

1 ETAPAS (3 min.) Pamokos temos pristatymas:

1. Mokytojas su <https://scratch.mit.edu/> primena kas yra kartojimas:



20 pav. Kartojimas

Kai mokytoja pristato pamokos temą, supažindina su ciklo sakinio sintakse, žinomo kartojimo skaičiaus ciklą ir nežinomo kartojimo skaičiaus ciklą ji dėvi baltą kepurę ar lentoje pakabina



21 pav. Žinių kepurė

2 ETAPAS 1 DALIS (17 minučių). Mokytojos pristatymas ir mokinių savarankiškas darbas.

Supažindina su žinomo kartojimų skaičiaus ciklo sakinio sintaksę:

Kai kartojamas vienas veiksmas:

for (<kintamojo vardas> = <pradinė reikšmė>; <kintamojo vardas> <= <galutinė reikšmė>; kintamojo reikšmė didinama vienetu)

veiksmas, kuris kartojamas;

Kai kartojami keli veiksmai:

for (<kintamojo vardas> = <pradinė reikšmė>; <kintamojo vardas> <= <galutinė reikšmė>; kintamojo reikšmė didinama vienetu)

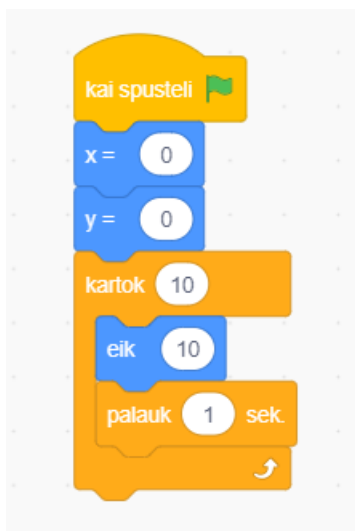
{

veiksmai, kurie kartojami;

}

1. Užduotis – žinomo kartojimų skaičiaus ciklas:

Mokytoja su scratch programa parodo žinomo kartojimo skaičiaus ciklą:



22 pav. Kartojimo skaičiaus ciklas

Ir <https://www.onlinegdb.com/#> aplinkoje paaiškina, žinomo kartojimų skaičiaus ciklas rašomas tuomet, kai žinome, kiek kartų reikės kartoti vieną ar kelis veiksmus.

```
//Išspausdinti skaičiaus 5 daugybos lentelę nuo 1 iki 10
#include <bits/stdc++.h>
using namespace std;
int main(){
    for (int i = 1; i <= 10; i++)
        cout << i << " * " << 5 << " = " << i * 5 << endl;
    return 0;
}
```

Detalيزuoja paaiškinimą:

Mokiniai sprendžia:

1.1. Pakeisti pavyzdžio programą, kad ji išvestų į ekraną skaičiaus 5 daugybos lentelę nuo 10 iki 20.

```
#include <bits/stdc++.h>
using namespace std;
int main(){
    for (int i = 10; i <= 20; i++)
        cout << i << " * " << 5 << " = " << i * 5 << endl;
    return 0;
}
```

1.2. Pakeisti pavyzdžio programą, kad ji išspausdintų skaičiaus 7 daugybos lentelę nuo 1 iki 20.

```
#include <bits/stdc++.h>
using namespace std;
int main(){
    for (int i = 1; i <= 20; i++)
        cout << i << " * " << 7 << " = " << i * 7 << endl;
    return 0;
}
```

1.3. Kai kartojami keli veiksmai: išsinagrinėkime tokį pavyzdį:

Robotas iš lego juda v cm/s pastoviu greičiu (sveikasis skaičius). Parašykite programą, kuri apskaičiuoja, kokį atstumą s nuvažiavo robotas nuo pradinės padėties iki dešimtosios sekundės, naudojant formulę $s = v * t$, kur v yra greitis, t – laikas. Jei skaičiuotume popieriuje, tuomet atliktume

tokius veiksmus (pvz., kai $v = 2$):

$s_0 = 2 * 0 = 0$; $s_1 = 2 * 1 = 2$; $s_2 = 2 * 2 = 4$; $s_3 = 2 * 3 = 6$; $s_4 = 2 * 4 = 8$; $s_5 = 2 * 5 = 10$;
 $s_6 = 12$; $s_7 = 2 * 7 = 14$; $s_8 = 2 * 8 = 16$; $s_9 = 2 * 9 = 18$; $s_{10} = 2 * 10 = 20$.⁹

Pastebime, kad roboto nuvažiuotas atstumas skaičiuojamas vykdant tą pačią operaciją kiekvieną kartą, kai kinta laiko reikšmė. Programą išnagrinėkite:

```

//Roboto judėjimas
#include <bits/stdc++.h>
using namespace std;
int main() {
    int v; // Roboto greitis
    int s; // Roboto nuvažiuotas kelias
    cout << "Kokiu greičiu juda robotas? "; cin >> v;
    for (int t = 0; t <= 10; t++)
    {
        s = v * t;
        cout << t << " " << s << endl;
    }
    return 0;
}

