

9-10 (I-II gimnazijos) klasė

Išorinių duomenų naudojimas. Pradinių duomenų saugojimas tekstiniuose failuose. Skaitymas iš failo (tiesiniai ir šakotieji algoritmai) (Olga Bielko)

Pasiekimų sritis	(B) Algoritmai ir programavimas. 29.2.2. Išorinių duomenų naudojimas
Klasė	9-10
Tema	Išorinių duomenų naudojimas. Pradinių duomenų saugojimas tekstiniuose failuose. Skaitymas iš failo (tiesiniai ir šakotieji algoritmai).
Integruojami dalykai, pasiekimai	Anglų kalba, lietuvių kalba, matematika.
Kompetencijos	<u>Pažinimo</u> – mokiniai ugdomi informatinį mąstymą – gebėjimą spręsti įvairias aplinkos problemas (uždavinius) analizuodami ir apdorojami duomenis, modeliuodami problemų sprendimo procesus ir, svarbiausia, siekdami automatizuoti tuos sprendimus.. <u>Skaitmeninė</u> – mokiniai užtikrintai ir sumaniai naudojami skaitmeninėmis priemonėmis siekdami veiksmingo ir konstruktyvaus mokymo(si), suvokia informacijos apdorojimo procesus, jų svarbą. <u>Komunikavimo</u> – mokiniai naudojami virtualiosiomis darbo priemonėmis mokymuisi įvairius dalykus, konstruktyviai planuoja ir organizuoja savo darbą, geba laiku gauti ir perduoti reikiamą informaciją.
Tikslas	Mokyti pradinius programos duomenis įvesti iš tekstinio failo.
Uždaviniai	1. Išmoksiti perskaityti pradinius duomenis iš tekstinio failo. 2. Įgytas žinias ir įgūdžius panaudosite rašydami programas, kuriose veiksmai atliekami nuosekliai (tiesiniai algoritmai). 3. Įgūdžius įtvirtinsite sprenddami uždavinius, kuriuose reikia panaudoti šakotuosius algoritmus.
Planuojamas rezultatas	1. Gebėsite parengti programą, kuri pradinius duomenis skaito iš tekstinio failo. 2. Gebėsite pritaikyti įgytas žinias ir įgūdžius sprenddami uždavinius.
Specifinės priemonės / programinė įranga	Kompiuteris, programavimo aplinka Code::Blocks.
Mokymosi metodai	Praktiniu tyrimu grįsta pamoka. Mokiniam pateikiamas paaiškinimas, kaip užrašomas programos algoritmas panaudojant tekstinį duomenų saugojimo failą ir jo skaitymas atliekant skaičiavimus. Mokiniam pateikiamas uždavinys ir duodamas laikas savarankiškai parašyti atitinkamą uždavinio algoritmą pagal išnagrinėtą pavyzdį. <u>Alternatyva</u> : mokiniai išnagrinėja uždavinį ir jei viskas aišku, gali neatlikti užduočių, numatytų plane, o patys sugalvoti uždavinius ir juos išspręsti.

	Mokiniai dirba individualiai arba porose, bendradarbiauja tarpusavyje ir konsultuojasi su mokytoju, rašydami programos algoritmą su tekstiniu failu. Parašytas programos, su duomenų skaitymu iš failo, pasitikrina.
Mokinių atlikto darbo vertinimas ir įsivertinimas	<ol style="list-style-type: none"> 1. Jei mokiniai dirbo pagal pamokos planą, galima jų paprašyti įvertinti kiekvieną žingsnį užpildant mokytojo parengtą apklausą. 2. Jei mokiniai patys išsikėlė uždavinius ir juos išsprendė, tuomet trumpai pristato atliktus darbus. Darbai vertinami pagal iš anksto apibrėžtus kriterijus. <p><u>Slenkstinis</u> – Įvardija, kad duomuo gali būti išorinis(B2.1). <u>Patenkinamas</u> – Nurodo, kaip duomenys nuskaityti ir išvedami į išorę (pavyzdžiui, failą) (B2.2). <u>Pagrindinis</u> – Spręsdamas problemas naudoja išorinius duomenis (B2.3). <u>Aukštesnysis</u> – Darbui su išoriniais duomenimis kuria paprogrames (B2.4).</p> <p>Mokiniai įsivertina kas buvo sunkiausia, lengviausia – kaupiamasis vertinimas.</p>
Žinios prieš	Mokiniai turi bazines žinias kaip naudotis Code::Blocks aplinka ir geba parašyti paprastą skaičiavimo programą.
Galimybės taikyti spec. poreikių mokiniams	Spec. poreikių mokiniams galima pateikti sukurta programą, kurioje jie galėtų atlikti paprastus pakeitimus. Pvz., pakeisti vieną žodį kitu, panaudoti duomenų faile skaičius iki 10 ir pan.
Patarimai kolegoms, kurie naudos parengtą medžiagą	<ol style="list-style-type: none"> 1. Patys atlikite visas pamokos plane numatytas užduotis. 2. Gabesniems mokiniams paruošti bent vieną papildomą užduotį. 3. Jei mokiniai dirbo pagal pavyzdį, parenkite klausimyną, kuris padėtų įvertinti mokinių pasiekimų lygius. 4. Jei aukštesniųjų gebėjimų mokiniai pasirinko alternatyvą, parenkite atlikto darbo pristatymo kriterijus ir pagal juos įvertinkite mokinių darbus. 5. Per pamoką mokiniai dirba savarankiškai, mokytojas yra pagalbininkas ir konsultantas.

1 ETAPAS (10-13 minučių). Medžiagos pateikimas per pavyzdį.

Uždavinys sprendžiamas kartu su mokiniais ir panaudojamas algoritmas pradinių duomenų skaitymui iš tekstinio failo.

1. Skaitoma užduoties sąlyga.

Pietų kaina. Trys draugai Antanas, Pranas ir Jonas pietavo kavinėje. Antano pietūs kainavo a eurų, Prano – p , o Jono – j eurų. Parašykite programą, skaičiuojančią, kiek eurų s iš viso sumokėta.

Pradiniai duomenys, skaitomi iš failo	Rezultatai, rodomi ekrane
5 4 4	13

- Po to atsiverčiame Code:Blocks aplinką ir kuriame naują failą, kurį išsisaugojame mums tinkamoje vietoje ir automatiškai aplinka mums užkrauna šabloną.
- Kadangi mes dirbsime su failais, turime prisijungti biblioteką: **#include <fstream>**.

- Aprašome pradinių duomenų failą: **const char CD[] = "pietus_data.txt";** .

const – pastovus dydis;

char – parodo, kad yra simboliai;

CD – failo vardas, C – konstanta (pastovus dydis), D – duomenys;

[] – parodo, kad tai yra simbolių seka.

- Programoje susisiejame pradinių duomenų failo kintamąjį **fd** su failo vardu **CD**:

ifstream fd(CD); .

ifstream – nuo žodžio „input stream“ ;

fd – file data.

- Apsirašome pradinius duomenis. Jeigu įvesdavome klaviatūra naudodavome *cin*, tai dabar iš failo **fd** mes perskaitome pradinius duomenis:

fd >> >> ; .

- Uždarome pradinių duomenų failą: **fd.close();** .

- Užrašome skaičiavimus ir rezultato išvedimo į ekraną eilutes.

- Programos užrašymas:

```
#include <iostream>
```

```
#include <iomanip>
```

```
#include <cmath>
```

```
#include <fstream>
```

```
using namespace std;
```

```
const char CD[] = "pietus_data.txt";
```

```
int main() {
```

```
    ifstream fd(CD);
```

```
    int a, p, j;
```

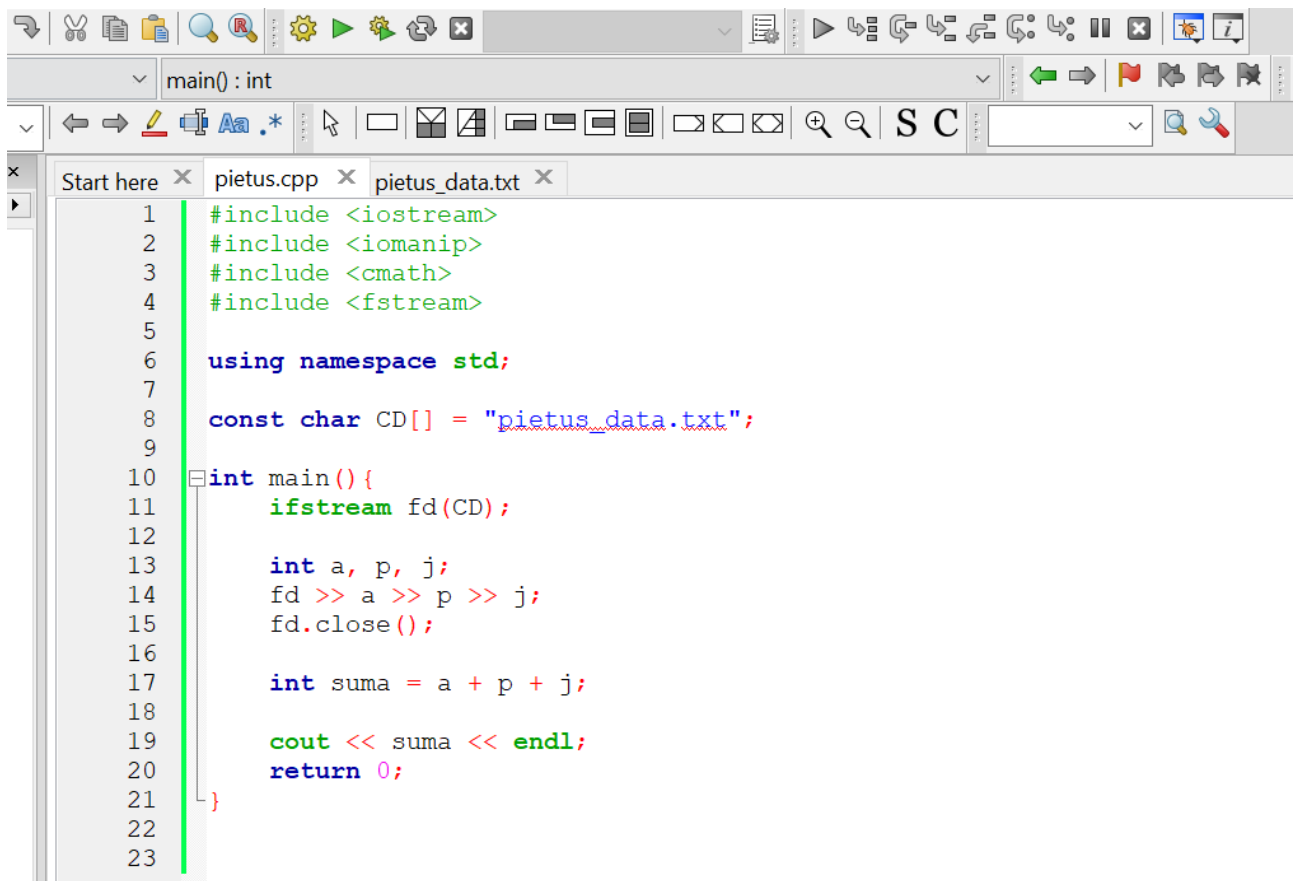
```
    fd >> a >> p >> j;
```

```
    fd.close();
```

```
int suma = a + p + j;

cout << suma << endl;

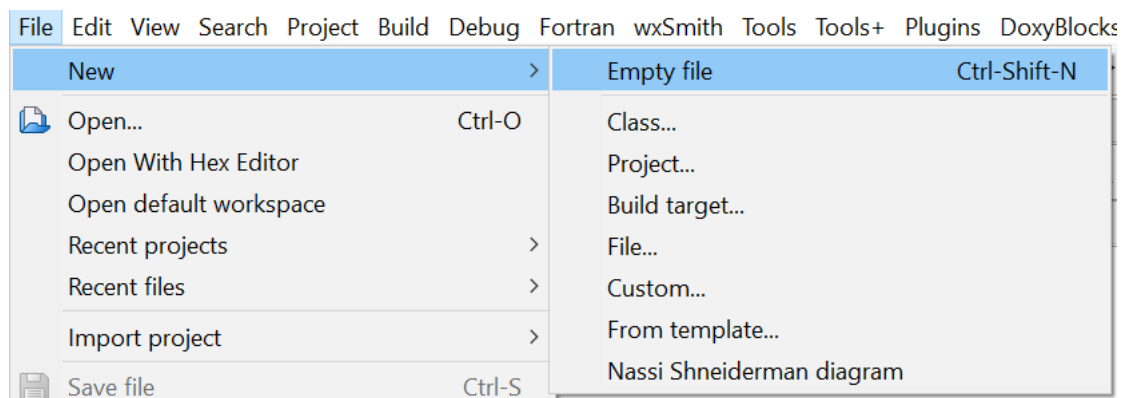
return 0;
}
```



```
Start here x pietus.cpp x pietus_data.txt x
1 #include <iostream>
2 #include <iomanip>
3 #include <cmath>
4 #include <fstream>
5
6 using namespace std;
7
8 const char CD[] = "pietus_data.txt";
9
10 int main() {
11     ifstream fd(CD);
12
13     int a, p, j;
14     fd >> a >> p >> j;
15     fd.close();
16
17     int suma = a + p + j;
18
19     cout << suma << endl;
20     return 0;
21 }
22
23
```

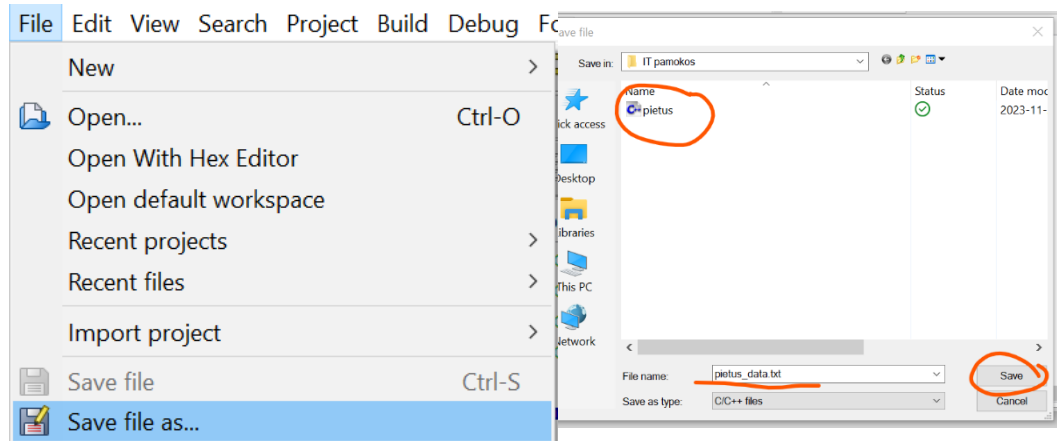
Pav. 1. Pradinių duomenų skaitymo iš failo programos užrašymas.

10. Sukuriame pradinių duomenų failą, tam, kad mes galėtume vykdyti programą:
File -> New -> Empty file.



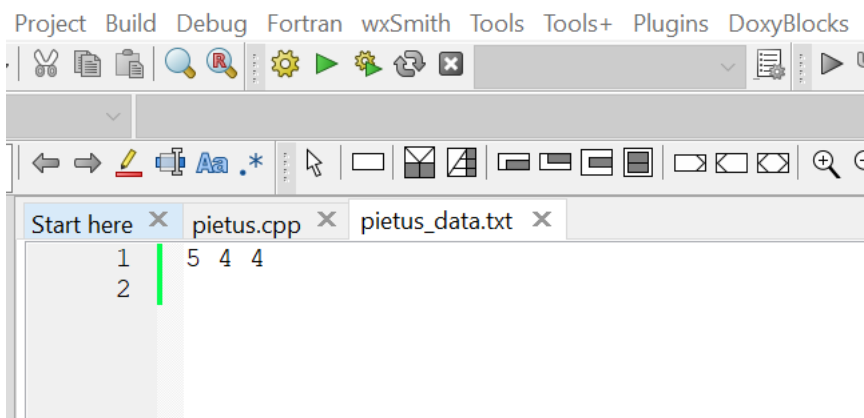
Pav. 2. Pradinių duomenų failo kūrimas.

11. Pradinių duomenų failas išsaugomas tame pačiame kataloge, kur yra išsaugotas užduoties failas:
File -> Save as -> ... -> pietus_data.txt .



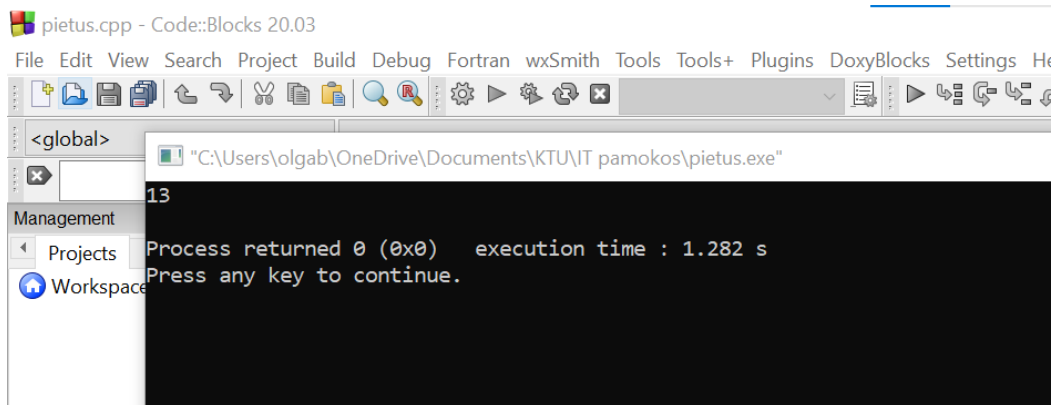
Pav. 3. Pradinių duomenų failo išsaugojimas.

12. Duomenys turi būti surašyti eilės tvarka, kaip yra aprašyti pradiniai duomenys programoje.



Pav. 4. Pradinių duomenų užrašymas faile.

13. Paleidžiant „Vykdyti“ rezultatas rodomas ekrane.



Pav. 5. Rezultatų išvedimas į ekraną.

2 ETAPAS (23-25 minutės). Praktinė pamokos dalis.

Užduočių atlikimas remiantis pavyzdžiu. Užduočių diferencijavimas.

Tiesiniai algoritmai

- *Uždaviniai silpniems mokiniams.*

1. Nurašyti programą į Code::blocks aplinką. Pakeisti pradinis duomenis ir išveskite rezultatą į ekraną.
2. Sukurkite skaičiuotuvą, kuris apskaičiuotų dviejų sveikųjų skaičių sumą, skirtumą ir sandaugą.

Pradiniai duomenys, skaitomi iš failo	Rezultatai, rodomi ekrane
5 6	$5 + 6 = 11$ $5 - 6 = -1$ $5 * 6 = 30$

- *Vidutinio sunkumo uždaviniai.*

1. **Dienos.** Nuo metų pradžios praėjo d dienų. Parašykite programą, kuri apskaičiuotų, kiek savaitių s praėjo nuo metų pradžios.
Pasitikrinkite: kai $d = 23$, $s = 3$.

Pradiniai duomenys, skaitomi iš failo	Rezultatai, rodomi ekrane
23	3

2. **Kriaušių dalyba.** Klasėje mokosi n mokinių. Kiekvienas atnešė po k kriaušių. Mokytoja atnešė m kriaušių. Mokytoja nusprendė visiems mokiniams padalinti kriaušes po lygiai. Parašykite programą, kuri apskaičiuotų, po kiek kriaušių teks kiekvienam mokiniui klasėje.
Pasitikrinkite: kai $n = 8$, $k = 2$, $m = 16$.

Pradiniai duomenys, skaitomi iš failo	Rezultatai, rodomi ekrane
16 2 8	4

- *Uždaviniai pažengusiems mokiniams.*

1. **Dovanos.** Kalėdų senelis, ruošdamasis aplankyti 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?

Pasitikrink: kai $n = 20, k = 3, v = 4$, ekrane rodoma: $d = 4, m = 1$.

Pradiniai duomenys, skaitomi iš failo	Rezultatai, rodomi ekrane
20 3 4	4 1

2. **Gimtadienis.** Andrius setpintojo gimtadienio progą gavo n balionų. Su draugais nusprendė balionus paleisti į dangų. Dalis pučamų 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? Parašykite program šiam uždaviniui spręsti.

Pasitikrinkite: kai $n = 77, d = 7, k = 3$, ekrane rodoma: $m = 9, a = 11$.

Pradiniai duomenys, skaitomi iš failo	Rezultatai, rodomi ekrane
77 7 3	9 11

Šakotieji algoritmai

- *Uždaviniai silpniems mokiniams.*

1. **Didžioji kūdrinė varlė.** Didžioji kūdrinė varlė – beuodegis varliagyvis. Ji sveria m gramų (m - realusis skaičius). Mokslininkai, tyrinėjantys didžiąsias kūdrines varles, nusprendė n varlių stebėti. Jei stebimos varlės svers daugiau kaip 5 kilogramus, ekrane turi būti rodomas pranešimas „Varlių stebėjimui pakanka“, jei mažiau – „Varlių stebėjimui per mažai“.

Pradiniai duomenys, skaitomi iš failo	Rezultatai, rodomi ekrane
100 1000 <hr/> Paiškinimas: 100 (Kiek sveria varlė?) 1000 (Kiek varlių norima stebėti?)	Varlių stebėjimui pakanka
75 50 <hr/> Paiškinimas: 75 (Kiek sveria varlė?) 50 (Kiek varlių norima stebėti?)	Varlių stebėjimui per mažai

- *Vidutinio sunkumo uždaviniai.*

1. **Skaičiai.** Per matematikos pamoką teko nustatyti, kurie iš duotųjų teigiamų skaičių a , b ir c yra skaičiaus trys kartotiniai. Jei yra keli, reikia išvesti pirmą rastą, jeigu nėra nei vieno, reikia išvesti - nėra. Parašykite programą, naudodami duomenų failą, sprendžiančią šį uždavinį.

