

## Programavimo pradmenų modulio kūrybinių darbų ir jų vertinimo pavyzdžiai

### **Programos „Ūkininkas“ sukūrimas**

Ūkininkas nuvažiavo į turgų ir bando parduoti trys karves:  $a$ ,  $b$  ir  $n$ . Iš pradžių sužinojo jų svorius  $a$ ,  $b$ ,  $n$  ir užsirašė, pagal tai jis nusprendė, kad pirmiausiai reikia parduoti vidutinio svorio karvę, tada liesiausią ir tada pačią storiausią. Karvės yra skirtingo svorio.

Pirkėjai vidutiniškai prieidavo kas 30 min. Pirmą karvę nusipirko trečiasis pirkėjas, antrą – devintas pirkėjas, po pirmosios pardavimo, paskutinę – pirmasis pirkėjas, po antrosios pardavimo.

**Tikslas:** sukurti programą, kuri apskaičiuotų kas kiek laiko vidutiniškai ūkininkas pardavė kiekvieną karvę ir per kiek laiko jis pardavė visas karves?

### **Algoritmas:**

- Patikriname įvestų duomenų seka pagal jų didumą.
- Žinodami, kuris pirkėjas nuperka kada apskaičiuojame laiką, kurį truko parduoti visas karves.
- Surandame pardavimo laiko vidurkį.
- Apskaičiuojame per kiek laiko pardavė visas karves.

### **Naudojamos konstrukcijos:**

Naudojami sąlygos sakiniai **if**, ciklas **while**.

### **Darbo atlikimo tvarkaraštis:**

1. Užduoties kūrimas.
2. Darbo atlikimo planavimas.
3. Darbo atlikimas.
4. Pristatymas.

## Galimas užduties sprendimas C++ kalba

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
    int a, // karvė a.
        b, // karvė b.
        n, // karvė n.
        p, // pirkėjas, kuris nuperka karvę.
        l1=0, // laikas, po kurio buvo nupirktą pirma karvė.
        l2=0, // laikas, po kurio buvo nupirktą antra karvė.
        l3=0, // laikas, po kurio buvo nupirktą trečia karvė.
        l=0; // laikas, po kurio buvo parduotos visos karvės.
        // (prilyginame nuliui, kad nustatyti pradinę kintamojo vertę)

    // įrašome sužinotą kiekvienos karvės svorį.

    cout<<"Irasykite karves a svori"<<endl;
    cin>>a;
    cout<<"Irasykite karves b svori"<<endl;
    cin>>b;
    cout<<"Irasykite karves n svori"<<endl;
    cin>>n;
    p=0; //
    // suzinome, kuria karve reikia parduot paskutine.
    {
        if(a>b&& a>n) cout<<"Paskutine parduodame karve a"<<endl;
        else
        {
            if (b>a&&b>n) cout<<"Paskutine parduodame karve b"<<endl;
            else
            {
                if (n>a&&n>b) cout<<"Paskutine parduodame karve n"<<endl;
                else cout<<0;
            }
        }
    }
    // suzinome kuria karve reik parduot antra.
    {
        if(a<b&&a<n) cout<<"Antraja parduodame karve a"<<endl;
        else
        {
            if (b<a&&b<n) cout<<"Antraja parduodame karve b"<<endl;
            else
            {
                if (n<a&&n<b) cout<<"Antraja parduodame karve n"<<endl;
                else cout<<0<<endl;
            }
        }
    }

    // sužinome, kuria karve reikia pardavinėti pirma.
    {
        if ((a < b || a < n) && (a > b || a > n)) {
            cout << "Pirmaja parduodame karve a" << endl;
        }
        if ((b < a || b < n) && (b > a || b > n)) {
            cout << "Pirmaja parduodame karve b" << endl;
        }
        if ((n < b || n < a) && (n > b || n > a)) {
            cout << "Pirmaja parduodame karve n" << endl;
        }
    }
}
```