```

Įvykdę programą ir įvedę greičio reikšmę, ekrane matysite:

```

Kokiu greičiu juda robotas? 2
0 0
1 2
2 4
3 6
4 8
5 10
6 12
7 14
8 16
9 18
10 20

```

23 pav. Rezultatai 1

1.4. Papildykime programą taip, kad ji skaičiuotų ir ekrane parodytų dviejų robotų nuvažiuotus atstumus per tą patį laiką, kurių greičiai gali skirtis.

```

//Roboto judėjimas
#include <bits/stdc++.h>
using namespace std;
int main() {
    int v1, v2; // Robotų greičiai

```

⁹ Laisvai platinamas uždavinynas sukurtas doc. R. Burbaitės

```

int      s1,      s2;      //      Robotų      nuvažiuoti      keliai
cout    <<      "Kokiu      greičiu      juda      pirmasis      robotas? ";      cin    >>      v1;
cout    <<      "Kokiu      greičiu      juda      antrasis      robotas? ";      cin    >>      v2;
for      (int      t      =      0;      t      <=      10;      t++)
{
    s1      =      v1      *      t;
    s2      =      v2      *      t;
    cout    <<      t      <<      " " <<      s1      <<      " " <<      s2      <<      endl;
}
return      0;
}

```

```

Kokiu greiciu juda pirmasis robotas? 2
Kokiu greiciu juda antrasis robotas? 5
0 0 0
1 2 5
2 4 10
3 6 15
4 8 20
5 10 25
6 12 30
7 14 35
8 16 40
9 18 45
10 20 50

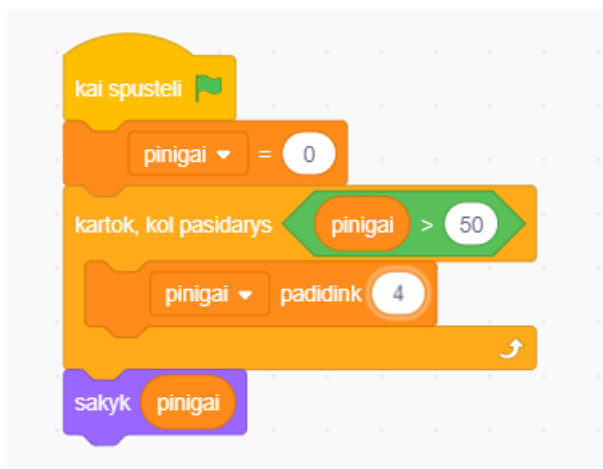
```

24 pav. Rezultatai 2

2 ETAPAS 2 DALIS (17 minučių). Mokytojos pristatymas ir mokinių savarankiškas darbas.

1. Užduotis – nežinomo kartojimų skaičiaus ciklas:

Mokytoja su scratch aplinka parodo nežinomo kartojimų skaičiaus ciklą:



25 pav. Nežinomo kartojimo skaičių ciklas

Ir <https://www.onlinegdb.com/#> aplinkoje paaiškina, kaip rašomas nežinomo kartojimų skaičiaus ciklas:

2.1. Pinigų skaičiavimas:

```
//Pinigų skaičiavimas
#include <bits/stdc++.h>
using namespace std;
int main() {
    int pinigai = 0;
    while (pinigai < 50){
        pinigai = pinigai + 4;
    }
    cout << pinigai << endl;
    return 0;
}
```

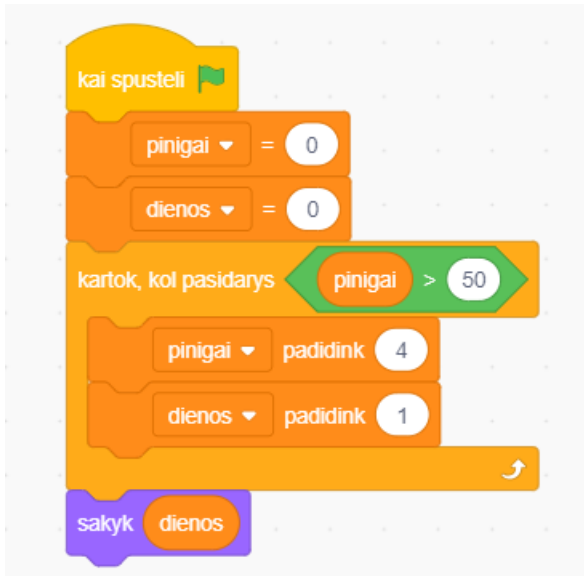
Jei reikia mokytoja dar detalizuoja paaiškinimą:

```
pinigai = 0, turim kartoti veiksmus tol, kol pinigai pasidarys > 50 kiekvieną kartą pinigus padidindami 4
Sąlyga pinigai > 50      pinigai + 4
0 > 50 - ne              pinigai = 0 + 4 = 4
4 > 50 - ne              pinigai = 4 + 4 = 8
8 > 50 - ne              pinigai = 8 + 4 = 12
12 > 50 - ne            pinigai = 12 + 4 = 16
16 > 50 - ne            pinigai = 16 + 4 = 20
20 > 50 - ne            pinigai = 20 + 4 = 24
24 > 50 - ne            pinigai = 24 + 4 = 28
28 > 50 - ne            pinigai = 28 + 4 = 32
32 > 50 - ne            pinigai = 32 + 4 = 36
36 > 50 - ne            pinigai = 36 + 4 = 40
40 > 50 - ne            pinigai = 40 + 4 = 44
44 > 50 - ne            pinigai = 44 + 4 = 48
48 > 50 - ne            pinigai = 48 + 4 = 52
52 > 50 - taip
Pinigų bus 52.
```