Pradiniai duomenys, skaitomi iš failo	Rezultatai, rodomi ekrane
5 2 3	3
6 3 7	6
1 2 5	nėra

2. **Jonuko problema.** Jonukas žino, kad mokytoja jam gali duoti spręsti vieną iš trijų kontrolinio darbo variantų. Atlikdamas užduotį jis gaus du skaičius a ir b , o jam reikės apskaičiuoti x reikšmę pagal vieną iš trijų formulių.

Pirmas variantas: $x = ab + 3$;

Antras variantas: $x = a + b$;

Trečias variantas: $x = a - b$.

Parašykite programą, naudodami duomenų failą, kuri padėtų Jonukui pasitikrinti, ar jis teisingai skaičiuoja x reikšmę spręsdamas kiekvieno varianto užduotį.

Pradiniai duomenys, skaitomi iš failo	Rezultatai, rodomi ekrane
1 5 2	$x = 13$
2 5 2	$x = 2$
3 2 3	$x = -1$

- *Uždaviniai pažengusiems mokiniams.*

1. **Bandelės.** Julius, grįžęs iš mokyklos namo, rado tokį mamos raštelį:

Nueik į parduotuvę ir nupirk bandelių su varške.

Jei viena bandelė kainuos ne daugiau kaip a eurų, nupirk $n1$ bandelių;

Jei daugiau nei a , bet mažiau nei b , nupirk $n2$ bandelių;

O jei kainuos b arba daugiau, negu b - $n3$ bandelių.

Nuėjęs į parduotuvę, Julius pamatė, kad bandelė kainuoja k eurų.

Parašykite programą, kuri apskaičiuotų, kiek pinigų už bandeles sumokės Julius.

(a visada mažiau nei b).

Pradiniai duomenys, skaitomi iš failo (įvedami tik skaičiai, be reikšmių)	Rezultatai, rodomi ekrane
1.50 1.70 (kainos a ir b) 3 2 1 (kiekiai n1, n2, n3) 1.30 (bandelės kaina)	3.90
1.50 1.70 (kainos a ir b) 3 2 1 (kiekiai n1, n2, n3) 1.60 (bandelės kaina)	3.20
1.50 1.70 (kainos a ir b) 3 2 1 (kiekiai n1, n2, n3) 2.05 (bandelės kaina)	2.05

3 ETAPAS ⌚ 5-7 minučių

Su mokiniais apibendrinamas atliktas darbas.

1. Programų testavimo aplinka. Jeigu mokykla pasirenka testavimo aplinką, mokiniai įkelia savo sudarytas programas ir mokytojas mato jų pažangą.
2. Jei mokiniai dirbo pagal scenarijų, galima jų paprašyti įvertinti kiekvieną žingsnį užpildant mokytojo parengtą apklausą (galima naudoti Google Forms).

Galimi klausimai apklausoje:

Ką išmokau, atlikau?

Ką pavyko atlikti gerai?

Kas rodo, kad atlikau gerai?

Ką pavyko atlikti savarankiškai?

Kuo labiausiai džiaugiuosi?

Kokių sunkumų kilo?

Kokius savo neteisingus veiksmus ar klaidas pastebėjau?

Kaip įveikiau kilusius sunkumus?

Kieno ir kokios pagalbos prireikė?

Ką kitą kartą galima būtų daryti kitaip?

Išorinių duomenų naudojimas. Rezultatų saugojimas tekstiniuose failuose. Rašymas į failą (tiesiniai ir šakotieji algoritmai) (Olga Bielko)

Pasiekimų sritis	(B) Algoritmai ir programavimas. 29.2.2. Išorinių duomenų naudojimas
Klasė	9-10
Tema	Išorinių duomenų naudojimas. Rezultatų saugojimas tekstiniuose failuose. Rašymas į failą (tiesiniai ir šakotieji algoritmai).
Integruojami dalykai, pasiekimai	Anglų kalba, lietuvių kalba, matematika.
Kompetencijos	<u>Pažinimo</u> – mokiniai ugdomi informatinį mąstymą – gebėjimą spręsti įvairias aplinkos problemas (uždavinius) analizuodami ir apdorojami duomenis, modeliudami problemų sprendimo procesus ir, svarbiausia, siekdami automatizuoti tuos sprendimus.. <u>Skaitmeninė</u> – mokiniai užtikrintai ir sumaniai naudojami skaitmeninėmis priemonėmis siekdami veiksmingo ir konstruktyvaus mokymo(si), suvokia informacijos apdorojimo procesus, jų svarbą. <u>Komunikavimo</u> – mokiniai naudojami virtualiosiomis darbo priemonėmis mokymuisi įvairius dalykus, konstruktyviai planuoja ir organizuoja savo darbą, geba laiku gauti ir perduoti reikiamą informaciją.
Tikslas	Mokytis gautus programos rezultatus įrašyti į tekstinį failą.
Uždaviniai	4. Išmoksite užrašyti programoms rezultatus į tekstinį failą.

	<p>5. Įgytas žinias ir įgūdžius panaudosite rašydami programas, kuriose veiksmai atliekami nuosekliai (tiesiniai algoritmai).</p> <p>6. Įgūdžius įtvirtinsite sprenddami uždavinius, kuriuose reikia panaudoti šakotuosius algoritmus.</p>
Planuojamas rezultatas	<p>3. Gebėsite parengti programą, kuri gautus rezultatus rašys į tekstinį failą.</p> <p>4. Gebėsite pritaikyti įgytas žinias ir įgūdžius sprenddami uždavinius.</p>
Specifinės priemonės / programinė įranga	Kompiuteris, programavimo aplinka Code::Blocks.
Mokymosi metodai	<p>Praktiniu tyrimu grįsta pamoka.</p> <p>Mokiniam pateikiamas paaiškinimas, kaip užrašomas programos algoritmas panaudojant tekstinį saugojimo failą ir atlikus skaičiavimus rezultatų rašymą į tekstinį failą. Mokiniam pateikiamas uždavinys ir duodamas laikas savarankiškai parašyti atitinkamą uždavinio algoritmą pagal išnagrinėtą pavyzdį.</p> <p><u>Alternatyva:</u> mokiniai išnagrinėja uždavinį ir jei viskas aišku, gali neatlikti užduočių, numatytų plane, o patys sugalvoti uždavinius ir juos išspręsti.</p> <p>Mokiniai dirba individualiai arba porose, bendradarbiauja tarpusavyje ir konsultuojasi su mokytoju, rašydami programos algoritmą su tekstiniu failu. Parašytas programos, su rezultatų rašymu į failą, patikrinama.</p>
Mokinių atlikto darbo vertinimas ir įsivertinimas	<p>3. Jei mokiniai dirbo pagal pamokos planą, galima jų paprašyti įvertinti kiekvieną žingsnį užpildant mokytojo parengtą apklausą.</p> <p>4. Jei mokiniai patys išsikėlė uždavinius ir juos išsprendė, tuomet trumpai pristato atliktus darbus. Darbai vertinami pagal iš anksto apibrėžtus kriterijus.</p> <p><u>Slenkstinis</u> – Įvardija, kad duomuo gali būti išorinis(B2.1).</p> <p><u>Patenkinamas</u> – Nurodo, kaip duomenys nuskaitomi ir išvedami į išorę (pavyzdžiui, failą) (B2.2).</p> <p><u>Pagrindinis</u> – Sprenddamas problemas naudoja išorinius duomenis (B2.3).</p> <p><u>Aukštesnysis</u> – Darbui su išoriniais duomenimis kuria paprogrames (B2.4).</p> <p>Mokiniai įsivertina kas buvo sunkiausia, lengviausia – kaupiamasis vertinimas.</p>
Žinios prieš	Mokiniai turi bazines žinias kaip naudotis Code::Blocks aplinka ir geba parašyti paprastą skaičiavimo programą.
Galimybės taikyti spec. poreikių mokiniams	Spec. poreikių mokiniams galima pateikti sukurtą programą, kurioje jie galėtų atlikti paprastus pakeitimus. Pvz., pakeisti vieną žodį kitu, panaudoti duomenų faile skaičius iki 10 ir pan.
Patarimai kolegoms, kurie	<p>6. Patys atlikite visas pamokos plane numatytas užduotis.</p> <p>7. Gabesniems mokiniams paruošti bent vieną papildomą užduotį.</p>

naudos parengtą medžiagą	8. Jei mokiniai dirbo pagal pavyzdį, parenkite klausimyną, kuris padėtų įvertinti mokinių pasiekimų lygius. 9. Jei aukštesniųjų gebėjimų mokiniai pasirinko alternatyvą, parenkite atlikto darbo pristatymo kriterijus ir pagal juos įvertinkite mokinių darbus. 10. Per pamoką mokiniai dirba savarankiškai, mokytojas yra pagalbininkas ir konsultantas.
--------------------------	--

1 ETAPAS (10-13 minučių). Medžiagos pateikimas per pavyzdį.

Imama praeitos pamokos užduotis ir papildoma rezultatų išvedimu į tekstinį failą algoritmu. Uždavinys sprendžiamas kartu su mokiniais.

14. Skaitoma uždavinio sąlyga.

Pietų kaina. Trys draugai Antanas, Pranas ir Jonas pietavo kavinėje. Antano pietūs kainavo a eurų, Prano – p , o Jono – j eurų. Parašykite programą, skaičiuojančią, kiek eurų s iš viso sumokėta.

Pradiniai duomenys, skaitomi iš failo	Rezultatai, rodomi faile
5 4 4	13

15. Po to atsiverčiame Code:Blocks aplinką ir atidarome šio uždavinio išsaugotą failą.

16. Patikriname ar yra biblioteka: **#include <fstream>**.

17. Pradiniai duomenys aprašomi failu: **const char CD[] = "pietus_data.txt";** .
const – pastovus dydis;
char – parodo, kad yra simboliai;
CD – failo vardas, C – *constant* (pastovus dydis), D – duomenys;
[] – parodo, kad tai yra simbolių seka.

18. Aprošomas rezultatu failas: **const char CR[] = "pietus_res.txt";**
const – pastovus dydis;
char – parodo, kad yra simboliai;
CR – failo vardas, C – *constant* (pastovus dydis), R – rezultatas;
[] – parodo, kad tai yra simbolių seka.

19. Kintamasis **fd** yra susietas su duomenų failu vardu **CD**: **ifstream fd(CD);** .
ifstream – nuo žodžių „*input stream*“ ;
fd – *file data*.

20. Susiejame rezultatų failo kintamąjį **fr** su failo vardu **CR**: **ofstream fr(CR);**
ofstream – nuo žodžių „*output stream*“
fr – *file result*.

21. Duomenys yra aprašyti: **int a, p, j;**

22. Pradiniai duomenys yra aprašyti naudojant kintamąjį **fd**:
fd >> a >> p >> j;

23. Pradinių duomenų failas uždaromas: **fd.close();**

24. Skaičiavimai užrašyti: **int suma = a + p + j;**

25. Atlikus skaičiavimus užrašomi rezultato išvedimo į tekstinį failą algoritmą: **fr << suma**
<< endl;

26. Rezultatų failas uždaromas: **fr.close();**

27. Programos užrašymas:

```
#include <iostream>
#include <iomanip>
#include <cmath>
#include <fstream>

using namespace std;

const char CD[] = "pietus_data.txt";
const char CR[] = "pietus_res.txt";

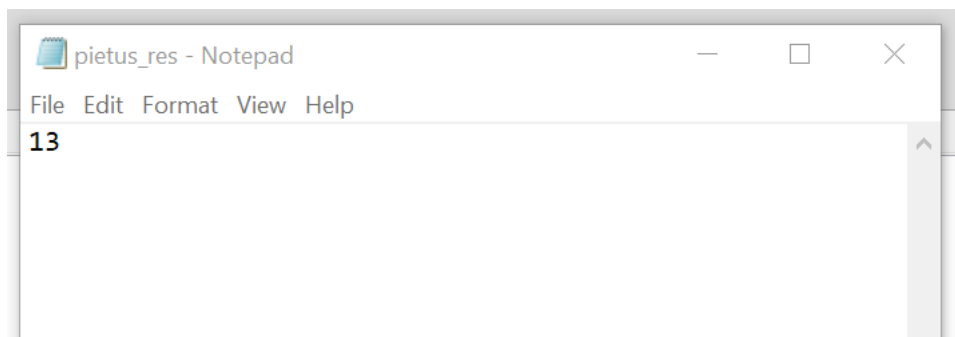
int main() {
    ifstream fd(CD);
    ofstream fr(CR);

    int a, p, j;
    fd >> a >> p >> j;
    fd.close();

    int suma = a + p + j;

    fr << suma << endl;
    fr.close();

    return 0;
}
```

Pav. 3. Rezultatas tekstiniame faile.

2 ETAPAS (23-25 minutės). Praktinė pamokos dalis.

Užduočių atlikimas remiantis pavyzdžiu. Užduočių diferencijavimas.

Tiesiniai algoritmai

- *Uždaviniai silpniems mokiniams.*
3. Nurašyti programą į Code::blocks aplinką. Pakeisti pradinis duomenis ir išveskite rezultatą į failą.
 4. Sukurkite skaičiuotuvą, kuris apskaičiuotų dviejų sveikųjų skaičių sumą, skirtumą ir sandaugą. Rezultatą išveskite į failą.

Pradiniai duomenys, skaitomi iš failo	Rezultatai, rodomi faile
5 6	$5 + 6 = 11$ $5 - 6 = -1$ $5 * 6 = 30$

- *Vidutinio sunkumo uždaviniai.*
3. **Automobilis.** Automobilių tunelio po Nepriklausomybės aikšte Vilniuje ilgis lygus 264 m. Parašykite programą, kuri apskaičiuotų, kelias sekundes s automobilis važiuoja šiuo tuneliu, jei jo greitis yra v km/h? Rezultatus pateikite šimtųjų tikslumu.

Pradiniai duomenys, skaitomi iš failo	Rezultatai, rodomi faile
60	15.84

4. **Trapecijos plotas.** Parašykite programą, kuri, įvedus trapecijos pagrindų a ir b bei aukštinės h ilgius, apskaičiuotų trapecijos plotą. Pradinius duomenis turi būti skaitomi iš tekstinio failo, o rezultatas rašomas į tekstinį failą.

Pradiniai duomenys, skaitomi iš failo	Rezultatai, rodomi faile
5 3 4	16

• *Uždaviniai pažengusiems mokiniams.*

3. **Puodelių pakavimas.** Į vieną kartoninę dėžutę telpa trys puodeliai. Pakuotojas užključia dėžutę ir išsiunčia ją į parduotuvę, jei ji pilna. Iš viso reikia supakuoti m puodelių. Parašykite programą, kuri apskaičiuotų, kelios bus pilnos dėžutės ir kiek puodelių liks nesupakuota. Pradiniai duomenys skaitomi iš failo, o rezultatus išveskite į failą.

Pradiniai duomenys, skaitomi iš failo	Rezultatai, rodomi faile
7	2 1

4. **Keltas.** Parašykite programą, kuri suskaičiuotų, kelis kartus keltui teks kelti per upę k automobilių, jeigu vienu metu jis gali perkelti m automobilių. Keltas kelia tik tada, kai yra pilnas (susidaro m automobilių.) Taip pat išveskite automobilių skaičių, kuriems persikelti per upę nepavyks (jei buvo 11 automobilių, o į keltą telpa 10, tai 10 perkels, o vienas liks neperkeltas).

Pradiniai duomenys, skaitomi iš failo	Rezultatai, rodomi faile
27 10	2 7
Paaiškinimas: 27 (automobilių skaičius) 10 (į keltą telpa automobilių)	Paaiškinimas: 2 (perkels per kartą) 7 (liks neperkelta)

Šakotieji algoritmai

- **Uždaviniai silpniems mokiniams.**

2. **Didžioji kūdrinė varlė.** Didžioji kūdrinė varlė – beuodegis varliagyvis. Ji sveria m gramų (m - realusis skaičius). Mokslininkai, tyrinėjantys didžiąsias kūdrines varles, nusprendė n varlių stebėti. Jei stebimos varlės svers daugiau kaip 5 kilogramus, Rezultatų faile turi būti rodomas pranešimas „Varlių stebėjimui pakanka“, jei mažiau – „Varlių stebėjimui per mažai“.

Pradiniai duomenys, skaitomi iš failo (rašomi tik skaičiai per tarpą)	Rezultatai, rodomi faile
100 1000 <hr/> Paaiškinimas: 100 (Kiek sveria varlė?) 1000 (Kiek varlių norima stebėti?)	Varlių stebėjimui pakanka
75 50 <hr/> Paaiškinimas: 75 (Kiek sveria varlė?) 50 (Kiek varlių norima stebėti?)	Varlių stebėjimui per mažai

- **Vidutinio sunkumo uždaviniai.**

3. **Matematika.** Petriukas per pusmetį gavo 5 matematikos pažymius. Mokytoja nusprendė padaryti vaikams staigmeną: mokiniai, kurių pažymių vidurkis yra didesnis už 9, gaus tris saldinius, o mokiniams, kurių vidurkis yra tarp 7 ir 9, įskaitant intervalo galus, bus apdovanoti dviem saldainiais. Visi likusieji gaus po vieną saldainį. Parašykite programą, kuri pagal įvestus Petriuko pažymius apskaičiuotų, kiek saldainių jis gaus.

Pradiniai duomenys, skaitomi iš failo	Rezultatai, rodomi faile
8 9 6 5 10	2
10 10 8 9 10	3

5 5 4 5 5	1
-----------	---

- *Uždaviniai pažengusiems mokiniams.*

2. **Žiemojantys paukščiai.** Gamtininkas registruoja likusius žiemoti paukščius. Jo tikslas yra nustatyti, kurios iš trijų paukščių rūšių atstovų liko žiemoti daugiausia. Pradinių duomenų faile įrašykite trijų paukščių rūšių kiekius, parašykite programą, kuri surikiuotų juos iš eilės nuo didžiausio iki mažiausio ir apskaičiuotų, kiek skiriasi didžiausias ir mažiausias kiekiai. Rezultatų faile rodomas skirtumas tarp didžiausio ir mažiausio kiekio.

Pradiniai duomenys, skaitomi iš failo	Rezultatai, rodomi faile
1000 1500 1800	800
5000 3500 6200	2700

3 ETAPAS ⌚ 5-7 minučių

Su mokiniais apibendrinamas atliktas darbas.

3. Programų testavimo aplinka. Jeigu mokykla pasirenka testavimo aplinką, mokiniai įkelia savo sudarytas programas ir mokytojas mato jų pažangą.
4. Jei mokiniai dirbo pagal scenarijų, galima jų paprašyti įvertinti kiekvieną žingsnį užpildant mokytojo parengtą apklausą (galima naudoti Google Forms).

Galimi klausimai apklausoje:

Ką išmokau, atlikau?

Ką pavyko atlikti gerai?

Kas rodo, kad atlikau gerai?

Ką pavyko atlikti savarankiškai?

Kuo labiausiai džiaugiuosi?

Kokių sunkumų kilo?

Kokius savo neteisingus veiksmus ar klaidas pastebėjau?

Kaip įveikiau kilusius sunkumus?

Kieno ir kokios pagalbos prireikė?

Ką kitą kartą galima būtų daryti kitaip?

5. Jei mokiniai patys išsikėlė uždavinius ir juos išsprendė, tuomet trumpai pristato atliktus darbus.

Išorinių duomenų naudojimas. Pradinių duomenų saugojimas tekstiniuose failuose. Skaitymas iš failo ir rezultatų rašymas į failą (cikliniai algoritmai) (Olga Bielko)

Pasiekimų sritis	(B) Algoritmai ir programavimas. 29.2.2. Išorinių duomenų naudojimas
Klasė	9-10
Tema	Išorinių duomenų naudojimas. Pradinių duomenų saugojimas tekstiniuose failuose. Skaitymas iš failo ir rezultatų rašymas į failą (cikliniai algoritmai).
Integruojami dalykai, pasiekimai	Anglų kalba, lietuvių kalba, matematika.
Kompetencijos	<u>Pažinimo</u> – mokiniai ugdomi informatinį mąstymą – gebėjimą spręsti įvairias aplinkos problemas (uždavinius) analizuodami ir apdorojami duomenis, modeliudami problemų sprendimo procesus ir, svarbiausia, siekdami automatizuoti tuos sprendimus.. <u>Skaitmeninė</u> – mokiniai užtikrintai ir sumaniai naudojami skaitmeninėmis priemonėmis siekdami veiksmingo ir konstruktyvaus mokymo(si), suvokia informacijos apdorojimo procesus, jų svarbą. <u>Komunikavimo</u> – mokiniai naudojami virtualiosiomis darbo priemonėmis mokydami įvairius dalykus, konstruktyviai planuoja ir organizuoja savo darbą, geba laiku gauti ir perduoti reikiamą informaciją.
Tikslas	Mokytis pradinius programos duomenis įvesti iš tekstinio failo ir rezultatus saugoti tekstiname faile.
Uždaviniai	7. Išmokssite užrašyti programoms pradinius duomenis iš tekstinio failo ir rezultatus saugoti tekstiname faile. 8. Įgytas žinias ir įgūdžius panaudosite rašydami programas, kuriose veiksmai atliekami ciklais (cikliniai algoritmai).
Planuojamas rezultatas	5. Gebėsite parengti programą, kuri pradinius duomenis skaito iš tekstinio failo ir rezultatus saugo tekstiname faile. 6. Gebėsite pritaikyti įgytas žinias ir įgūdžius sprenddami uždavinius.
Specifinės priemonės programinė įranga	/ Kompiuteris, programavimo aplinka Code::Blocks.
Mokymosi metodai	Praktiniu tyrimu grįsta pamoka. Mokiniam pateikiamas paaiškinimas, kaip užrašomas programos algoritmas panaudojant tekstinį saugojimo failą jo skaitymas atliekant skaičiavimus. Mokiniam pateikiamas uždavinys ir duodamas laikas savarankiškai parašyti atitinkamą uždavinio algoritmą pagal išnagrinėtą pavyzdį. Alternatyva: mokiniai išnagrinėja uždavinį ir jei viskas aišku, gali neatlikti užduočių, numatytų plane, o patys sugalvoti uždavinius ir juos išspręsti.

	Mokiniai dirba individualiai arba porose, bendradarbiauja tarpusavyje ir konsultuojasi su mokytoju, rašydami programos algoritmą su tekstiniu failu. Parašytas programos, su duomenų skaitymu iš failo, pasitikrina.
Mokinių atlikto darbo vertinimas ir įsivertinimas	<p>5. Jei mokiniai dirbo pagal pamokos planą, galima jų paprašyti įvertinti kiekvieną žingsnį užpildant mokytojo parengtą apklausą.</p> <p>6. Jei mokiniai patys išsikėlė uždavinius ir juos išsprendė, tuomet trumpai pristato atliktus darbus. Darbai vertinami pagal iš anksto apibrėžtus kriterijus.</p> <p><u>Slenkstinis</u> – Įvardija, kad duomuo gali būti išorinis(B2.1).</p> <p><u>Patenkinamas</u> – Nurodo, kaip duomenys nuskaitomi ir išvedami į išorę (pavyzdžiui, failą) (B2.2).</p> <p><u>Pagrindinis</u> – Spręsdamas problemas naudoja išorinius duomenis (B2.3).</p> <p><u>Aukštesnysis</u> – Darbui su išoriniais duomenimis kuria paprogrames (B2.4).</p> <p>Mokiniai įsivertina kas buvo sunkiausia, lengviausia – kaupiamasis vertinimas.</p>
Žinios prieš	Mokiniai turi bazines žinias kaip naudotis Code::Blocks aplinka ir geba parašyti paprastą skaičiavimo programą.
Galimybės taikyti spec. poreikių mokiniams	Spec. poreikių mokiniams galima pateikti sukurta programą, kurioje jie galėtų atlikti paprastus pakeitimus. Pvz., pakeisti vieną žodį kitu, panaudoti duomenų faile skaičius iki 10 ir pan.
Patarimai kolegoms, kurie naudos parengtą medžiagą	<p>11. Patys atlikite visas pamokos plane numatytas užduotis.</p> <p>12. Gabesniems mokiniams paruošti bent vieną papildomą užduotį.</p> <p>13. Jei mokiniai dirbo pagal pavyzdį, parenkite klausimyną, kuris padėtų įvertinti mokinių pasiekimų lygius.</p> <p>14. Jei aukštesniųjų gebėjimų mokiniai pasirinko alternatyvą, parenkite atlikto darbo pristatymo kriterijus ir pagal juos įvertinkite mokinių darbus.</p> <p>15. Per pamoką mokiniai dirba savarankiškai, mokytojas yra pagalbininkas ir konsultantas.</p>

1 ETAPAS (10-13 minučių). Medžiagos pateikimas per pavyzdį.

Uždavinys sprendžiamas kartu su mokiniais. Paimamas užduotį su „Nežinomo kartojimų skaičiaus ciklu (WHILE)“.

29. Skaitoma uždavinio sąlyga.

Medžiai. Pavasarį pradėjęs dirbti sode sodininkas pirmąją dieną išgenėjo m_1 , antrąją – m medžių daugiau negu pirmąją, trečiąją – m medžių daugiau negu antrąją ir t.t. Parašykite programą, skaičiuojančią kiek dienų d dirbs sodininkas, kol išgenės n sode augančių medžių. Paskutinei dienai gali likti mažiau medžių. Visi duomenys yra sveikojo tipo.

Pasitikrinkite: kai $m1 = 3$, $m = 1$, $n = 16$, tuomet kompiuterio ekrane turi būti rodoma: Sodininkas dirbs 4 dienas.

Pradiniai duomenys, skaitomi iš failo	Rezultatai, rodomi faile
3 1 16	Sodininkas dirbs 4 dienas

30. Po to atsiverčiame Code:Blocks aplinką ir atidarome langl naujam projektui. Užvadiname projektą „Medziai“ ir išsaugojame mums tinkamoje vietoje.

31. Kadangi mes dirbsime su failais, turime prisijungti biblioteką: **#include <fstream>**.

32. Aprašome pradinių duomenų failą: **const char CD[] = “medziai_data.txt“;** .

const – pastovus dydis;

char – parodo, kad yra simboliai;

CD – failo vardas, C – *constant* (pastovus dydis), D – duomenys;

[] – parodo, kad tai yra simbolių seka.

33. Aprošomas rezultatų failas: **const char CR[] = “pietus_res.txt“;**

const – pastovus dydis;

char – parodo, kad yra simboliai;

CR – failo vardas, C – *constant* (pastovus dydis), R – rezultatas;

[] – parodo, kad tai yra simbolių seka.

34. Programoje aprašome pradinių duomenų failo kintamąjį **fd** su failo vardu **CD**:

ifstream fd(CD); .

ifstream – nuo žodžių „*input stream*“ ;

fd – *file data*.

35. Susiejame rezultatų failo kintamąjį **fr** su failo vardu **CR**: **ofstream fr(CR);**

ofstream – nuo žodžių „*output stream*“

fr – *file result*.

36. Apsirašome pradinius duomenis. Jeigu įvesdavome klaviatūra naudodavome *cin*, tai dabar iš failo **fd** mes perskaitome pradinius duomenis:

fd >> ... >> ; .

37. Pradinių duomenų failas uždaromas: **fd.close();**

38. Užrašome skaičiavimus ir rezultato išvedimo į ekraną eilutes.

39. Atlikus skaičiavimus užrašomi rezultato išvedimo į tekstinį failą algoritmą: **fr << ... << endl;**

40. Rezultatų failas uždaromas: **fr.close();**

41. Programos užrašymas:

```
#include <iostream>
#include <fstream>

using namespace std;

const char CD[] = "medziai_data.txt";
const char CR[] = "medziai_res.txt";

int main(){
    ifstream fd(CD);
    ofstream fr(CR);

    int m1, m, d, n, s; // s-pirmoj dienoj
    fd >> m1 >> m >> n;
    fd.close();

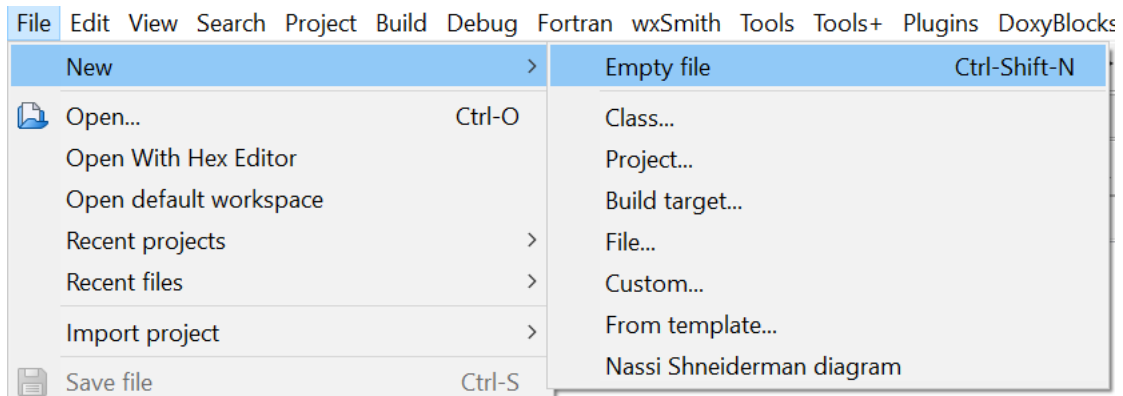
    d=1;
    s=m1;
    while (m1<n) {
        d=d+1;
        s=s+m; // kiek ta diena isgenejo medziu
        m1=s+m1; // kiek isviso isgenejo medziu
    }
    fr << "Sodininkas dirbs " << d << " dienas" << endl;
    fr.close();

    return 0;
}
```

```
1 #include <iostream>
2 #include <fstream>
3
4 using namespace std;
5
6 const char CD[] = "medziai_data.txt";
7 const char CR[] = "medziai_res.txt";
8
9 int main() {
10     ifstream fd(CD);
11     ofstream fr(CR);
12
13     int m1, m, d, n, s; // s-pirmoji diena
14     fd >> m1 >> m >> n;
15     fd.close();
16
17     d=1;
18     s=m1;
19     while (m1<n) {
20         d=d+1;
21         s=s+m; // kiek ta diena isgenejo medziu
22         m1=s+m1; // kiek isviso isgenejo medziu
23     }
24     fr << "Sodininkas dirbs " << d << " dienas" << endl;
25     fr.close();
26
27     return 0;
28 }
```

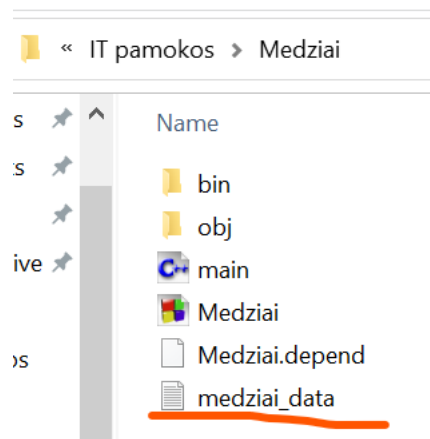
Pav. 1. Pradinių duomenų skaitymo iš failo programos užrašymas.

42. Sukuriame pradinių duomenų failą, tam, kad mes galėtume vykdyti programą:
File → **New** → **Empty file**.



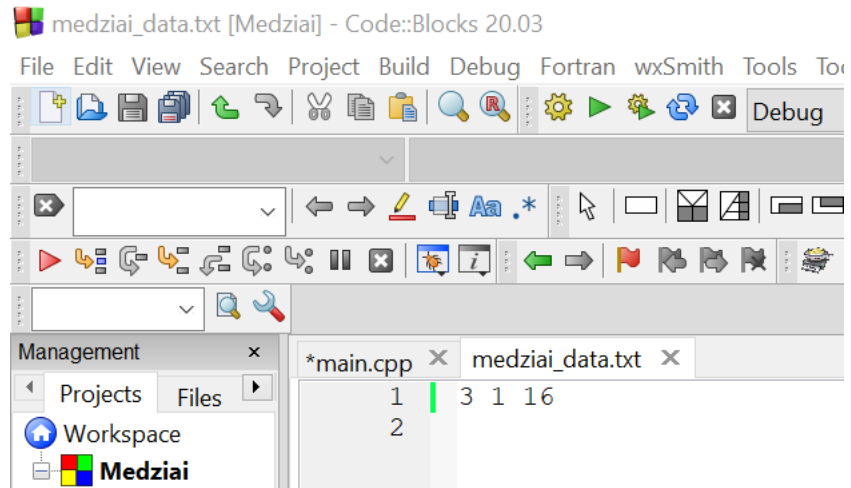
Pav. 2. Pradinių duomenų failo kūrimas.

43. Pradinių duomenų failas išsaugomas tame pačiame kataloge, kur yra išsaugotas užduoties failas. Užrašome pradinių duomenų failo pavadinimą: **medziai_data.txt**



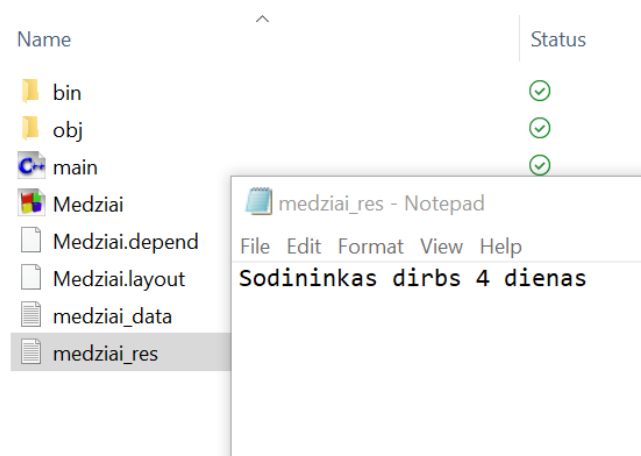
Pav. 3. Pradinių duomenų failo išsaugojimas.

44. Duomenys turi būti surašyti eilės tvarka, kaip yra aprašyti pradiniai duomenys programoje.



Pav. 4. Pradinių duomenų užrašymas faile.

45. Paleidžiant „Vykdyti“ rezultatas rodomas faile.



Pav. 5. Rezultatų išvedimas į failą.

2 ETAPAS (23-25 minutės). Praktinė pamokos dalis.

Užduočių atlikimas remiantis pavyzdžiu. Užduočių diferencijavimas.

Cikliniai algoritmai

- *Uždaviniai silpniems mokiniams.*
5. Nurašyti programą į Code::blocks aplinką. Pakeisti pradinis duomenis ir išveskite rezultatą į failą.
 6. **Kelias į mokyklą.** Kiekvieną dieną Petriukas, eidamas į mokyklą, skaičiuoja kiekvieną savo žingsnį ir žaidžia tokį žaidimą: kai žingsnių skaičius baigiasi nuliu, Petriukas suploja

rankomis, o kai penketu - spragтели pirštais. Parašykite programą, kuri suskaičiuotų, kiek kartų Petriukas suplos rankomis ir kiek – spragtelės pirštais, jei jam iki mokyklos yra lygiai n žingsnių.

(Žinomo kartojimų skaičiaus ciklas FOR)

Pradiniai duomenys, skaitomi iš failo	Rezultatai, rodomi ekrane
15 _____ _____ Paaškinimas: <i>Žingsnių kiekis iki mokyklos: 15.</i>	1 2 _____ _____ Paaškinimas: <i>Suplojo rankomis: 1. Spragtelėjimų bus: 2.</i>

• **Vidutinio sunkumo uždaviniai.**

5. **Snaigės už lango.** Per atostogas Simas turėjo daugiau laisvo laiko ir nutarė suskaičiuoti, kiek sningant po jo namo langu nukrenta snaigių. Jis pastebėjo, kad kiekvieną kitą sekundę nukrenta dvigubai daugiau snaigių, nei prieš tai buvusią. Parašykite programą, skaičiuojančią kiek snaigių s bus nukritę per n sekundžių, kai per pirmąją sekundę nukrito k snaigių.

(Žinomo kartojimų skaičiaus ciklas FOR)

Pradiniai duomenys, skaitomi iš failo	Rezultatai, rodomi ekrane
5 3 _____ _____ Paaškinimas: <i>Kiek snaigių nukrito per pirmąją sekundę ir kiek sekundžių snigo.</i>	35 _____ _____ Paaškinimas: <i>Nukrito snaigių.</i>
2 4 _____ _____ Paaškinimas: <i>Kiek snaigių nukrito per pirmąją sekundę ir kiek sekundžių snigo.</i>	30 _____ _____ Paaškinimas: <i>Nukrito snaigių.</i>

6. **Draugų skaičiai.** 10 draugų sugalvojo po vieną skaičių iki 100 ir juos užrašė ant kortelių. Parašykite programą, surandančią keli draugai sugalvojo lyginius skaičius. Jeigu tokių skaičių bus, ekrane turi būti rodomas jų kiekis, priešingu atveju ekrane turi būti rodomas žodis Nėra.

(Žinomo kartojimų skaičiaus ciklas FOR)

Pradiniai duomenys, skaitomi iš failo	Rezultatai, rodomi ekrane
---------------------------------------	---------------------------

2 9 4 100 25 5 6 3 5 85 <hr/> Paaiškinimas: <i>Draugų sugalvoti skaičiai.</i>	4 <hr/> Paaiškinimas: <i>4 draugai sugalvojo lyginius skaičius.</i>
5 3 99 55 35 47 11 63 51 91 <hr/> Paaiškinimas: <i>Draugų sugalvoti skaičiai.</i>	Nėra

- **Uždaviniai pažengusiems mokiniams.**

5. **Siuntos.** Siuntų pervežimo įmonėje dirbantis kurjeris gavo užduotį parengti pervežimų statistiką:

- ♣ kiek įvykdė užsakymų, kurių suma viršijo 100 eurų;
- ♣ už kokią vidutinę sumą per dieną išvežiojo prekių;
- ♣ kiek iš viso prekių išvežiojo.

Kurjeris nežino, kiek užsakymų įvykdys per dieną, todėl baigęs darbą į programą įves nulį (0), tai reikš, kad darbo diena baigta ir reikia pateikti rezultatus. Parenkite programą, kuri leistų nežinomą skaičių kartų kurjeriui įvesti užsakymo sumą (skaičiavimai baigiami įvedus nulį, nulis skaičiuojant vidurkį ir kiekį nebus pridėtas) ir pateiktų skaičiavimų rezultatus.

(Nežinomo kartojimų skaičiaus ciklas WHILE)

Pradiniai duomenys, skaitomi iš failo	Rezultatai, rodomi ekrane
110 80 50 0 <hr/> Paaiškinimas: <i>Sumos įvedimas.</i>	1 80 3

6. **Skaitytojas.** Tadas mėgsta skaityti knygas, tačiau jam labai sunku pradėti skaityti. Knygoje yra m skyrių Pirmą dieną Tadas perskaitė 1 skyrių, antrą – 2, trečią – 3 ir t.t. Kiekvieną kitą dieną jis perskaito vienu skyriumi daugiau, negu prieš tai buvusią dieną. Programa turi apskaičiuoti, per kelias dienas d Tadas perskaitys visą knygą ir kelis skyrius s vidutiniškai per dieną perskaito Tadas. Paskutinei dienai gali likti mažiau skyrių.

(Nežinomo kartojimų skaičiaus ciklas WHILE)

Pradiniai duomenys, skaitomi iš failo	Rezultatai, rodomi ekrane
8	4 2 <hr/>

<p>_____</p> <p>Paaškinimas: <i>Knygų skyrių skaičius.</i></p>	<p>Paaškinimas: <i>Tadas visą knygą perskaitys per 4 dienas(-ų).</i> <i>Tadas vidutiniškai per dieną perskaitė 2 skyrius (-ų).</i></p>
<p>17</p> <p>_____</p> <p>Paaškinimas: <i>Knygų skyrių skaičius.</i></p>	<p>6 2.83</p> <p>_____</p> <p>Paaškinimas: <i>Tadas visą knygą perskaitys per 6 dienas(-ų).</i> <i>Tadas vidutiniškai per dieną perskaitė 2.83 skyrius (-ų).</i></p>

3 ETAPAS 🕒 5-7 minučių

Su mokiniais apibendrinamas atliktas darbas.

6. Programų testavimo aplinka. Jeigu mokykla pasirenka testavimo aplinką, mokiniai įkelia savo sudarytas programas ir mokytojas mato jų pažangą.

7. Jei mokiniai dirbo pagal pamokos scenarijų, galima jų paprašyti įvertinti kiekvieną žingsnį užpildant mokytojo parengtą apklausą (galima naudoti Google Forms).

Galimi klausimai apklausoje:

Ką išmokau, atlikau?

Ką pavyko atlikti gerai?

Kas rodo, kad atlikau gerai?

Ką pavyko atlikti savarankiškai?

Kuo labiausiai džiaugiuosi?

Kokių sunkumų kilo?

Kokius savo neteisingus veiksmus ar klaidas pastebėjau?

Kaip įveikiau kilusius sunkumus?

Kieno ir kokios pagalbos prireikė?

Ką kitą kartą galima būtų daryti kitaip?