26 pav. Paaiškinimo lentelė

2.2. Užduotis- kelias dienas reikės mesti į taupyklę, kad sutaupytumėte 50 eurų?

Mokytojas atlieka scratch aplinkoje:



27 pav. Užduotis scratch aplinkoje

2.3. Mokiniai tą patį bando padaryti <https://www.onlinegdb.com/#> aplinkoje:

//Pinigų skaičiavimas

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int main () {
```

```
    int pinigai = 0;
```

```
    int dienos = 0;
```

```
    while (pinigai < 50){
```

```
        pinigai = pinigai + 4;
```

```
        dienos = dienos + 1;
```

```
    }
```

```
    cout << "Sutaupyti pinigai: " << pinigai << endl;
```

```
    cout << "Pinigai bus sutaupyti per: " << dienos << endl;
```

```
    return 0;
```

```
}
```

2. 4. Martynas didelis smaližius. Po tėčio lova Martynas rado m saldainių. Berniukas pirmą dieną suvalgė 1 saldainį, antrą – 2, trečią – 3 ir t.t. Kiekvieną kitą dieną jis suvalgydavo vienu saldainiu daugiau negu prieš tai buvusią dieną. Per kiek dienų d berniukas suvalgys visus

```

saldainius.    Mažiau    saldainių    gali    likti    paskutinei    dienai.10
//
#include
using          namespace          std;
int            main          ()
{
int            m,            //          Saldainių          skaičius          slėptuvėje
k,            //          Kiekvieną          dieną          suvalgomų          saldainių          skaičius
d;            //          Dienų,          per          kurias          bus          suvalgyti          visi          saldainiai,          skaičius
cout << "Įveskite          saldainių          skaičių          slėptuvėje:          ";          cin >>          m;
d            =            0;          k            =            0;
while          (m            >            0)          {
k            =            k            +            1;
m            =            m            -            k;
d            =            d            +            1;
}
cout << "Martynas          visus          saldainius          suvalgys          per          " <<          d <<          "          dienas          (-ų)." <<          endl;
return
}

```

2.5 Stipresniems mokiniams:

Sodininkas pradėjo dirbti sode pavasarį. Jis pirmą dieną išgenėjo m_1 , antrą – m medžių daugiau negu pirmą, trečią – m medžių daugiau negu antrą ir t.t. Parašykite programą, kuri apskaičiuotų, kiek dienų d dirbs sodininkas, kol išgenės n sode augančių medžių. Paskutinei dienai gali likti mažiau medžių. Visi duomenys yra sveikojo tipo. Patikrinkite programą: jei $m_1 = 3$, $m = 1$, $n = 16$, išveskite pranešimą į ekraną: „Sodininkas dirbs 4 dienas“.¹¹

3. ETAPAS (8 min) darbų vertinimas, įsivertinimas, refleksija:

Mokiniai atsivertę savo elektroninius paštus laiškuose randa pateiktus klausimus ir du atsakymus:

1. Ką reiškia four?

- a) žinomo kintamojo žymėjimas
- b) nežinomo kintamojo žymėjimas

2. Ką reiškia while?

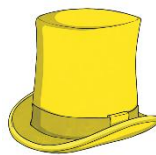
- a) nežinomo kintamojo žymėjimas
- b) žinomo kintamojo žymėjimas

¹⁰ Laisvai platinamas uždavinynas sukurtas doc. R. Burbaitės

¹¹ Laisvai platinamas uždavinynas sukurtas doc. R. Burbaitės

Atsakę persiunčia mokytojai.

Mokytoja kilo kepure, mokiniai atsakinėja:



aptaria kuo naudotas sprendimų būdas yra geras

28 pav. Teigiamas vertinimas

kur silpniau, sunkiau pavyko suvokti dėstomą medžiagą ar sprendžiant uždavinius



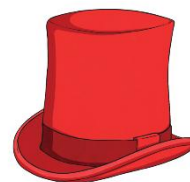
29 pav. Kritiškas vertinimas

padaro išvadas ko išmoko



30 pav. Apibendrinimas

kaip jautėsi spęsdami sužinoję naują temą ir spęsdami uždavinius.



31 pav. Subjektyvus vertinimas