8. Jei mokiniai patys išsikėlė uždavinius ir juos išsprendė, tuomet trumpai pristato atliktus darbus.

Žinomo kartojimų skaičiaus ciklas (Deividas Urbonas)

Pasiekimų sritis	Algoritmai ir programavimas (B1)
Klasė	9
Tema	Žinomo kartojimų skaičiaus ciklas

Integruojami dalykai, pasiekimai	Informatika: Programavimas. Matematika: Loginio mąstymo vystymas. Anglų kalba. lietuvių kalba.	
Kompetencijos	Skaitmeninė kompetencija – naujų skaitmeninių įrankių pažinimas. Pažinimo kompetencija – nauja skaitmeninio raštingumo taikymo sritis. DI kompetencijos – dirbtinio intelekto įrankiai gali labai padėti pradedant vystyti programavimo įgūdžius. kūrybiškumo kompetencijos – programavimas leidžia įgyvendinti savo sumanymą. Komunikavimo kompetencijos – pagalba draugams, atliktų darbų pristatymas, kolektyvinė sprendimo paieška.	
Tikslas	Išmokti kas yra „for“ ciklo algoritmai, kaip jie yra kuriami, kam naudojami, sugebėti juos kurti savarankiškai.	
Uždaviniai	<ol style="list-style-type: none"> 1. Susipažinti su „for“ ciklo algoritmais 2. Įsisavinti kurti „for“ ciklo algoritmus 3. Pristatyti savo sukurtus algoritmus 	
Planuojamas rezultatas	<ol style="list-style-type: none"> 1. Mokiniai gebės kurti „for“ ciklo algoritmus 2. Gebės pritaikyti “for” ciklą programavime 3. Gebės pristatyti savo darbo rezultatus 	
Specifinės priemonės / programinė įranga	<ol style="list-style-type: none"> 1. Kompiuteris 2. Interneto prieiga 3. Vaizdo įrašai, skirti peržiūrėti namuose prieš pamoką, taikant apverstos klasės metodą: https://www.youtube.com/watch?v=8jLOx1hD3_o&ab_channel=freCodeCamp.org (nuo 07:55:20 iki 08:25:20) 4. C++ programavimo aplinka: https://www.onlinegdb.com/online_c++_compiler 5. Mokomoji medžiaga apie „For“ ciklus 	
Mokymosi metodai	<ol style="list-style-type: none"> 1. Apversta klasė (Teorinė medžiaga apie “for” ciklą, youtube video su aplinkos paleidimu) 2. Praktinis darbas 3. Individualus darbas. Mokiniai turės savarankiškai išspręsti uždavinius, parodydami įgautas žinias. 	
Mokinių atlikto darbo vertinimas ir įsivertinimas	Įvertinimas (balas)	Vertinimo kriterijai
	10 (puikiai)	Abiejų pamokų pradžioje mokinys atsakinėjo į klausimus; Atliekant užduotį porose aktyviai dalyvavo procese, užduotys atliktos teisingai;

		<p>Užduoties porose aptarimo metu sugebėjo pagrįsti ir paaiškinti kaip atliko užduotį; Namų darbas atliktas teisingai, mokinys gali paaiškinti ir pagrįsti savo sprendimą; Antros pamokos individualus darbas atliktas savarankiškai, geba paaiškinti savo sprendimą; Aktyviai dalyvauja diskusijoje aptarinėjant kitų mokinių darbus</p>
	9 (labai gerai)	<p>Abiejų pamokų pradžioje mokinys atsakinėjo į klausimus; Atliekant užduotį porose aktyviai dalyvavo procese, užduotys atliktos teisingai; Užduoties porose aptarimo metu sugebėjo pagrįsti ir paaiškinti kaip atliko užduotį; Namų darbas atliktas teisingai, mokinys gali paaiškinti ir pagrįsti savo sprendimą; Antros pamokos individualus darbas atliktas savarankiškai, geba paaiškinti savo sprendimą;</p>
	8 (gerai)	<p>Atliekant užduotį porose aktyviai dalyvavo procese, užduotys atliktos teisingai; Užduoties porose aptarimo metu sugebėjo pagrįsti ir paaiškinti kaip atliko užduotį; Namų darbas atliktas teisingai, mokinys gali paaiškinti ir pagrįsti savo sprendimą; Antros pamokos individualus darbas atliktas savarankiškai, geba paaiškinti savo sprendimą;</p>
	7 (vidutiniškai)	<p>Atliekant užduotį porose silpnai dalyvavo procese, užduotys atliktos teisingai; Užduoties porose aptarimo metu sugebėjo pagrįsti ir paaiškinti kaip atliko užduotį; Namų darbas atliktas teisingai, mokinys gali paaiškinti ir pagrįsti savo sprendimą; Antros pamokos individualus darbas atliktas savarankiškai, geba paaiškinti savo sprendimą;</p>
	6 (patenkinamai)	<p>Atliekant užduotį porose silpnai dalyvavo procese, užduotys atliktos teisingai; Antros pamokos individualus darbas atliktas savarankiškai, geba paaiškinti savo sprendimą;</p>
	5 (silpnai)	<p>Atliekant užduotį porose silpnai dalyvavo procese, užduotys atliktos teisingai; Antros pamokos individualus darbas atliktas su mokytojo pagalba;</p>

Žinios prieš	Mokėti naudotis C++ programavimo aplinka, būti susipažinus su C++ programavimo kalbos sintakse
Galimybės taikyti spec. poreikių mokiniams	Pagal poreikį pamoka gali būti vedama nuotoliniu būdu, galima užduotį porose pakeisti į lengvesnę individualią užduotį. Mokymosi medžiagą galima skaidyti į didesnę pamokų skaičių. Vertinimas gali būti rašomas pagal atlikto namų darbo pasiekimus.
Patarimai kolegoms, kurie naudos parengtą medžiagą	Prieš pamoką susipažinti su medžiaga, pateiktą savarankiškam susipažinimui. Prieš pamoką pačiam mokytojui atlikti visus duodamas užduotis, susipažinti su skirtingais sprendimo būdais. Pamokos metu mokiniai dirba savarankiškai, mokytojas yra pagalbininkas/konsultantas.

Detalus scenarijus

Pirma pamoka:

1 Etapas. 45 minutės

1. Žinių patikrinimas pamokos pradžioje. Rekomendacinio pobūdžio pavyzdžiai mokytojui. Klausimus galima keisti priklausant nuo mokinių gebėjimų, galima kurti klausimų banką.

1.1 Kas yra „for“ ciklo struktūra?

1.2 Apibūdinkite „for“ ciklo struktūrą savo žodžiais (kokią funkciją atlieka kiekviena dalis: inicializacija, sąlyga ir iteracija?).

1.3 Kokiose situacijose „for“ ciklai gali būti naudingi?

1.4 Kaip „for“ ciklas padeda automatizuoti veiksmus? Pateikite pavyzdį, kur tai galėtų būti naudinga realiame gyvenime.

1.5 Kaip manote, ar galima sukurti „for“ ciklą, kuris niekada nesustoja? Ką reikėtų padaryti, kad to išvengtumėte?

1.6 Kaip „for“ ciklas galėtų padėti išvengti klaidų, jei reikėtų atlikti tą patį veiksmą daugelį kartų rankiniu būdu?

2. Pamokos pradžia. Diskusija, pagrindinių sąvokų aptarimas ir „For“ ciklo pavyzdžių pateikimas skaidrėse su detaliu paaiškinimu.

2.1 **Trumpa diskusija:** Paprašyti mokinių trumpai paaiškinti, ką išmoko iš vaizdo įrašų ir kokie klausimai jiems kilo. Kiekvienas mokinytis gali pasidalinti savo atsakymais, pavyzdžiui:

2.1.1 Kas yra „for“ ciklo struktūra?

2.1.2 Kokiose situacijose „for“ ciklai gali būti naudingi?

2.2 Skaidrės su detaliu išaiškinimu:

2.2.1 Trumpai pakartojama “for” ciklo struktūra:

```
for (int i = 0; i < 10; i++) {  
    // Veiksmai, kurie bus atliekami 10 kartų  
}
```

2.2.2 **Pavyzdys klasėje:** Parodyti paprastą uždavinį, pavyzdžiui, „for“ ciklą, kuris išveda skaičius nuo 1 iki 5 ir detalai paaiškinti šią kodo dalį:

```
for (int i = 1; i <= 5; i++) {  
    cout << "Skaičius: " << i << endl;  
}
```

2.2.3 **Pavyzdys klasėje:** Parodyti kiek sunkesnę uždavinį, kuriame apskaičiuojama visų skaičių nuo 1 iki 100 suma ir išvedama į ekraną. Taip pat detalai paaiškinamas kodas:

```
int suma = 0; // Inicializuojame kintamąjį sumai saugoti  
  
// „for“ ciklas, kuris eina nuo 1 iki 100  
for (int i = 1; i <= 100; i++) {  
    suma += i; // Pridedame kiekvieną skaičių prie sumos  
}  
  
// Atspausdiname galutinę sumą  
cout << "Skaičių nuo 1 iki 100 suma: " << suma << endl;
```

3. Mokiniai suskirstomi poromis, ir kiekvienai porai yra skiriamos užduotys sukurti “for” ciklą pagal pateiktus reikalavimus:

3.1 **Užduotis 1:** Parašyti „for“ ciklą, kuris atspausdina lyginius skaičius nuo 2 iki 20.

3.2 **Užduotis 2:** Parašyti „for“ ciklą, kuris sudeda skaičius nuo 1 iki 10 ir pateikia atsakymą.

3.3 **Užduotis 3 (pažengusiems):** Parašyti „for“ ciklą, kuris išveda visus skaičius nuo 1 iki 10, bet praleidžia tuos, kurie dalijasi iš 3.

4. Aptarimas ir analizė: Kiekviena pora paaiškina savo sprendimą. Aptariame dažniausias klaidas ir kaip jas ištaisyti.

5. Namų darbai:

5.1 **Užduotis:** Parašyti programą, kuri skaičiuoja ir atspausdina visų nelyginių skaičių nuo 1 iki 20 sumą (parengia mokinius pamokai nr. 2).

Antra pamoka:

1 Etapas. 35 minutės

1. Prisimename kas buvo išmokta praeitą pamoką:

1.1 Užduodami keli klausimai:

- 1.1.1 Kas prisimena pagrindines „for“ ciklo dalis ir gali paaiškinti, ką reiškia kiekviena iš jų (inicializacija, sąlyga, iteracija)?
- 1.1.2 Kaip pasikeistų „for“ ciklo veikimas, jei padidintume žingsnį daugiau nei 1 (pvz., $i += 2$ vietoje $i++$)? Ar kas nors išbandė tokią iteraciją namų darbuose?
- 1.1.3 Ar kas nors gali paaiškinti, kodėl „for“ cikle svarbu, kad sąlyga būtų teisinga, ir kas nutinka, jei ji neteisinga?
- 1.1.4 Kodėl galėtų būti naudinga turėti kintamąjį (pvz., suma) už ciklo ribų, kai norime suskaičiuoti reikšmes ciklo viduje?

2. Diskusija apie namų darbą. Mokiniai pristato atliktą namų darbą, paaiškina panaudotą sprendimą, atsakymai į kilusius klausimus.

3. Individuali praktika – „for“ ciklo pritaikymas:

3.1 Individuali užduotis: Paprašyti kiekvieno mokinio parašyti programą, kuri naudoja „for“ ciklą tam tikrai problemai spręsti. Pavyzdžiui:

- 3.1.1 Parašyti programa, kuri atspausdina daugybos lentelę nuo 1 iki 10.
- 3.1.2 Parašyti programa, kuri skaičiuotų skaičių nuo 1 iki 50 sumą.
- 3.1.3 Parašyti programą, kuri atspausdina skaičių kvadratus nuo 1 iki 10.
- 3.1.4 Parašyti programą, kuri apskaičiuoja ir atspausdina visų nelyginių skaičių nuo 1 iki 100 sumą.
- 3.1.5 Parašyti programą, kuri atspausdina skaičius nuo 10 iki 1 atvirkštine tvarka.

2 Etapas. 10 minučių

Su mokiniais apibendrinamas atliktas darbas, mokiniai pristato savo atliktus darbus, paaiškina panaudotą sprendimą. Pateikiami atsakymai į visus iškilusius klausimus. Vyksta diskusija bei užduodami klausimai mokiniams žinių įtvirtinimui ir savęs įsivertinimui. Galimi klausimų pavyzdžiai:

A: Kas šiame uždavinyje buvo lengviausia?

B: Kokie aspektai vis dar sunkiai suprantami?

C: Kur ateityje, jų manymu, „for“ ciklai gali būti naudingi?

Paprogramių svarba programavime (Rasa Vilimienė-Jurkė)

Pasiekimų sritis	B. Algoritmai ir programavimas. B3. Kuria ir vykdo algoritmus, programas.
Klasė	9-10
Tema	Paprogramių svarba programavime.
Integruojami dalykai, pasiekimai	Matematika, užsienio kalbos (anglų k. ir kt.), technologijos.
Kompetencijos	Pažinimo – (mokėjimas mokytis) susipažinimas su paprogramėmis, jų sintakse. Skaitmeninė – (skaitmeninis turinys, problemų sprendimas) uždavinio sprendimas, programos kūrimas. Komunikavimo – bendradarbiaujama grupėje, diskutuojama klasėje.
Tikslas	Susipažinti su paprogramėmis, jų svarba programavime.
Uždaviniai	1. Savarankiškai susipažinti su paprogramėmis, pasinaudojant iš anksto pateikta medžiaga bei surandant apie ją informacijos įvairiuose kituose informacijos šaltiniuose. 2. Suprasti paprogramių svarbą programavime. 3. Prisiminti stačiakampio ploto, trikampio perimetro skaičiavimus. 4. Išmokti skaidyti algoritmus į smulkesnes, savarankiškas ir užbaigtas dalis. 5. Gebėti taikyti paprogrames uždavinyje, taisyklingai jas užrašyti.
Planuojamas rezultatas	1. Gebėsite suprasti kas yra paprogramės. 2. Suprasite paprogramių svarbą programavime. 3. Prisiminsite įvairių figūrų plotų, trikampio perimetro apskaičiavimus. 4. Sužinosite paprogramės sintaksę ir ją pritaikysite kurdami programą.
Specifinės priemonės programinė įranga	Projektorius arba išmanioji lenta. Kompiuteriai. Padalomoji medžiaga (spausdinta arba skaitmeninė). Popieriaus lapai. Rašymo priemonės. Skaičiuotuvai. Programa C++ (CodeBlock ar kt.). Informacijos šaltiniai: <ul style="list-style-type: none">Jonas Blonskis, Vytautas Bukšnaitis, Renata Burbaitė. Šiuolaikiškas žvilgsnis į programavimo pagrindus. Pasirenkamasis informacinių technologijų kursas IX-X klasėms. TEV, 2010.Renata Burbaitė, Jonas Blonskis, Vytautas Bukšnaitis. Šiuolaikiškas žvilgsnis į programavimą C++. Pasirenkamasis informacinių technologijų kursas XI-XII klasėms. TEV, 2011.

	<ul style="list-style-type: none"> • https://www.youtube.com/watch?v=-cPXx1GvNdE&list=PLKofrzwSXUn3lAXlTpQsZpVHODKELj5tA&index=58 • https://www.youtube.com/watch?v=SW-XlpJVxCA • https://it.senamiescio-g.lt/Failai/2kl/15temaFunkcijos.pdf
Mokymosi metodai	„Apverstos“ pamokos metodas. Diskusija. Individualus darbas. Darbas grupėse. Praktinis tyrimas.
Mokinių atlikto darbo vertinimas ir įsivertinimas	<p>Slenkstinis pasiekimų lygis – nagrinėja programų su paprogramėmis pavyzdžius, nurodo jų ryšius, padedamas naudoja parašytas paprogrames.</p> <p>Patenkinamas pasiekimų lygis – detalizuoja pateiktą uždavinį, nurodo reikalingus žingsnius jam spręsti, atpažįsta galimas paprogrames, padedamas rašo paprogrames ir jas naudoja.</p> <p>Pagrindinis pasiekimų lygis – projektuoja programą, naudoja paprogrames.</p> <p>Aukštesnysis lygis – tinkamai aprašo, komentuoja paprogramių veikimą, kuria paprogrames, kurios grąžina rezultatus.</p>
Žinios prieš	Geba surasti reikiamą informaciją įvairiuose informacijos šaltiniuose, tinkamai naudojasi programine ir aparatine kompiuterio įranga, turi pagrindinius darbo su programavimo kalba C++ įgūdžius.
Galimybės taikyti spec. poreikių mokiniams	Mokytojas pamokos scenarijų pritaiko priklausomai nuo vaiko specialiųjų poreikių. Specialiųjų poreikių mokiniams galima pateikti sukurtos programos šabloną, kurioje jie minimaliai turėtų kažką pakeisti, priklausomai nuo mokinio sutrikimo atvejo.
Patarimai kolegoms, kurie naudos parengtą medžiagą	<ul style="list-style-type: none"> • Galima parinkti kitus informacijos šaltinius, nuorodas. • I etape galima pateikti mažesnę figūrų skaičių. • II etape, priklausomai nuo mokinių skaičiaus, galima suskirstyti į kelias grupes. • III etape galimi kitokie programos kūrimo variantai. Viskas priklauso nuo mokinių, kiek jie pažengę programavimo srityje. Jei matote, kad nespės sukurti ir ištestuoti programų, darbą galite pratęsti sekančią pamoką arba užduoti namų darbams užbaigti programą ir ištestuoti. Jei mokiniams sunkiai sekasi rašyti programas, galima šį pamokos scenarijų skaidyti į 2 pamokas: pirmą pamoką spręsti II variantą (lengvesnį), o sekančią pamoką kurti I variantą.

Prieš pamoką. Mokiniam siūloma medžiaga darbui namuose:

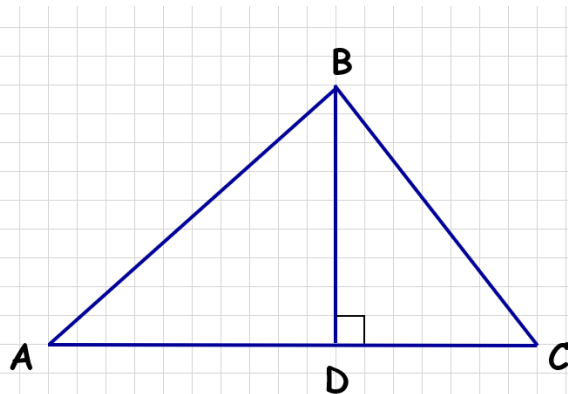
- Jonas Blonskis, Vytautas Bukšnaitis, Renata Burbaitė. Šiuolaikiškas žvilgsnis į programavimo pagrindus. Pasirenkamasis informacinių technologijų kursas IX-X klasėms. TEV, 2010. (109 psl. – Funkcijos).
- Renata Burbaitė, Jonas Blonskis, Vytautas Bukšnaitis. Šiuolaikiškas žvilgsnis į programavimą C++. Pasirenkamasis informacinių technologijų kursas XI-XII klasėms. TEV, 2011. (149-150 psl. – Funkcijos).
- <https://www.youtube.com/watch?v=cPXx1GvNdE&list=PLKofrzwSXUn3lAX1TpQsZpVHODKELj5tA&index=58>
- <https://www.youtube.com/watch?v=SW-XlpJVxCA>
- <https://it.senamiescio-g.lt/Failai/2kl/15temaFunkcijos.pdf> (plačiau).

1 ETAPAS ⌚ (5 min.)

Mokiniam išdalinti padalomą medžiagą. Prisiminti, kaip apskaičiuojamas plotas. Užduotis – apskaičiuoti visų kambarių plotus, namo plotą, šalia namo esančių pastatų plotus. Užduotį daro kiekvienas individualiai. Galima naudotis skaičiuotuvais. Maždaug po 5 min. darbas stabdomas.

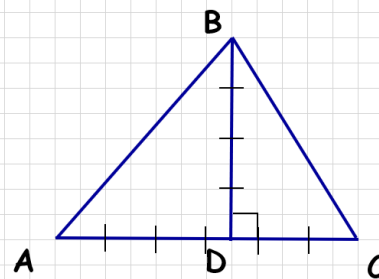
Plotų (S) apskaičiavimo prisiminimas:

- Stačiakampio (lygiagretainio) ploto formulė: $S = a \cdot b$
- Kvadrato ploto formulė: $S = a \cdot a = a^2$.
- Trikampio ploto formulė yra $S = 1/2bh$. Kur S - plotas, b – pagrindo ilgis, h – aukštinė.



Trikampio plotas

$$S = \frac{AC \cdot BD}{2}$$



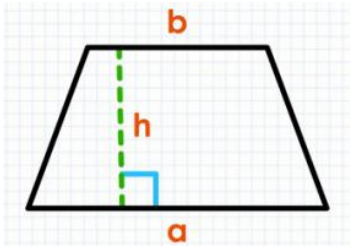
Pagrindo ilgis AC = 6 cm

Aukštinės ilgis BD = 4 cm

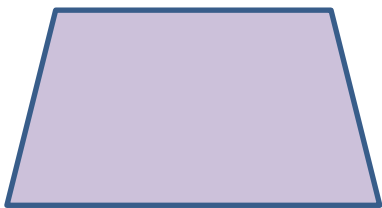
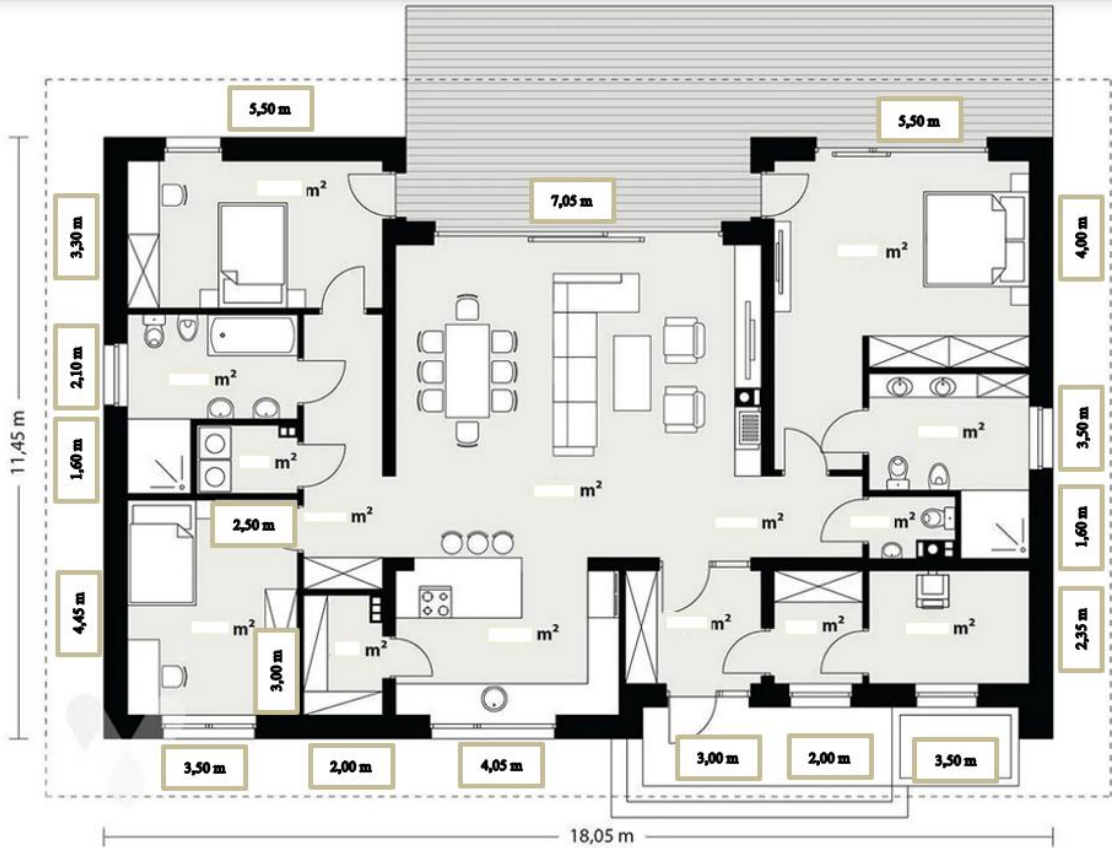
Trikampio plotas

$$S = \frac{AC \cdot BD}{2} = \frac{6 \cdot 4}{2} = 12$$

- Trapecijos ploto formulė: $S = (a+b)/2 \cdot h$. Kur a, b – pagrindai, h - aukštinė.



- Skritulio ploto formulė $S = \pi R^2$.



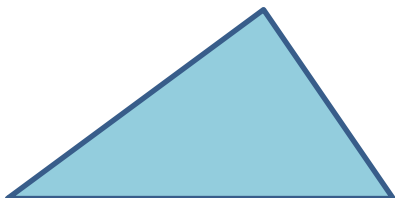
Trapecijos formos pastatai:

1. a=4 m, b=5 m, h=2,5 m.
2. a=4,5 m, b=5,5 m, h=2,7 m.
3. a=2,1 m, b=3,1 m, h=4,2 m.



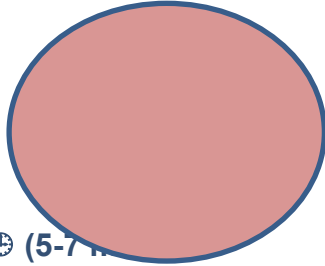
Kvadrato formos pastatai:

1. a=4,4 m.
2. a=5,6 m.
3. a=3,3 m.
4. a=2,7 m.



Trikampio formos pastatai:

1. pagrindas=4,1 m, aukštinė=3,2 m.
2. pagrindas=1,7 m, aukštinė=2,6 m.
3. pagrindas=2,8 m, aukštinė=3 m.



2 ETAPAS ⌚ (5-7 min.)

Skritulio formos pastatai:

1. $R=2,7$ m.
2. $R=3,4$ m.
3. $R=1,7$ m.

Mokinių klausama, kaip būtų galima greičiau suskaičiuoti visų kambarių plotus, kai dirbama komandoje? Mokiniai pasiskirsto kambarius, kurių plotus skaičiuos ir bendrai komandoje užbaigia darbą. Mokiniai bendradarbiauja tarpusavyje, konsultuojasi su mokytoju.

3 ETAPAS ⌚ (5-7 min.)

PAPROGRAMĖS (funkcijos).

Aptariama I užduotis. Darbų pasiskirstymas prilyginamas paprogramėms programoje.

Diskusija apie paprogrames, ką apie jas sužinojo studijuodami informacijos šaltinius.

Paprogramių pristatymas. Teorinė medžiaga.

- Paprogramė – tai struktūrinis elementas, kuris padeda geriau struktūrizuoti programas (padalyti programos tekstą į mažus lengvai suprantamus gabalėlius), padaro jas lengviau skaitomas ir analizuojamas.
- Paprogramė – savarankiška programos dalis, vykdoma iškvietus ją kreipiniu.
- Į vieną aprašytą paprogramę galima kreiptis daug kartų ir daug kartų atlikti jos veiksmus.
- Vienoje programoje gali būti daug paprogramių.
- C++ kalboje paprogramės yra vadinamos funkcijomis.
- Paprogramės atliekamas veiksmas turi būti aiškus, apibrėžtas ir nepriklausantis nuo kitų veiksmų.

Paprogramės sintaksė. Susideda iš 2 dalių:

Prototipas (išankstinis
aprašas)

Funkcijos realizavimo
aprašymas

*rezultatoTipas funkcijosVardas
(formaliųjų parametru
sąrašas);*

*rezultatoTipas funkcijosVardas
(formaliųjų parametru
sąrašas){
funkcijosKamienas
return ...;
}*

Galima ir kitaip įvardinti paprogramės (funkcijos) dalis.

```
<tipas> <pavadinimas> (<argumentai / parametrai>
{
<kodas>
return <reikšmė>;
```

```
rezultato_tipas funkcijos_vardas (argumentų aprašai)
{
sakinys_1;
sakinys_2;
.....
return rezultatas;
```

Paprogramės (funkcijos) pavyzdys:

```
double StTrikPlotas (double a, double b)
{
double S = a * b / 2 // plotas
return S; // grąžiname plotą
```

4 ETAPAS 🕒 (5 min.)

Teorijos įtvirtinimui.

<https://learningapps.org/31919295>



5 ETAPAS 🕒 (15-20 min.)

I variantas. Trikampio perimetro (pagal kampų koordinates) apskaičiavimas. Programa kuriama C++ programavimo kalba.

I variantas.

```
#include <iostream>
#include <cmath>

using namespace std;

int main()
{
    double xa, ya, xb, yb, xc, yc;

    // Įveskite koordinates taškams
    cout << "Taškų A, B, C koordinates: ";
    cin >> xa >> ya >> xb >> yb >> xc >> yc;

    // Skaičiuojame atstumus tarp taškų
    double ab = sqrt(pow(xa - xb, 2) + pow(ya - yb, 2));
    double ac = sqrt(pow(xa - xc, 2) + pow(ya - yc, 2));
    double bc = sqrt(pow(xc - xb, 2) + pow(yc - yb, 2));

    // Skaičiuojame trikampio perimetrą
```

```

double perimetras = ab + ac + bc;

// Spausdiname rezultata
cout << "Trikampio perimetras: " << perimetras << endl;

return 0;
}

```

Programa su paprograme.

```

#include <iostream>
#include <cmath>

using namespace std;
// Funkcija, kuri skaičiuoja atstumą tarp dviejų taškų (x1, y1)
ir (x2, y2)
double Atstumas(double x1, double y1, double x2, double y2);

int main()
{
    double xa, ya, xb, yb, xc, yc;

    // Įveskite koordinates taškams
    cout << "Taškų A, B, C koordinatės: ";
    cin >> xa >> ya >> xb >> yb >> xc >> yc;

    double ab = Atstumas(xa, ya, xb, yb);
    double ac = Atstumas(xa, ya, xc, yc);
    double bc = Atstumas(xb, yb, xc, yc);

    // Skaičiuojame trikampio perimetra
    double perimetras = ab + ac + bc;

    // Spausdiname rezultata
    cout << "Trikampio perimetras: " << perimetras << endl;

    return 0;
}

double Atstumas(double x1, double y1, double x2, double y2) {
    double ats = sqrt(pow(x1 - x2, 2) + pow(y1 - y2, 2));
    return ats;
}

```

II variantas. (Lengvesnis).

```
// Stačiakampio plotas
#include <iostream>
using namespace std;

int Plotas(int a, int b); // funkcijos Plotas prototipas


int main()
{
    int x = 5, y = 4, s;
    s = Plotas(x, y); // kreipinys į funkciją Plotas
    cout << "Plotas = " << s << endl;
    return 0;
}

// Skaičiuoja stačiakampio, kurio kraštinės a ir b, plotą
int Plotas(int a, int b) // funkcijos antraštė
{
    return a * b; // per funkcijos vardą gražinama apskaičiuota
    ploto reikšmė
}
```

6 ETAPAS 🕒 (3-5 min)

Su mokiniais apibendrinamas atliktas darbas pamokoje.

Mokiniai savo mintimis (kaip sekėsi pamokoje, kas patiko, kas labiausiai pasisekė, kas buvo sunkiausia) pasidalina linoit lentoje (<https://en.linoit.com/>).

 Pasidalink
mintimis, kaip
sėkėsi šioje
pamokoje.

linor



Paprogramės, kai parametru perduodama reikšmė (Rasa Vilimienė-Jurkė)

Pasiekimų sritis	B. Algoritmai ir programavimas. B3. Kuria ir vykdo algoritmus, programas.
Klasė	9-10
Tema	Paprogramės, kai parametru perduodama reikšmė.
Integruojami dalykai, pasiekimai	Matematika, lietuvių k., užsienio kalbos (anglų k. ir kt.).
Kompetencijos	Pažinimo – (mokėjimas mokytis) informatikos žinių gilinimas, susipažinimas paprogramėmis, kai parametru perduodama reikšmė, jų sintakse. Skaitmeninė – (skaitmeninis turinys, problemų sprendimas) uždavinio sprendimas, programos kūrimas. Komunikavimo – bendradarbiaujama grupėse, diskutuojama. Socialinė, emocinė ir sveikos gyvensenos – dirbdami grupėmis mokiniai laikosi sutartų taisyklių, diskutuoja, argumentuoja savo veiksmus, komentuodami laikosi etikos principų, korektiškai vertina kitų darbą.
Tikslas	Susipažinti su paprogramėmis, kai parametru perduodama reikšmė.
Uždaviniai	1. Prisiminti paprogramių sąvoką, jų svarbą programavime. 2. Susipažinti su paprogramėmis, kai parametru perduodama reikšmė, jų sintakse. 3. Išmokti skaidyti algoritmus į smulkesnes, savarankiškas ir užbaigtas dalis. 4. Gebėti taikyti paprogrames, kai parametru perduodama reikšmė, uždavinyje, taisyklingai jas užrašyti.
Planuojamas rezultatas	1. Įvardinsite paprogramių sąvokas, jų svarbą programavime. 2. Gebėsite paaiškinti kas yra paprogramė, kai parametru perduodama reikšmė. 3. Sužinosite paprogramės, kai parametru perduodama reikšmė, sintaksę ir ją pritaikysite kurdami programą.
Specifinės priemonės / programinė įranga	Projektorius arba išmanioji lenta. Kompiuteriai. Popieriaus lapai. Rašymo priemonės. Refleksijai – „Šviesoforo“ įsivertinimui (atspausdinta arba skaitmeninė) priemonė. Programa C++ (CodeBlock ar kt.). Informacijos šaltiniai: <ul style="list-style-type: none">Jonas Blonskis, Vytautas Bukšnaitis, Renata Burbaitė. Šiuolaikiškas žvilgsnis į programavimo pagrindus. Pasirenkamasis informacinių technologijų kursas IX-X klasėms. TEV, 2010.Renata Burbaitė, Jonas Blonskis, Vytautas Bukšnaitis. Šiuolaikiškas žvilgsnis į programavimą C++. Pasirenkamasis informacinių technologijų kursas XI-XII klasėms. TEV, 2011.

	<ul style="list-style-type: none"> • Renata Burbaitė. Šiuolaikiškas žvilgsnis į programavimą C++. Pasirenkamasis informacinių technologijų kursas XI-XII klasėms. Mokytojo knyga. Atmintukas. TEV, 2011. • https://www.ugdome.lt/kompetencijos5-8/pagrindinis/kompetenciju-ugdymo-praktika/aktyvaus-mokymo-ir-mokymosi-metodai-ir-ju-taikymo-pavyzdziai/aktyvaus-mokymosi-metodai/durstinys-2/#title • Valentina Dagienė, Gintautas Grigas. Programavimo kalbų teoriniai pagrindai. 2007. • https://slideplayer.lt/slide/17914346/
Mokymosi metodai	Durstinio metodas. Diskusija. Uždavinio sprendimo metodas. Įsivertinimo metodas „Šviesoforas“.
Mokinių atlikto darbo vertinimas ir įsivertinimas	<p>Slenkstinis pasiekimų lygis – nagrinėja programų su paprogramėmis, kai parametru perduodama reikšmė, pavyzdžius, padedamas naudoja parašytas paprogrames.</p> <p>Patenkinamas pasiekimų lygis – detalizuoja pateiktą uždavinį, nurodo reikalingus žingsnius jam spręsti, atpažįsta galimas paprogrames, kai parametru perduodama reikšmė, padedamas rašo paprogrames ir jas naudoja.</p> <p>Pagrindinis pasiekimų lygis – projektuoja programą, naudoja paprogrames kai parametru perduodama reikšmė.</p> <p>Aukštesnysis lygis – tinkamai aprašo, komentuoja paprogramių, kai parametru perduodama reikšmė, veikimą, kuria paprogrames, kurios grąžina rezultatus.</p>
Žinios prieš	Geba surasti reikiamą informaciją įvairiuose informacijos šaltiniuose, tinkamai naudojami programine ir aparatine kompiuterio įranga, turi pagrindinius darbo su programavimo kalba C++ įgūdžius.
Galimybės taikyti spec. poreikių mokiniams	Mokytojas pamokos scenarijų pritaiko priklausomai nuo specialiųjų poreikių mokinio gebėjimų. Specialiųjų poreikių mokiniams, priklausomai nuo jų sutrikimo atvejo, galima pateikti sukurtos programos šabloną, kurioje jie minimaliai turėtų kažką pakeisti.
Patarimai kolegoms, kurie naudos parengtą medžiagą	<ul style="list-style-type: none"> • Patiems išsibandyti išspręsti uždavinius. • I etape galima parinkti kitus klausimus. • II etape medžiaga teorijos apibendrinimui gali būti ženkliai sumažinta, priklausomai, kaip paskirstomas pamokos laikas. • Jei klasėje mokiniai nespėja atlikti užduoties, ją gali užbaigti namuose.

1 ETAPAS ⌚ (10-15 min.)

Metodas durstinys. Mokiniai suskirstomi į grupes po 3-4. Tai „namų“ grupė, kurios dalyviai gauna užduotį – pagal pateiktą temą surasti informaciją, išsiaiškinti. Temos:

- Kas yra paprogramė? Paprogramių svarba programavime.
- Paprogramės (funkcijos) sintaksė (praėjusios pamokos tema – prisiminti pačiam ir suteikti informaciją nebuvusiams pamokoje).
- Paprogramės, kai parametru perduodama reikšmė.

Gavę temas, mokiniai persigrupuoja – susiskirstoma į „ekspertų“ grupes. Kiekvieną grupę sudaro tą pačią mokymosi medžiagos dalį turintys mokiniai. Jie kartu nagrinėja medžiagą ir planuoja, kaip mokys „namų“ grupės draugus, kaip pateiks informaciją, kad kiti ją suprastų. Mokiniai ieško informacijos vadovėliuose, internete. Mokytojas konsultuoja, stebi. Atlikę šią užduotį mokiniai grįžta į „namų“ grupes ir moko vieni kitus. Grupės tikslas – kad visi gerai išmoktų visą mokymosi medžiagą. Mokytojas, norėdamas išsiaiškinti, ar mokiniai suprato medžiagą, užduoda klausimus „namų“ grupės dalyviams. Mokiniai įvertina vieni kitus.

Grupės narys	Aiškino puikiai, viską supratau.	Nelabai aišku, supratau ne viską.	Aiškino blogai, beveik nieko nesupratau.

2 ETAPAS ⌚ (10-15 min.)

Paprogramės, kai parametru perduodama reikšmė, prototipas sintaksė. Susideda iš 2 dalių:

Prototipas (išankstinis aprašas)	Funkcijos realizavimo aprašymas
<pre>void funkcijosVardas (tipas & vardas1, tipas & vardas2);</pre>	<pre>void funkcijosVardas (tipas & vardas1, tipas & vardas2){ funkcijosKamienas }</pre>

Teorija:

Funkcija yra savarankiška programos dalis, kuri gali apskaičiuoti ir gražinti vieną rezultatą per funkcijos vardą, arba funkcija su parametrais-nuorodomis, apskaičiuojanti kelis rezultatus.

Jei funkcija grąžina apskaičiuotas reikšmes per parametrus (void), į ją kreipiamasi taip:

```
funkcijosVardas (faktiniaiParametrai);
```

Jei funkcija turi grąžinti keletą reikšmių, tuomet naudojami parametrai-nuorodos. Prieš juos funkcijos antraštėje rašomas ženklas &:

```
rezultatoTipas funkcijosVardas (tipas & vardas1, tipas & vardas2);
```

Tuo atveju, kai į funkciją kreipiamasi perduodant jai parametrus-reikšmes, funkcija sukuria naujus tų pačių tipų kintamuosius, kaip ir perduodami parametrai, ir jiems priskiria parametrų reikšmes. Vadinasi, funkcija dirba su parametrų reikšmių kopijomis, bet ne su pačiais parametrais. Tai patogu, kai funkcijai nereikia keisti parametrų reikšmių.

Tuo atveju, kai į funkciją kreipiamasi perduodant jai parametrus-nuorodas, ji gauna ne kintamųjų reikšmes, o nuorodas į kintamuosius (kintamųjų adresus). Vadinasi, funkcija tiesiogiai naudoja perduodamus kintamuosius.

Paprogramės antraštėje esantys parametrai vadinami formaliaisiais parametrais. Parametrai, esantys kreipinyje, vadinami faktiniais parametrais. Kreipimosi į procedūrą metu faktiniai parametrai susiejami su formaliaisiais.

Daugiau informacijos rasite pateiktuose informacijos šaltiniuose.

3 ETAPAS 🕒 (7-10 min.)

Nagrinėjamas pavyzdys, kuriame funkcija grąžina trikampio plotą ir perimetrą, kai žinomos trikampio kraštinės.

```
#include <iostream>
#include <cmath>

using namespace std;

void apskaiciuotiTrikampio(double a, double b, double c, double
& plotas, double & perimetras, double & pusp);

int main() {

    double a, b, c, plotas, perimetras, s;

    cout << "Įveskite trikampio kraštines (a, b, c): ";
    cin >> a >> b >> c;

    apskaiciuotiTrikampio(a, b, c, plotas, perimetras, s);

    cout << "Trikampio plotas: " << plotas << endl;
    cout << "Trikampio perimetras: " << perimetras << endl;

    return 0;
```

```

}

void apskaiciuotiTrikampio(double a, double b, double c, double
& plotas, double & perimetras, double & pusp) {
    perimetras = a + b + c;
    pusp = perimetras / 2;
    plotas = sqrt(pusp * (pusp - a) * (pusp - b) * (pusp - c));
}

```

4 ETAPAS 🕒 (10-15 min.)

Uždavinių sprendimas.

Patenkinamas lygis.

Užduotis. Parašyti programą, kurioje gražinama dvigubo skaičiaus reikšmė.

```

#include <iostream>

using namespace std;

// Funkcija, kuri padvigubina gautą skaičių, naudodama nuorodą
void padvigubinti(int &skaicius);

int main() {
    int manoSkaicius = 7;
    system("chcp 1257"); // rašysime lietuviškai

    cout << "Pradinis skaičius: " << manoSkaicius << endl;

    // Kvietimas funkcijai perduodant nuorodą į skaičių
    padvigubinti(manoSkaicius);

    cout << "Po funkcijos kvietimo: " << manoSkaicius << endl;

    return 0;
}

```

Pagrindinis lygis.

Užduotis. Rasti trijų skaičių sumą, naudojant paprogramę, kai parametru perduodama reikšmė.

```

#include <iostream>

using namespace std;

```

```

// Funkcija, kuri randa trijų skaičių sumą ir išspausdina
rezultata
void rastiIrSpausdintiSuma(int a, int b, int c);

int main() {
    int skaicius1, skaicius2, skaicius3;

    system("chcp 1257"); // rašysime lietuviškai

    cout << "Įveskite pirmąjį skaičių: ";
    cin >> skaicius1;

    cout << "Įveskite antrąjį skaičių: ";
    cin >> skaicius2;

    cout << "Įveskite trečiąjį skaičių: ";
    cin >> skaicius3;

    // Kreipimasis į funkcija
    rastiIrSpausdintiSuma(skaicius1, skaicius2, skaicius3);

    return 0;
}

void rastiIrSpausdintiSuma(int a, int b, int c) {
    int suma = a + b + c;
    cout << "Trijų skaičių suma yra: " << suma << endl;
}

```

Aukštesnysis lygis.

Užduotis. Mokytoja pateikė mokiniui n skaičių degtukų. Mokinys gavo užduotį: iš degtukų sudėlioti didžiausio ploto stačiakampį, surasti stačiakampio kraštinių ilgius ir kiek degtukų liko nepanaudota.

```

#include <iostream>

using namespace std;

// Funkcija, kuri randa didžiausio ploto stačiakampį
void rastiStaciakampius(int n, int & krastine1, int & krastine2,
int & plotas, int & nepanaudoti);

int main() {

```

```

int n, plotas, krastine1, krastine2, nepanaudoti;

cout << "Įveskite degtukų skaičių: ";
cin >> n;

// Kvietimas į funkciją
rastiStaciakampus(n, plotas, krastine1, krastine2,
nepanaudoti);

cout << "Kraštinė1: " << krastine1 << endl;
cout << "Kraštinė2: " << krastine2 << endl;
cout << "Nepanaudoti: " << nepanaudoti << endl;
cout << "Stačiakampio plotas: " << plotas << endl;

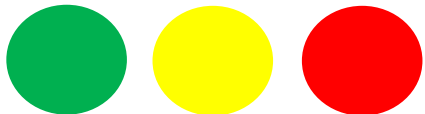
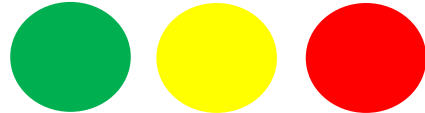
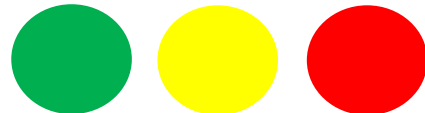
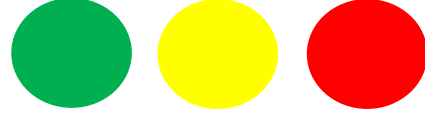
return 0;
}

//-----
void rastiStaciakampus(int n, int & krastine1, int & krastine2,
int & plotas, int & nepanaudoti) {
    if (n < 4){
        krastine1 = 0;
        krastine2 = 0;
        plotas = 0;
        nepanaudoti = n;
    }
    else if (n % 4 < 2){
        krastine1 = n / 4;
        krastine2 = n / 4;
        plotas = krastine1 * krastine2;
        nepanaudoti = n - 4 * krastine1;
    }
    else{
        krastine1 = n / 4;
        krastine2 = n / 4 + 1;
        plotas = krastine1 * krastine2;
        nepanaudoti = n - 2 * krastine1 - 2 * krastine2;
    }
}

```

5 ETAPAS 🕒 (2-3 min.)

Pamokos refleksija. Su mokiniais apibendrinamas atliktas darbas pamokoje. Įsivertinimo metodas „Šviesoforas“.

Pasirink tinkamus atsakymus į 1–3 klausimus, pažymėk skritulius. Savais žodžiais atsakyk į 4–5 klausimą.	
1. Kaip supratai pamokos medžiagą? Viską supratau – žalias Ne viską supratau – geltonas Nieko nesupratau – raudonas	
2. Kaip sekėsi dirbti „namų“ grupėje? Puikiai bendradarbiauome – žalias Dirbome kas sau – geltonas Nesutarėme – raudonas	
3. Kaip sekėsi dirbti „ekspertų“ grupėje? Puikiai bendradarbiauome – žalias Dirbome kas sau – geltonas Nesutarėme – raudonas	
4. Kaip jautėsi pamokoje? Puikiai – žalias Kartais gerai, kartais nelabai – geltonas Blogai – raudonas	
5. Ką darytum kitaip, kad patobulintum savo darbą?	

***Uždavinių sprendimas panaudojant paprogrames, kai parametru perduodama reikšmė
(Rasa Vilimienė-Jurkė)***

Pasiekimų sritis	B. Algoritmai ir programavimas. B3. Kuria ir vykdo algoritmus, programas.
Klasė	9-10
Tema	Uždavinių sprendimas panaudojant paprogrames, kai parametru perduodama reikšmė.
Integruojami dalykai, pasiekimai	Matematika, lietuvių k.
Kompetencijos	<p>Pažinimo – mokiniai ugdomi informatinį mąstymą – gebėjimą spręsti įvairias aplinkos problemas (uždavinius) analizuodami ir apdorodami duomenis, modeliuodami problemų sprendimo procesus ir, svarbiausia, siekdami automatizuoti tuos sprendimus.</p> <p>Skaitmeninė – mokiniai ugdomi gebėjimus patikimai, kritiškai ir atsakingai naudoti skaitmenines technologijas mokymuisi, darbui ir dalyvavimui visuomenės gyvenime.</p> <p>Kūrybiškumo – kurti produktus, kūrybiškai modeliuoti sprendimus, juos sistemiškai vertinti.</p>
Tikslas	Įtvirtinti žinias apie paprogrames, kai parametru perduodama reikšmė, sprendžiant uždavinius.
Uždaviniai	<ol style="list-style-type: none"> 1. Prisiminti kas yra paprogramė, kai parametru perduodama reikšmė, jos sintaksę. 2. Gebėti skaidyti algoritmus į smulkesnes, savarankiškas ir užbaigtas dalis. 3. Gebėti pritaikyti paprogrames, kai parametru perduodama reikšmė, taisyklingai jas užrašyti kuriant programą.

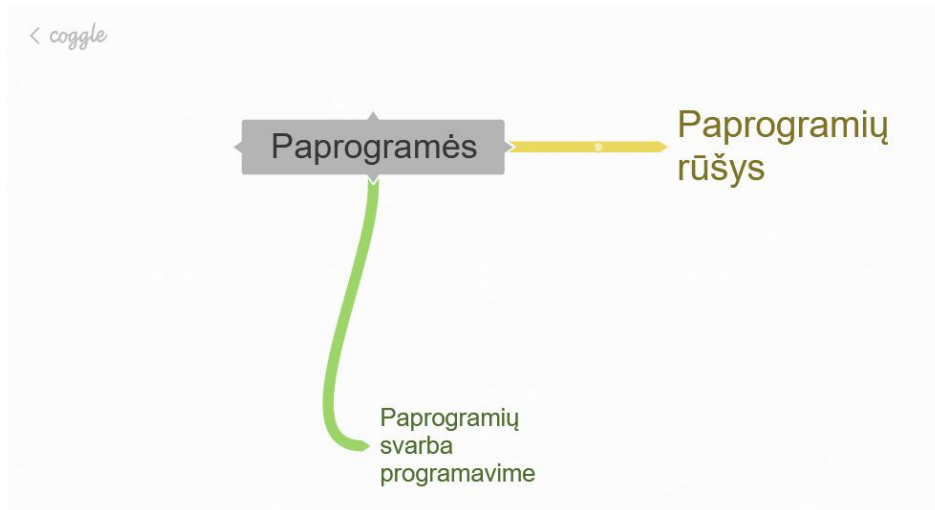
Planuojamas rezultatas	<ol style="list-style-type: none"> 1. Suprasite kas yra paprogramės, kai parametru perduodama reikšmė, jos sintaksę. 2. Skaidysite algoritmus į smulkesnes, savarankiškas ir užbaigtas dalis. 3. Pritaikysite paprogrames, kai parametru perduodama reikšmė. 4. Taisyklingai jas užrašysite kurdami programą.
Specifinės priemonės / programinė įranga	<p>Projektorius arba išmanioji lenta. Kompiuteriai. Interneto ryšys. Programa C++ (CodeBlock ar kt.). Informacijos šaltiniai:</p> <ul style="list-style-type: none"> • http://www.ugdome.lt/kompetencijos5-8/Irankiai/voratinklis_mob/grid/index.html • Renata Burbaitė, Jonas Blonskis, Vytautas Bukšnaitis. Šiuolaikiškas žvilgsnis į programavimą C++. Pasirenkamasis informacinių technologijų kursas XI-XII klasėms. TEV, 2011. • Renata Burbaitė. Šiuolaikiškas žvilgsnis į programavimą C++. Pasirenkamasis informacinių technologijų kursas XI-XII klasėms. Mokytojo knyga. Atmintukas. TEV, 2011. • https://if.ktu.edu/mokykloms/c-programavimo-pamokeles/
Mokymosi metodai	Minčių žemėlapis. Darbas porose. Individualus darbas (užduotys programėlėse). Uždavinio sprendimo metodas. Įsivertinimo metodas „Voratinklis“.
Mokinių atlikto darbo vertinimas ir įsivertinimas	<p>Slenkstinis pasiekimų lygis – nagrinėja programų su paprogramėmis, kai parametru perduodama reikšmė, pavyzdžius, padedamas naudoja parašytas paprogrames.</p> <p>Patenkinamas pasiekimų lygis – detalizuoja pateiktą uždavinį, nurodo reikalingus žingsnius jam spręsti, atpažįsta galimas paprogrames, kai parametru perduodama reikšmė, padedamas rašo paprogrames ir jas naudoja.</p> <p>Pagrindinis pasiekimų lygis – projektuoja programą, naudoja paprogrames kai parametru perduodama reikšmė.</p> <p>Aukštesnysis lygis – tinkamai aprašo, komentuoja paprogramių, kai parametru perduodama reikšmė, veikimą, kuria paprogrames, kurios grąžina rezultatus.</p>
Žinios prieš	Tinkamai naudojasi programine ir aparatine kompiuterio įranga, turi pagrindinius darbo su programavimo kalba C++ įgūdžius.
Galimybės taikyti spec. poreikių mokiniams	Mokytojas pamokos scenarijų pritaiko priklausomai nuo vaiko specialiųjų poreikių gebėjimų. Specialių poreikių mokiniams gali būti pateikiamas kuriamos programos šablonas.
Patarimai kolegoms, kurie naudos parengtą medžiagą	<ul style="list-style-type: none"> • Išbandyti išspręsti uždavinius patiems. • Išbandyti patiems įrankius: minčių žemėlapio kūrimą, voratinklį. • II etape, jei klasėje nėra išmaniosios lentos, užduotį galima aptarti žodžiu („nenešiojant“ atsakymų).

	<ul style="list-style-type: none">• Galima parinkti ir kitus uždavinius ar pateikti mažiau uždavinių, priklausomai nuo klasės lygio.• Voratinklyje nurodyti tiek uždavinių, kiek planuojate spręsti.
--	---

1 ETAPAS ⌚ (5-7 min.)

Minčių žemėlapis. Darbas porose. Mokiniai suskirstomi poromis. Kuria minčių žemėlapi.
Tema „Paprogramės“.

Minčių žemėlapis <https://coggle.it/>



2 ETAPAS ⌚ (4-7 min.)

Teorijos įtvirtinimui atliekami pratimai programėlėje individualiai.

<https://wordwall.net/resource/62083576>

Jei funkcija turi grąžinti keletą reikšmių, tuomet naudojami parametrai-nuorodos. Prieš juos funkcijos antraštyje rašomas ženklas

Paprogramės, kai parametru perduodama reikšmė, prototipas

Paprogramės antraštyje esantys parametrai vadinami

Funkcijos realizavimo aprašymas

Parametrai, esantys kreipinyje, vadinami

<input type="text"/>	faktiniais parametrais
<input type="text"/>	&
<input type="text"/>	void funkcijosVardas (tipas & vardas1, tipas & vardas2);
<input type="text"/>	void funkcijosVardas (tipas & vardas1, tipas & vardas2){ funkcijosKamienas }
<input type="text"/>	formaliaisiais parametrais

Atsakymas.

Parametrai, esantys kreipinyje, vadinami ✓

faktiniais parametrais

Jei funkcija turi grąžinti keletą reikšmių, tuomet naudojami parametrai-nuorodos. Prieš juos funkcijos antraštėje rašomas ženklas ✓

&

Paprogramės, kai parametru perduodama reikšmė, prototipas ✓

void funkcijosVardas (tipas & vardas1, tipas & vardas2);

Funkcijos realizavimo aprašymas ✓

void funkcijosVardas (tipas & vardas1, tipas & vardas2){ funkcijosKamienas }

Paprogramės antraštėje esantys parametrai vadinami ✓

formaliaisiais parametrais

Išmanojoje lentoje atliekama užduotis – sudėti teisingus atsakymus.

Jei funkcija turi grąžinti keletą reikšmių, tuomet naudojami parametrai-nuorodos. Prieš juos funkcijos antraštėje rašomas ženklas .

Tuo atveju, kai į funkciją kreipiamasi perduodant jai parametrus-reikšmes, funkcija sukuria naujus tų pačių tipų kintamuosius, kaip ir perduodami parametrai, ir jiems priskiria parametrų reikšmes. Vadinasi, funkcija dirba su parametrų reikšmių , bet ne su pačiais parametrais. Tai patogu, kai funkcijai nereikia keisti parametrų reikšmių.

Tuo atveju, kai į funkciją kreipiamasi perduodant jai parametrus-nuorodas, ji gauna ne kintamųjų reikšmes, o į kintamuosius

&	kopijomis	nuorodas	netikromis reikšmėmis	kintamaisiais
*	,			

Atsakymas

Jei funkcija turi grąžinti keletą reikšmių, tuomet naudojami parametrai-nuorodos. Prieš juos funkcijos antraštėje rašomas ženklas & ✓ .

Tuo atveju, kai į funkciją kreipiamasi perduodant jai parametrus-reikšmes, funkcija sukuria naujus tų pačių tipų kintamuosius, kaip ir perduodami parametrai, ir jiems priskiria parametrų reikšmes. Vadinasi, funkcija dirba su parametrų reikšmių kopijomis ✓ , bet ne su pačiais parametrais. Tai patogu, kai funkcijai nereikia keisti parametrų reikšmių.

Tuo atveju, kai į funkciją kreipiamasi perduodant jai parametrus-nuorodas, ji gauna ne kintamųjų reikšmes, o nuorodas ✓ į kintamuosius

3 ETAPAS 🕒 (30-35 min.)

Uždavinių sprendimas.

1 uždavinys. Nupiešti stačiakampį.

```
#include <iostream>

using namespace std;

void spausdinkEilute(int ilgis);

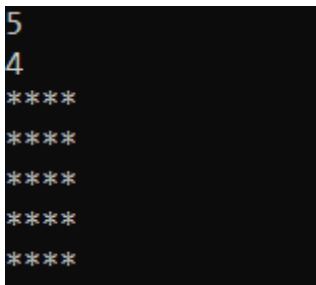
void spausdinkStaciakampi(int a, int p);
```

```

int main() {
    int aukstis, plotis;
    cin >> aukstis >> plotis; // iveda du skaičius - stačiakampio
    plotį ir aukštį
    spausdinkStaciakampi(aukstis, plotis);
    return 0;
}

void spausdinkEilute(int ilgis) {
    for(int i = 0; i < ilgis; i++) {
        cout << "*"; // simbolį gali pasirinkti patys
    }
    cout << endl;
}
//-----
void spausdinkStaciakampi(int a, int p) {
    for(int i = 0; i < a; i++) {
        spausdinkEilute(p); // a kartų spausdiname eilutę,
        kurios dydis p
    }
}

```



2 uždavinys. Kambaryje stovi trikojės taburetės ir keturkojės kėdės. Kai ant visų taburečių ir kėdžių susėdo žmonės, kambaryje iš viso buvo n kojų. Parašykite programą kėdžių ir taburečių skaičiui rasti kambaryje? Pradiniai duomenys – kojų skaičius n, įvedamas klaviatūra.

```

#include <iostream>

using namespace std;

// Funkcija, kuri randa kėdžių ir taburečių skaičių
void rastiKedesIrTaburetes(int n, int &kedes, int &taburetes);

```

```

int main() {
    int n, kedes, taburetes;

    cout << "Įveskite kojų skaičių: ";
    cin >> n;

    // Kvietimas į funkciją
    rastiKedesIrTaburetes(n, kedes, taburetes);

    cout << "Kėdžių skaičius: " << kedes << endl;
    cout << "Taburečių skaičius: " << taburetes << endl;

    return 0;
}

//-----
void rastiKedesIrTaburetes(int n, int &kedes, int &taburetes) {
    // Pagal sąlygą, viena kėdė turi 4 kojas, o taburetė turi 3
    // kojas.
    kedes = n / 4;
    taburetes = (n % 4) / 3;
}

```

3 uždavinys. Gerai užderėjus obuolių derliui, ūkininkas nusprendė gaminti obuolių sultis. Pagamintas sultis išpilstė į 5, 2 ir 1 litro talpos indus. Pirmiausia užpildė 5, po to – 2 ir po to – 1 litro talpos indus. Parenkite programą, kuri apskaičiuotų, kiek 5, 2 ir 1 litro indų sulčių buvo pagaminta. Pradiniai duomenys – litrų skaičius n, įvedamas klaviatūra.

```

#include <iostream>
using namespace std;

void Sultys (int & kiek_sulciu, int talpa, int & kiek_indu);

int main(){
    int sulciu_kiekis;    int k5, k3, k1;
    cin >> sulciu_kiekis;
    Sultys(sulciu_kiekis, 5, k5);
    Sultys(sulciu_kiekis, 3, k3);
    Sultys(sulciu_kiekis, 1, k1);
    cout << k5 << " " << k3 << " " << k1 << endl;
    return 0; }

//-----
void Sultys (int & kiek_sulciu, int talpa, int & kiek_indu){

```

```
kiek_indu = kiek_sulciu / talpa;
kiek_sulciu = kiek_sulciu % talpa; }
```

```
39
7 1 1
```

4 uždavinys. Iš Vilniaus autobusų stoties autobusai važiuoja tam tikrais maršrutais. Autobusų tvarkaraštyje nurodyta autobuso išvykimo laikas ir kelionės trukmė. Sukurkite funkciją, kuri apskaičiuotų kada autobusas turi atvykti į galutinę stotelę.

```
#include <iostream>
#include <cmath>

using namespace std;

//-----
void Laikas(int v1, int m1, int v2, int m2, int & v3, int & m3);
//-----

int main()
{
    int v1, m1, // autobuso išvykimo laikas (val. ir min.)
        v2, m2, // kelionės trukmė (val. ir min. skaičiai)
        v3, m3; // atvykimo į paskirties vietą laikas (val. ir
min.)

    cout << "Įveskite išvykimo laiką: ";
    cin >> v1 >> m1;

    cout << "Įveskite kelionės trukmės laiką: ";
    cin >> v2 >> m2;
    Laikas(v1, m1, v2, m2, v3, m3);
    cout << "Atvykimo laikas: " << v3 << " " << m3 << endl;

    return 0;
}
//-----
// Apskaičiuoja autobuso atvykimo į paskirties vietą laiką
// v1, m1 - išvykimo laikas (val. ir min.)
// v2, m2 - kelionės trukmė val. ir min.
// v3, m3 - atvykimo į paskirties vietą laikas (val. ir min.)
```

```

void Laikas(int v1, int m1, int v2, int m2, int & v3, int & m3)
{
    v3 = (v1 * 60 + m1 + v2 * 60 + m2) / 60 % 24;
    m3 = (v1 * 60 + m1 + v2 * 60 + m2) % 60;

}

```

5 uždavinys. Sukurti colinę matavimo sistemą, kuri colius paverstų į kitus matavimo vienetus.

```

#include <iostream>

using namespace std;

void konvertuotiColiusIMetrus(double coliai, double & metrai,
double & pedos, double & jardai);

int main() {
    double coliai, metrai, pedos, jardai;

    cout << "Įveskite colius: ";
    cin >> coliai;

    konvertuotiColiusIMetrus(coliai, metrai, pedos, jardai);

    cout << "Atsakymas: " << coliai << " coliai = " << metrai
<< " metrų arba " << pedos << " pėdos arba " << jardai << "
jardai" <<endl;

    return 0;
}

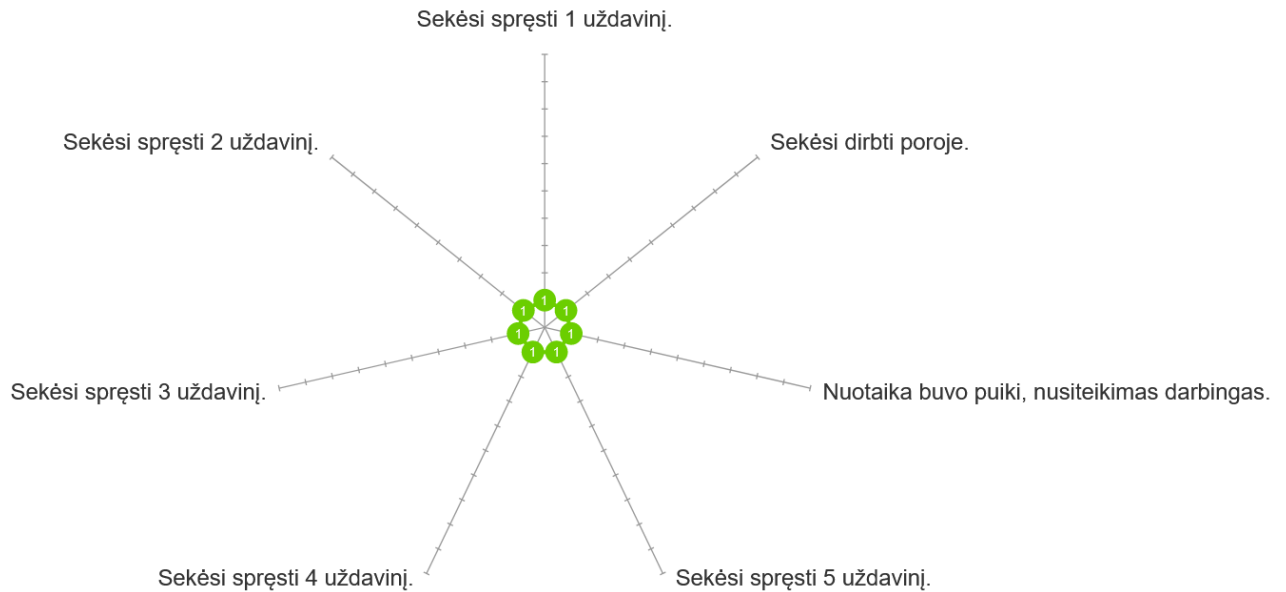
void konvertuotiColiusIMetrus(double coliai, double & metrai,
double & pedos, double & jardai) {
    // 1 colis = 0.0254 metrų
    // 1 colis = 0.083 pėdos
    // 1 colis = 0.028 jardo

    metrai = coliai * 0.0254;
    pedos = coliai * 0.083;
    jardai = coliai * 0.028;
}

```

4 ETAPAS 🕒 (3-4 min.)

Įsivertinimas. Voratinklis. Įsivertinti nuo 1 iki 10. Puiku – 10, blogai – 1.



Įvadas į programų projektavimą (Jolita Lazauskienė)

Pasiekimų sritis	Programų projektavimas (B sritis: Algoritmai ir programavimas)
Klasė	9-10 kl.
Tema	Įvadas į programų projektavimą
Integruojami dalykai, pasiekimai	Lietuvių kalba, anglų kalba, gamtos mokslai.
Kompetencijos	<p>Kūrybiškumo kompetencija: inžinerinis programų projektavimo ciklas pristatomas kaip itin įvairiapusė ir kūrybiška veikla, reikalaujanti sumanumo, naujų idėjų, jungianti technologinius, psichologinius, socialinius, meninius aspektus. Skatinama įvairiapusė mokinių veikla, ypatingas dėmesys kreipiamas į inžinerinę ir su gamtamoksliniais dalykais susijusias kūrybines veiklas, jų plėtojimą, ugdomas poreikis savarankiškai tirti, ieškoti, nagrinėti ir kritiškai vertinti informaciją, skaitmeninius įrankius, programas, generuoti sau ir kitiems reikšmingas idėjas, kurti produktus, kūrybiškai modeliuoti sprendimus, juos sistemiškai vertinti.</p> <p>Pažinimo kompetencija: problemos identifikavimas – mokiniai išmoksta identifikuoti ir aiškiai apibrėžti esamas problemas. Reikalavimų analizė –</p>

	<p>mokiniai išmoksta analizuoti programos projektavimo poreikius ir reikalavimus. Projektavimo planavimas – mokiniai išmoksta kurti planą programos projektavimo darbams.</p> <p>Skaitmeninė kompetencija: mokiniai mokosi naudotis programine įranga, kuria galima atlikti programos projektavimą.</p> <p>Komunikavimas – mokiniai supranta, kad tinkama komunikacija su komandos nariais yra svarbi programos projektavimo proceso dalis.</p>
Tikslas	Išmokti formuluoti problemas ir planuoti savo veiklas projektuojant programas.
Uždaviniai	<p>9. Suprasti, kaip teisingai identifikuoti ir formuluoti problemas.</p> <p>10. Išmokti kritiškai vertinti problemas ir jų sprendimo būdus.</p> <p>11. Išmokti sudaryti veiksmingą planą programos projektavimui.</p>
Planuojamas rezultatas	<p>7. Gebėsite aiškiai suformuluoti problemą ir apibūdinti programų projektavimo proceso etapus.</p> <p>8. Identifikuosite ir aprašysite reikalavimus, kurie turi būti tenkinami kuriant programą.</p> <p>9. Gebėsite sukurti paprastą projektavimo planą, kuriame būtų nurodyti veiksmingi programos projektavimo etapai.</p>
Specifinės priemonės / programinė įranga	Teksto rengyklė „Microsoft Word“
Mokymosi metodai	<p>Demonstravimas. Mokiniam pristatomi pagrindiniai programų projektavimo etapai.</p> <p>Darbas grupėmis. Mokiniai padalinami į mažas grupes, jiems pateikiama užduotis parengti apibrėžtos problemos sprendimo projektą. Suteikiamas laikas, kad grupėse dirbantys mokiniai turėtų galimybę aptarti užduoties sprendimo būdus. Taip skatinamas bendradarbiavimas, minčių lietus ir praktinis tyrimas.</p> <p>Diskusija. Mokiniai galėtų aptarti, kaip būtų galima spręsti nurodytą problemą.</p>
Mokinių atlikto darbo	Slenkstinis – su mokytojo pagalba geba įvardinti programų projektavimo etapus.

vertinimas ir įsivertinimas	<p>Patenkinamas – su mokytojo pagalba identifikuoja ir suformuluoja programų projektavimo etapus, dalyvauja diskusijoje kaip juos įgyvendinti.</p> <p>Pagrindinis – savarankiškai identifikuoja ir suformuluoja programų projektavimo etapus, dalyvauja diskusijoje ir siūlo sprendimo variantus.</p> <p>Aukštesnysis – savarankiškai identifikuoja ir suformuluoja programų projektavimo etapus, dalyvauja diskusijoje bei sukuria veiksmingą planą etapams įgyvendinti.</p>
Žinios prieš	Geba kurti paprastas programas, paaiškinti jų veikimą.
Patarimai kolegoms, kurie naudos parengtą medžiagą	<ol style="list-style-type: none"> 1. Turėkite papildomų mokiniams teikiamų pavyzdžių. 2. Per pamoką mokiniai dirba savarankiškai, mokytojas yra pagalbininkas ir konsultantas.

1 ETAPAS (5-8 minutės). Įvadas į temą.

Mokytojas paaiškina kodėl programų kūrimo procese problemos formulavimas ir projektavimo planavimas yra toks svarbus, aptaria kaip šie etapai gali paveikti visą projektą:

1. Problemos formulavimas nustato tikslus. Tai leidžia aiškiai apibrėžti, kokių problemų sprendimui bus kuriama programa. Tai suteikia aiškų tikslą komandai ir klientui bei padeda išvengti nereikalingų funkcijų įtraukimo į programą.
2. Projektavimo planavimas organizuoja darbo procesą: Projekto planas nustato, kaip bus vykdomas visų užduočių atlikimas ir kokios yra pagrindinės etapinės fazės. Tai padeda efektyviai valdyti projektą, skirstyti užduotis, numatyti laiko reikmes ir išteklius.
3. Rizikos mažinimas: Problemos formulavimas ir projektavimo planavimas padeda nustatyti potencialias problemas ir rizikos faktorius ankstyvaisiais etapais, kai jų šalinimas yra paprasčiausias ir pigiausias. Tai sumažina klaidų ir problemų riziką vėlesniais projektų etapais.
4. Efektyvus komandos darbas: Aiškiai suformuluotos užduotys ir projektavimo planas padeda komandai suprasti, kas turi būti padaryta, ir kaip tai turi būti padaryta. Tai skatina geresnį bendradarbiavimą ir sumažina nesusipratimų riziką.
5. Reikalavimų nustatymas: Problemos formulavimas padeda nustatyti programos reikalavimus, o projektavimo planavimas padeda užtikrinti, kad šie reikalavimai bus atitinkamai įgyvendinti.

2 ETAPAS (30 minučių). Grupinis darbas.

1 dalis. Mokiniamis pateikiama užduotis. Tarkime, kad mokiniai yra atsakingi už programos kūrimą, kuri leis žmonėms užsisakyti maistą internetu. Mokiniamis pateikiama užduotis suformuluoti tokios programos kūrimo tikslą, reikalavimus ir galimų problemų apibrėžimus (tikslai, reikalavimai ir problemos užrašomos „Microsoft Word“ programoje). Pavyzdžiui:

- **Tikslas:** Kūrimo tikslas yra sukurti internetinę maisto užsakymo svetainę, kuri leis vartotojams užsisakyti maisto, naršyti siūlomų patiekalų sąrašą, rasti informaciją apie patiekalų sudėtį ir alergenus ir vykdyti užsakymus.

- **Reikalavimai:** Nustatyti pagrindinius reikalavimus, tokius kaip vartotojo registracija, patiekalų sąrašo peržiūra, užsakymo pateikimas ir apmokėjimas, maisto pristatymas ir kt.

- **Problemos apibrėžimas:** Nustatyti, kaip efektyviai valdyti patiekalų užsakymo katalogą, kaip užtikrinti saugų apmokėjimą ir kaip organizuoti užsakymų vykdymą.

Mokinių grupėms užduočiai atlikti suteikiama 15 minučių. Dar 5-8 minutės skiriamos užduoties aptarimui ir diskusijoms.

2 dalis. Aptarus užduoties tikslą, reikalavimus ir galimas problemas mokiniamis duodama 2 užduoties dalis. Aptarti tokios programos projektavimo planavimo dalis: reikalingi ištekliai, programos funkcijos, dizainas, testavimo strategijos (Užduočiai atlikti taip pat naudojama „Microsoft Word“ programa). Pavyzdys:

- **Ištekliai nustatymas:** Nustatyti, kiek programuotojų, dizainerių ir kitų specialistų reikės projektui.

- **Funkcijų sąrašas:** Nustatyti visas funkcijas, kurias programa turi turėti, įskaitant vartotojo registraciją, prisijungimą, patiekalų peržiūrą, užsakymo pateikimą, apmokėjimą ir kt.

- **Interfeiso dizainas:** Nustatyti, kaip svetainė turi atrodyti, įtraukiant spalvas, šriftus, logotipą ir t.t.

- **Testavimo strategija:** Nustatyti, kaip bus testuojamos programos funkcijos, kad būtų užtikrinta kokybė.

Mokinių grupėms 2 užduoties daliai atlikti taip pat suteikiama 15-20 minučių. Dar 10 minučių skiriama užduoties aptarimui ir diskusijoms.

Vertinimas: mokytojas stebi mokinių diskusijas ir vertina atliktą darbą pagal iš anksto apibrėžtus kriterijus.

Kriterijus	1	2	3	4	5
Problemos formulavimas					
Reikalavimai					
Projektavimo etapų detalizavimas					

Programų projektavimo uždavinių skaidymas dalimis (Jolita Lazauskienė)

Pasiekimų sritis	Programų projektavimas (B sritis: Algoritmai ir programavimas)
Klasė	9-10 kl.
Tema	Programų projektavimo uždavinių skaidymas dalimis
Integruojami dalykai, pasiekimai	Lietuvių kalba, anglų kalba, gamtos mokslai.
Kompetencijos	<p>Kūrybiškumo kompetencija: inžinerinis programų projektavimo ciklas pristatomas kaip itin įvairiapusė ir kūrybiška veikla, reikalaujanti sumanumo, naujų idėjų, jungianti technologinius, psichologinius, socialinius, meninius aspektus. Skatinama įvairiapusė mokinių veikla, ypatingas dėmesys kreipiamas į inžinerinę ir su gamtamoksliniais dalykais susijusias kūrybines veiklas, jų plėtojimą, ugdomas poreikis savarankiškai tirti, ieškoti, nagrinėti ir kritiškai vertinti informaciją, skaitmeninius įrankius, programas, generuoti sau ir kitiems reikšmingas idėjas, kurti produktus, kūrybiškai modeliuoti sprendimus, juos sistemiškai vertinti.</p> <p>Pažinimo kompetencija: problemos identifikavimas – mokiniai išmoksta identifikuoti ir aiškiai apibrėžti esamas problemas. Reikalavimų analizė – mokiniai išmoksta analizuoti programos projektavimo poreikius ir reikalavimus. Projektavimo planavimas – mokiniai išmoksta kurti planą programos projektavimo darbams.</p> <p>Skaitmeninė kompetencija: mokiniai mokosi naudotis programine įranga, kuria galima atlikti programos projektavimą.</p>

	Komunikavimas – mokiniai supranta, kad tinkama komunikacija su komandos nariais yra svarbi programos projektavimo proceso dalis.
Tikslas	Išmokinti mokinius su programų projektavimo uždavinių skaidymo dalimis principais.
Uždaviniai	12. Paaiškinti, kas yra programų projektavimo užduotys ir kodėl svarbu jas skaidyti. 13. Išmokinti diagramų užduotis skaidyti dalimis. 14. Išmokinti mokinius praktiškai išbandyti skaidyti užduotis.
Planuojamas rezultatas	1. Gebėsite aiškiai paaiškinti, kodėl svarbu skaidyti programavimo užduotis į mažesnes dalis. 2. Gebėti naudoti diagramas užduočių skaidymui dalimis.
Specifinės priemonės / programinė įranga	Pristatymo skaidrės. Internetinė mokymo priemonė diagramoms kurti „Smartdraw“ (https://www.smartdraw.com/flowchart/flowchart-maker.htm).
Mokymosi metodai	Demonstravimas. Mokytojas pateikia mokiniams naują medžiagą naudojant skaidres. Šis metodas naudingas pristatant teorinę medžiagą ir svarbius sąvokų pagrindus. Darbas grupėse. Mokiniai dirba grupėse, kartu sprendžia užduotis ir diskutuoja apie mokomą temą. Tai skatina bendradarbiavimą, komunikaciją ir mokinių gebėjimą dirbti komandoje. Diskusija. Mokiniai dalyvauja pokalbyje apie mokomą temą ir su ja susijusias problemas. Diskusijai vadovauja mokytojas, tačiau ji gali vykti ir laisva forma. Diskusijos skatina mokinių kritinį mąstymą, argumentaciją ir nuomonės laisvės išraišką. Savarankiškas darbas. Mokiniai atlieka savarankišką užduotį be tiesioginio mokytojo vadovavimo. Mokiniai yra skatinami prisiimti atsakomybę, problemų sprendimą ir savęs valdymą.
Mokinių atlikto darbo vertinimas ir įsivertinimas	Mokiniai bus vertinami pagal tai, kaip aktyviai dalyvavo diskusijoje ir kaip gerai suprato, kaip skaidomos užduotys. Taip pat jie bus vertinami už savarankišką darbą, kurio metu turės skaidyti konkretų užduoties pavyzdį.

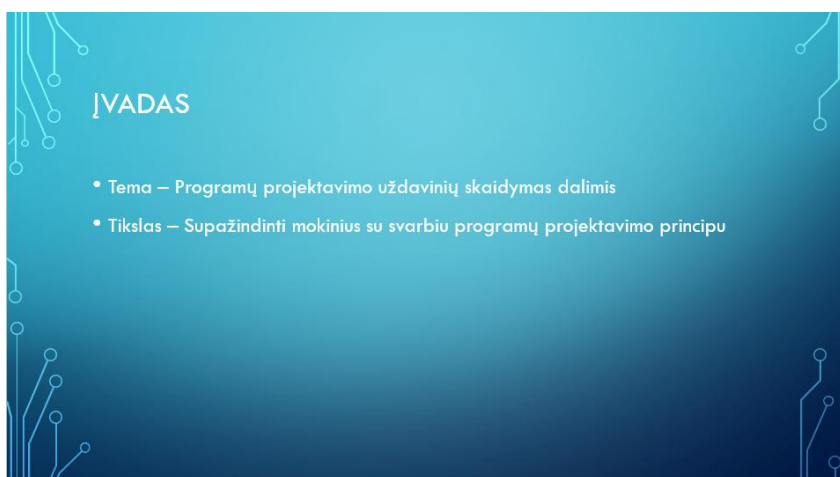
Žinios prieš	Mokiniai turėtų žinoti, kas yra programų projektavimas, mokėti formuluoti ir identifikuoti jų problemas ir sprendimo būdus. Mokėti kurti diagramas.
Patarimai kolegoms, kurie naudos parengtą medžiagą	<p>16. Užtikrinkite, kad mokiniai turėtų prieigą prie kompiuterių ir interneto, jei planuojate naudoti praktines užduotis.</p> <p>17. Skatinkite aktyvų dalyvavimą ir diskusijas, kad mokiniai geriau suprastų, kodėl programų projektavimo užduotys turi būti skaidomos į mažesnes dalis.</p> <p>18. Pasiruoškite papildomos medžiagos, jei mokiniai yra greičiau besimokantys ir nori gilintis į šią temą.</p> <p>19. Pabrėžkite praktinio mokymosi ir mokinių gebėjimų taikyti tai, ką jie išmoko svarbą.</p>

1 ETAPAS (10 minučių). Teorinės dalies pristatymas skaidrėse.

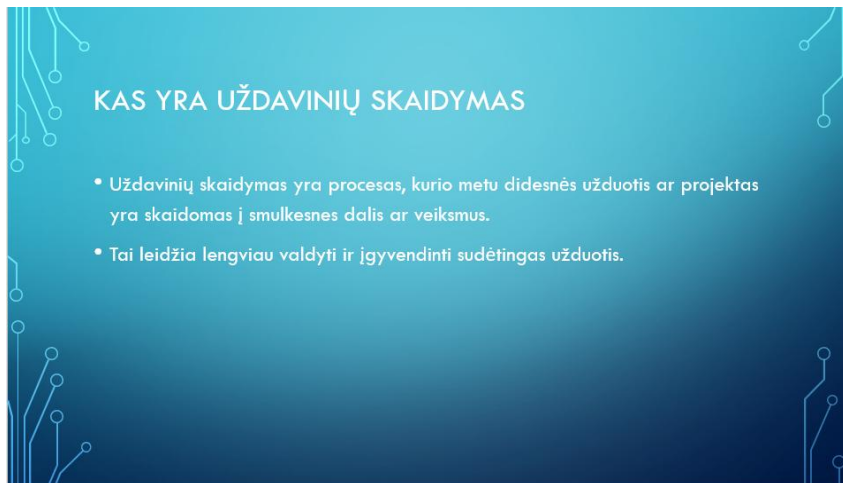
Mokytojas skaidrėse pateikia teorinę medžiagą, kad mokiniai susipažintų su tema „Programų projektavimo uždavinių skaidymas dalimis“:



1 pav. 1 teorinės medžiagos skaidrė



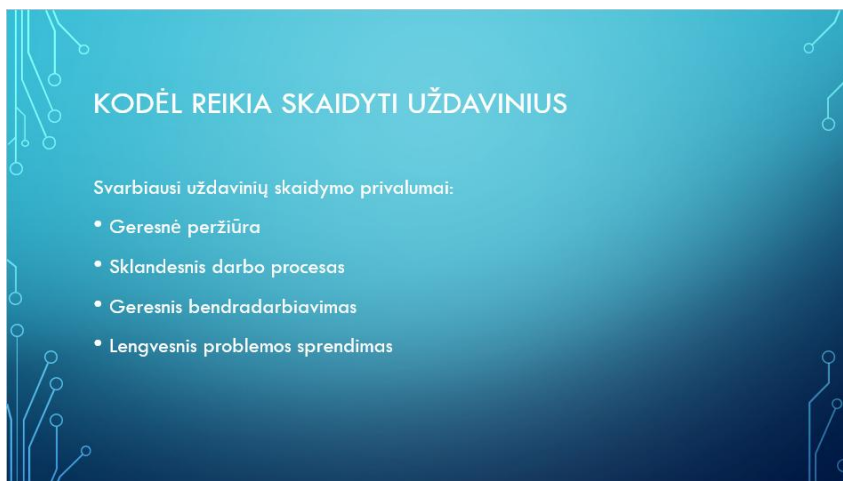
2 pav. 2 teorinės medžiagos skaidrė



KAS YRA UŽDAVINIŲ SKAIDYMAS

- Uždavinių skaidymas yra procesas, kurio metu didesnės užduotys ar projektas yra skaidomas į smulkesnes dalis ar veiksmus.
- Tai leidžia lengviau valdyti ir įgyvendinti sudėtingas užduotis.

3 pav. 3 teorinės medžiagos skaidrė

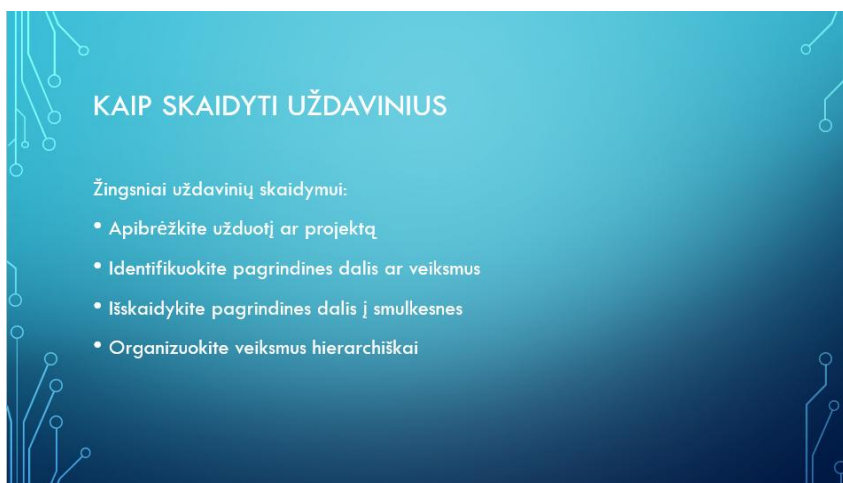


KODĖL REIKIA SKAIDYTI UŽDAVINIUS

Svarbiausi uždavinių skaidymo privalumai:

- Geresnė peržiūra
- Sklandesnis darbo procesas
- Geresnis bendradarbiavimas
- Lengvesnis problemos sprendimas

4 pav. 4 teorinės medžiagos skaidrė

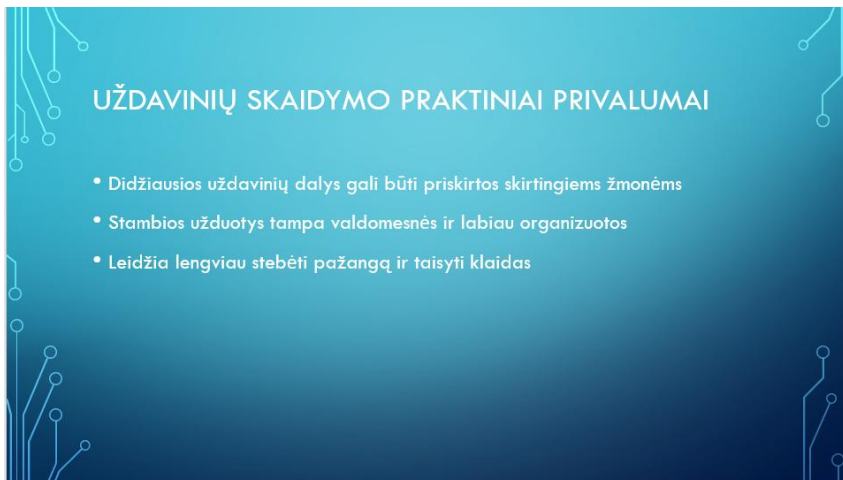


KAIP SKAIDYTI UŽDAVINIUS

Žingsniai uždavinių skaidymui:

- Apibrėžkite užduotį ar projektą
- Identifikuokite pagrindines dalis ar veiksmus
- Išskaidykite pagrindines dalis į smulkesnes
- Organizuokite veiksmus hierarchiškai

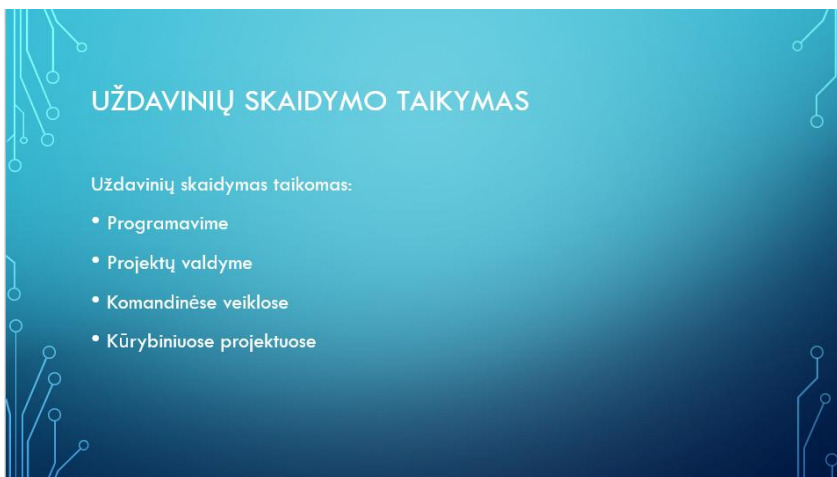
5 pav. 5 teorinės medžiagos skaidrė



UŽDAVINIŲ SKAIDYMO PRAKTINIAI PRIVALUMAI

- Didžiausios uždavinių dalys gali būti priskirtos skirtingiems žmonėms
- Stambios užduotys tampa valdomesnės ir labiau organizuotos
- Leidžia lengviau stebėti pažangą ir taisyti klaidas

6 pav. 6 teorinės medžiagos skaidrė

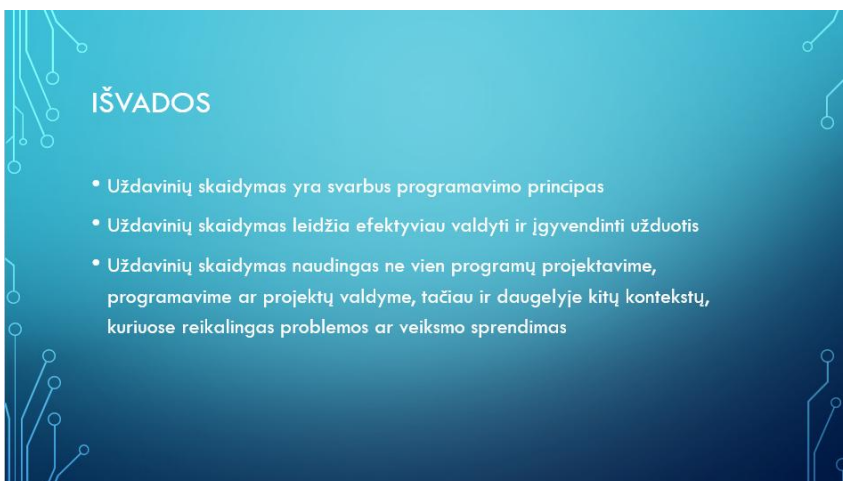


UŽDAVINIŲ SKAIDYMO TAIKYMAS

Uždavinių skaidymas taikomas:

- Programavime
- Projektų valdyme
- Komandinėse veiklose
- Kūrybiniuose projektuose

7 pav. 7 teorinės medžiagos skaidrė



IŠVADOS

- Uždavinių skaidymas yra svarbus programavimo principas
- Uždavinių skaidymas leidžia efektyviau valdyti ir įgyvendinti užduotis
- Uždavinių skaidymas naudingas ne vien programų projektavime, programavime ar projektų valdyme, tačiau ir daugelyje kitų kontekstų, kuriuose reikalingas problemos ar veiksmo sprendimas

8 pav. 8 teorinės medžiagos skaidrė

2 ETAPAS (15 minučių). Darbas grupėse.

1 dalis. Mokiniai pasiskirsto į grupes ir jiems pateikiama užduotis. Kelionės planavimo užduočių skaidymas dalimis. Mokiniais suteikiama 10 minučių laiko, kad kelionės planavimo užduotį suskaidytų į veiksmų seką (bent 5 veiksmų seka). Pavyzdys:

1. Pasirinkti kelionės tikslą
2. Nuspręsti transporto priemonę
3. Užsakyti skrydį, traukinį, ar išnuomuoti automobilį
4. Rezervuoti viešbučio kambarį
5. Sudaryti kelionės grafiką
6. Suplanuoti kelionės išlaidas

2 dalis. Po 10 minučių mokiniai pateikia savo veiksmų sekas ir kartu su mokytoju diskutuoja bei išrenka optimaliausią veiksmų seką šiai užduočiai įgyvendinti.

3 ETAPAS (15 minučių). Savarankiškas darbas.

Atlikus darbą grupėse bei mokiniams susipažinus su užduočių skaidymu į smulkesnes dalis, mokiniams skiriama savarankiško darbo užduotis – internetinės svetainės kūrimo proceso užduočių skaidymas į smulkesnes dalis ar veiksmus. Užduočiai atlikti pasitelkiama internetinė mokymo priemonė diagramoms kurti „Smartdraw“:

1. Mokiniai naršyklės lange paieškos langelyje įveda užklausa „Smartdraw“
2. Pasirodžiusiame lange spaudžiame mygtuką „Start Now“ (Pradėti dabar):

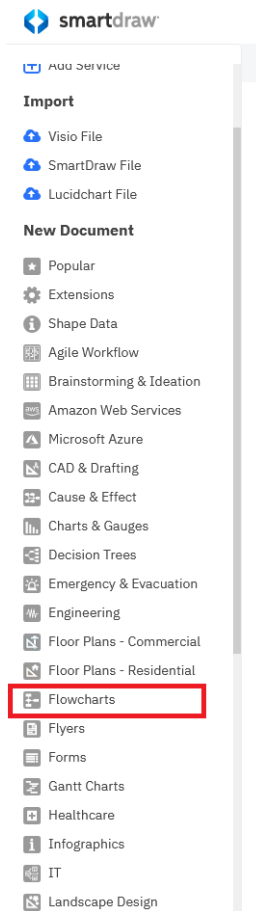
A Unified Visual App

SmartDraw combines diagramming, whiteboarding and data into one enterprise-friendly solution

Start Now >

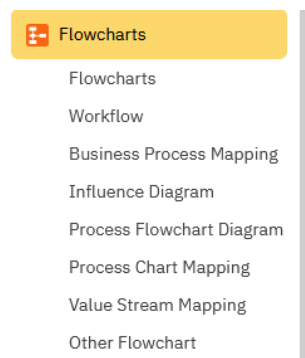
9 pav. Internetinės mokymo priemonės „Smartdraw“ pradžios ekranas

3. Kairėje interneto naršyklės lango pusėje susirandame ir pasirenkame priemonę „Flowcharts“:



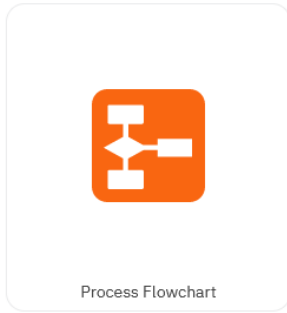
10 pav. Internetinės mokymo priemonės „Smartdraw“ diagramų kūrimo funkcija

4. Spustelėjus funkcija „flowcharts“ išsiskleidžia diagramos tipų pasirinkimo galimybė:



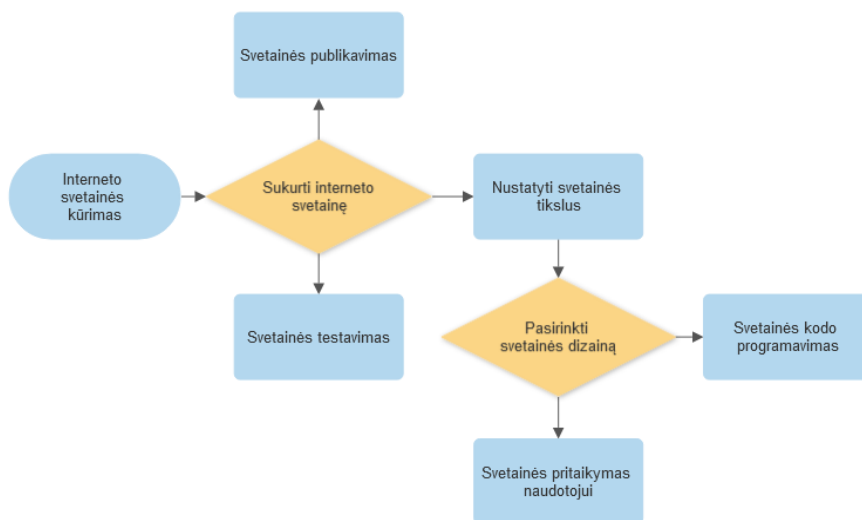
11 pav. Internetinės mokymo priemonės „Smartdraw“ diagramos tipo pasirinkimas

5. Pasirenkame „Process Flowchart diagram“ (procesų diagrama).
6. Paspaudžiame mygtuką „Process flowchart“:



12 pav. Internetinės mokymo priemonės „Smartdraw“ proceso diagrama

7. Galime pradėti kurti Interneto svetainės kūrimo proceso užduočių skaidymo dalimis diagramą:



13 pav. „Smartdraw“ internetinės svetainės kūrimo proceso diagrama

Mokiniams užduočiai atlikti skiriamos 10 minučių po kurių kartu su mokytoju aptariamos sukurtos diagramos ir vykdoma diskusija koks yra šios užduoties sprendimo optimaliausias variantas.

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

Su mokiniais apibendrinamas atliktas darbas.

Vertinimas: kai viena grupė pristato savo darbą, kitos stebi ir vertinimo lentelėje įrašo savo vertinimus balais, pagal iš anksto apibrėžtus kriterijus. Po to parašo atsiliepimus: 2 pagyrimus ir 2 patarimus ir pasirašo.

Kriterijus	1	2	3	4	5
Problemos formulavimas					
Reikalavimai					
Projektavimo etapų detalizavimas					

1. Mokiniai tarpusavyje pasidalina ir pristato atliktais savarankiškais darbais.
2. Mokiniai įsivertina pasirinktą diagramos tipą ir jo tinkamumą užduočiai atlikti.
3. Mokytojas įvertina mokinių aktyvumą visos pamokos metu, ar mokiniai supranta teorinę medžiagą ir ar geba praktiškai taikyti užduočių skaidymo principus.
4. Mokiniai pasidalina savo pastebėjimais apie tai, kas jiems buvo lengva ar sunku. Atsiliepimai padeda mokytojui suprasti, kaip tobulinti pamokos turinį ir metodiką kitų pamokų metu. Mokinių įsivertinimas padės mokytojui adaptuoti pamokos eigą pagal mokinių galimybes, reakcijas ir sunkumus.

Programų projektavimas: praktinis tyrimas (Jolita Lazauskienė)

Pasiekimų sritis	Programų projektavimas (B sritis: Algoritmai ir programavimas)
Klasė	9-10 kl.
Tema	Programų projektavimas: praktinis tyrimas
Integruojami dalykai, pasiekimai	Lietuvių kalba, anglų kalba, gamtos mokslai.
Kompetencijos	Kūrybiškumo kompetencija: inžinerinis programų projektavimo ciklas pristatomas kaip itin įvairiapusė ir kūrybiška veikla, reikalaujanti sumanumo,

	<p>naujų idėjų, jungianti technologinius, psichologinius, socialinius, meninius aspektus. Skatinama įvairiapusė mokinių veikla, ypatingas dėmesys kreipiamas į inžinerinę ir su gamtamoksliniais dalykais susijusias kūrybines veiklas, jų plėtojimą, ugdomas poreikis savarankiškai tirti, ieškoti, nagrinėti ir kritiškai vertinti informaciją, skaitmeninius įrankius, programas, generuoti sau ir kitiems reikšmingas idėjas, kurti produktus, kūrybiškai modeliuoti sprendimus, juos sistemiškai vertinti.</p> <p>Pažinimo kompetencija: problemos identifikavimas – mokiniai išmoksta identifikuoti ir aiškiai apibrėžti esamas problemas. Reikalavimų analizė – mokiniai išmoksta analizuoti programos projektavimo poreikius ir reikalavimus. Projektavimo planavimas – mokiniai išmoksta kurti planą programos projektavimo darbams.</p> <p>Skaitmeninė kompetencija: mokiniai mokosi naudotis programine įranga, kuria galima atlikti programos projektavimą.</p> <p>Komunikavimas – mokiniai supranta, kad tinkama komunikacija su komandos nariais yra svarbi programos projektavimo proceso dalis.</p>
Tikslas	Gilinti problemų formulavimo ir savo veiklas planavimo projektuojant programas, atvaizduoti projektavimo etapus grafiškai.
Uždaviniai	<ol style="list-style-type: none"> 1. Suprasti, kaip teisingai identifikuoti ir formuluoti problemas, jas pateikti suprantama forma. 2. Mokyti kritiškai vertinti problemas ir jų sprendimo būdus, tinkamai juos atvaizduoti. 3. Sudaryti veiksmingą planą programos projektavimui.
Planuojamas rezultatas	<ol style="list-style-type: none"> 1. Gebėsite aiškiai suformuluoti problemą, ją pateikti suprantama forma. 2. Identifikuosite ir aprašysite reikalavimus, kurie turi būti tenkinami kuriant programą, juos atvaizduosite grafiškai. 3. Gebėsite sukurti paprastą projektavimo planą, kuriame būtų nurodyti veiksmingi programos projektavimo etapai.
Specifinės priemonės / programinė įranga	<p>Diagramų kūrimo priemonė „Draw.io“ https://app.diagrams.net/</p> <p>Teksto rengyklė „Microsoft Word“</p>

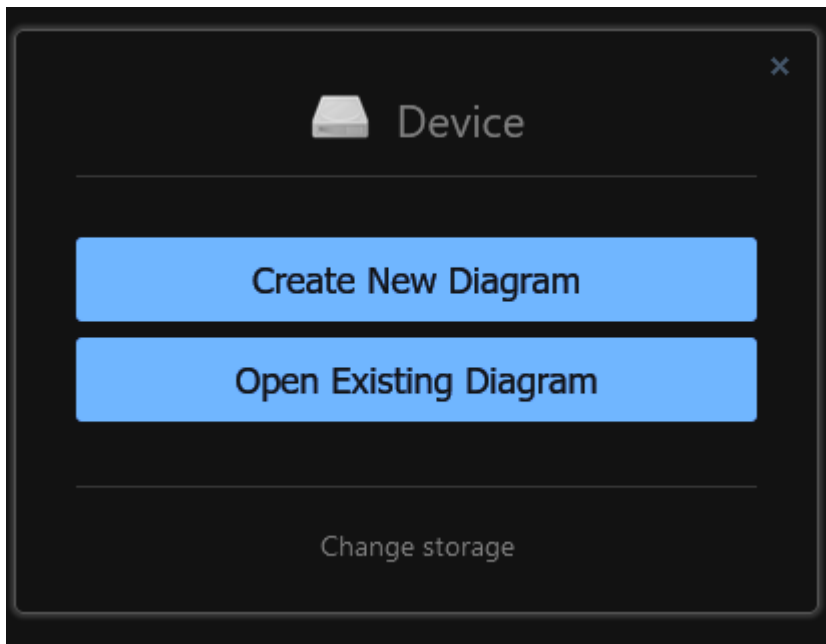
Mokymosi metodai	<p>Apversta klasė. Mokiniai savarankiškai namuose išsiaiškina kaip dirbti diagramų kūrimo priemone „Draw.io“.</p> <p>Praktinis tyrimas. Naudodami diagramų kūrimo priemonę „Draw.io“ mokiniai praktiškai išbando gautos užduoties problemos identifikavimo, formulavimo ir projektavimo plano kūrimo procesą.</p> <p>Darbas grupėse. Mokiniai grupėse kuria problemos identifikavimo, formulavimo ir projektavimo plano diagramą pasitelkus diagramų kūrimo įrankį.</p>
Mokinių atlikto darbo vertinimas ir įsivertinimas	<p>Slenkstinis – su mokytojo pagalba geba atvaizduoti programų projektavimo etapus panaudodami diagramų kūrimo įrankį.</p> <p>Patenkinamas – su mokytojo pagalba identifikuoja ir suformuluoja programų projektavimo etapus, savarankiškai geba atvaizduoti programų projektavimo etapus panaudodami diagramų kūrimo įrankį.</p> <p>Pagrindinis – savarankiškai identifikuoja ir suformuluoja programų projektavimo etapus, atvaizduoja juos panaudodami diagramų kūrimo įrankį.</p> <p>Aukštesnysis – savarankiškai identifikuoja ir suformuluoja programų projektavimo etapus, sukuria veiksmingą planą etapams įgyvendinti ir tinkamai atvaizduoja viską panaudodami diagramų kūrimo įrankį.</p>
Žinios prieš	Geba kurti paprastas programas, paaiškinti jų veikimą.
Patarimai kolegoms, kurie naudos parengtą medžiagą	<ol style="list-style-type: none"> 1. Patys išbandykite „Draw.io“ veikimo principus ir galimybes. 2. Turėkite papildomų mokiniams teikiamų pavyzdžių. 3. Per pamoką mokiniai dirba savarankiškai, mokytojas yra pagalbininkas ir konsultantas.

1 ETAPAS (5-10 minučių). Trumpas „Draw.io“ programos galimybių, kurias mokiniai nagrinėjo namuose, aptarimas, savarankiško darbo užduoties formulavimas

Mokiniai namuose nagrinėjo Draw.io galimybes, apgalvojo, kaip pateiks programos projektavimo etapus panaudodami šią įrangą.

Mokytojo parengta medžiaga, pateikta mokinių darbui namuose.

- Interneto naršyklės paieškos lange įvedame užklausą „draw.io“.
- Atsidariusiame lange pasirenkame „Create new diagram“ (Kurti naują diagramą):



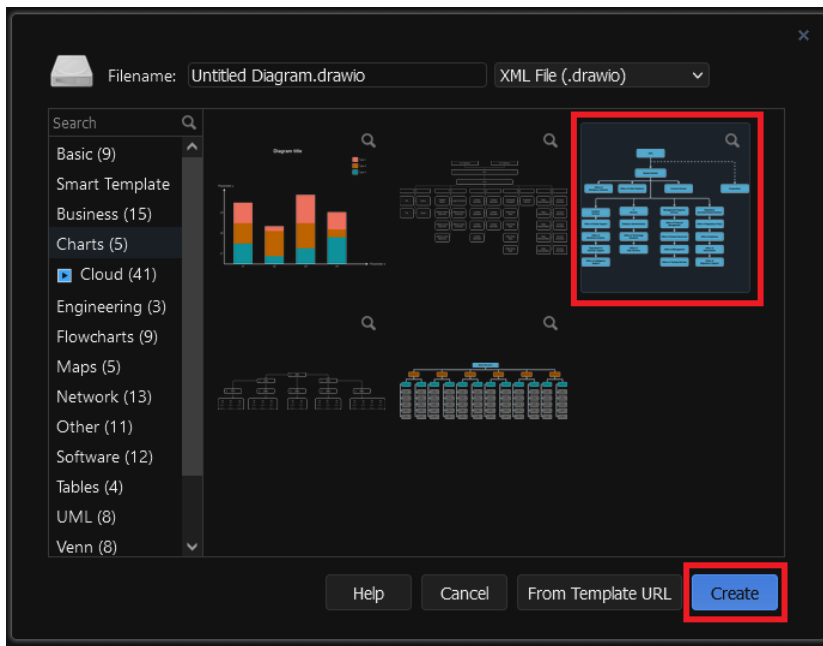
1 pav. „Draw.io“ diagramos kūrimas

- Mokiniams suteikiama galimybė pasirinkti jų požiūriu užduočiai atlikti labiausiai tinkantį diagramos tipą:



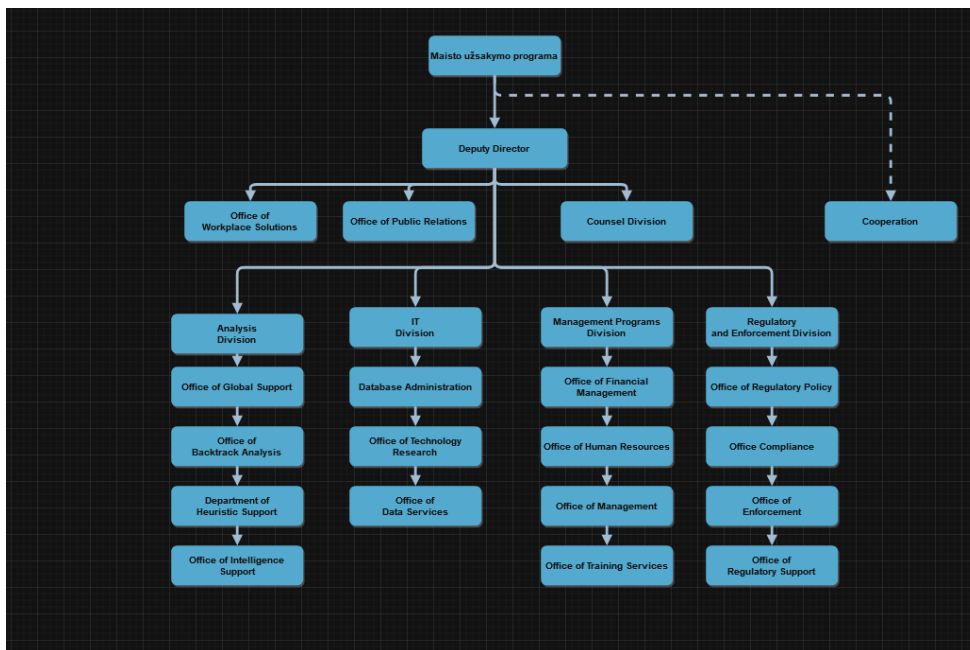
2 pav. „Draw.io“ diagramų tipai

- Pasirinkus diagramos tipą spaudžiame mygtuką „Create“ (Sukurti):



3 pav. „Draw.io“ diagramos tipo pasirinkimas

- Atsidariusiame lange pasirodo pasirinkta diagrama ir jos redagavimo parinktys:



4 pav. „Draw.io“ pasirinkta diagrama

2 ETAPAS (25-30 minučių). Praktinis tyrimas

Atlikus ir aptarus abi užduoties dalis mokiniams skiriama užduotis sukurti problemos identifikavimo, formulavimo ir projektavimo plano diagramą pasitelkus diagramų kūrimo priemonę „Draw.io“. Rekomenduojamas darbas grupėmis arba poromis.

3 ETAPAS (5 minutės). Darbų vertinimas, įšivertinimas.

Vertinimas: kai viena grupė pristato savo darbą, kitos stebi ir vertinimo lentelėje įrašo savo vertinimus balais, pagal iš anksto apibrėžtus kriterijus. Po to parašo atsiliepimus: 2 pagyrimus ir 2 patarimus ir pasirašo.

Kriterijus	1	2	3	4	5
Problemos formulavimas					
Reikalavimai					
Projektavimo etapų detalizavimas					

Po pamokos ir vertinimo su mokiniais apibendrinamas atliktas darbas.

5. Mokiniai tarpusavyje pasidalina įspūdžiais kaip pavyko atlikti savarankiškus darbus.
6. Mokiniai įsivertina pasirinktą diagramos tipą ir jo tinkamumą užduočiai atlikti.
7. Mokytojas paaiškina mokiniams, kad tai tik pavyzdinė užduotis ir realiuose projektuose reikalavimai ir etapai gali skirtis, bet pagrindiniai pamokoje aptarti programų projektavimo problemų formulavimo ir planavimo principai leidžia sistemingai organizuoti darbą ir užtikrinti, kad bet kokia programa bus sukurta efektyviai ir atitinkant visus poreikius.