```

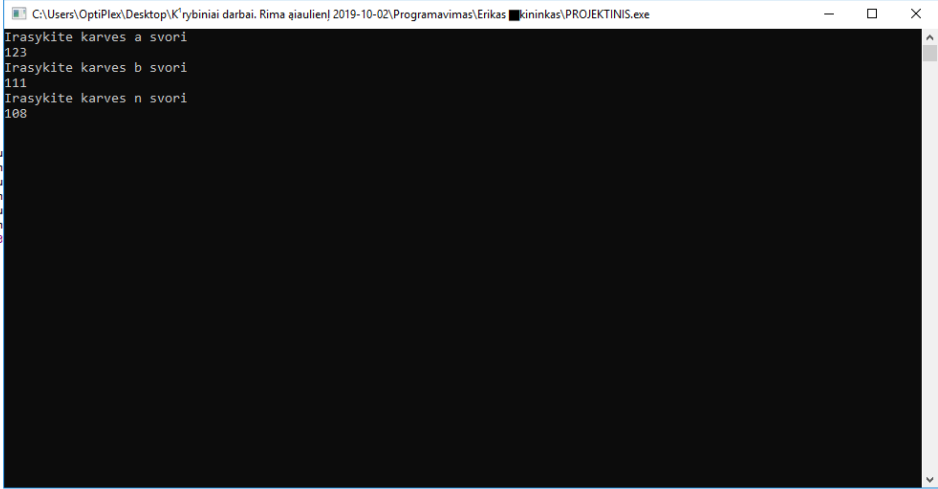
}
// zinodami, kuris pirkejas nusipirko sia karve suzinome po kiek lako parduota
// pirmoji karve.
{
    while (p<3)
    {
        p=p+1;
        l1=l1+30;
    }
    cout<<"Pirma karve parduota po "<<l1<<" minuciu"<<endl;
}
p=0; // prilyginame nuliui, kad atnaujinti kintamojo reiksme, jog panaudoti kitame
// cikle.
// zinodami, kuris pirkejas nusipirko sia karve suzinome po kiek laiko parduota
// antroji karve.
{
    while (p<9)
    {
        p=p+1;
        l2=l2+30;
    }
    cout<<"Antra karve parduota po "<<l2<<" minuciu"<<endl;
}
p=0;
// zinodami, kuris pirkejas nusipirko sia karve suzinome po kiek laiko parduota
// trecioji karve.
{
    while (p<1)
    {
        p=p+1;
        l3=l3+30;
    }
    cout<<"Trecia karve parduota po "<<l3<<" minuciu"<<endl;
}
// suzinoje po kiek bus parduotos karves, kiekvienos karves laika pardavime
// sudedame ir gauname galutini atsakyma.
l=l1+l2+l3;
cout<<"Visos karves buvo parduotos po "<<l<<" minuciu"<<endl;
}

```

Programos „*Ūkininkas*“ tekstą galite rasti faile PROGRAMAMA\_Ūkininkas.zip

## Programos fragmentai

```
1 #include <iostream>
2 #include <iomanip>
3 using namespace std;
4 int main()
5 {
6     int a, // karve a.
7         b, // karve b.
8
9
10    Irasykite karves a svori
11    123
12    Irasykite karves b svori
13    111
14    Irasykite karves n svori
15    108
16
17    //
18    cout<<endl;
19    cin>>a;
20    cout<<endl;
21    cin>>b;
22    cout<<endl;
23    cin>>n;
24    p=0;
25    //
26    {
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
```



```
Irasykite karves a svori
123
Irasykite karves b svori
111
Irasykite karves n svori
108
Paskutine parduodame karve a
Antraja parduodame karve n
Pirmaja parduodame karve b
Pirma karve parduota po 90 minuciu
Antra karve parduota po 270 minuciu
Trecia karve parduota po 30 minuciu
Visos karves buvo parduotos po 390 minuciu

-----
Process exited after 34.68 seconds with return value 0
Press any key to continue . . .
```

**Programos „Ūkininkas“ vertinimas**

<b>Eil. Nr.</b>	<b>Vertinimo kriterijai</b>	<b>Kriterijaus aprašas</b>	<b>Vertinimas</b>	
<b>1</b>	<b>Bendrujų gebėjimų vertinimas</b>			
1.1	Išsikelti tikslus.	Savarankiškai kelia klausimus, formuluoja užduotį, pasirenka sprendimo būdus.	1	<b>2 (2)</b>
		Mokytojo padedamas, formuluoja užduotį.	0	
		Savarankiškai pasirenka užduoties sprendimo būdus.	1	
		Mokytojo padedamas, pasirenka užduoties sprendimo būdus.	0	
1.2	Planuoti išteklius.	Savarankiškai pasirenka išteklius užduočiai atlikti, analizuoja problemą ir daro išvadas, apibendrina, modeliuoja sprendimo būdus; apibendrina, vertina rezultatus.	4	<b>4 (4)</b>
		Randa atsakymus į paprastus klausimus įvairiuose šaltiniuose, atrenka, įvertina informaciją, naudoja analogijas.	2	
		Padedamas mokytojo ar draugų, randa atsakymus į paprastus klausimus įvairiuose šaltiniuose, bando taikyti turimas žinias problemai spręsti.	0	
1.3	Sudaryti darbų tvarkaraštį.	Savarankiškai planuoja veiklą užduočiai atlikti, sudaro darbų tvarkaraštį.	4	<b>4 (4)</b>
		Atlieka paprastas užduotis, naudodamas nurodytas priemones, naudodamasis detaliuoju planu ir mokytojo pagalba.	2	
		Dirba pagal mokytojo sudarytą darbų tvarkaraštį.	0	
<b>Vertinimas taškais</b>			<b>10 (10)</b>	

<b>Eil. Nr.</b>	<b>Vertinimo kriterijai*</b>	<b>Kriterijaus aprašas</b>	<b>Vertinimas</b>	
<b>2</b>	<b>Dalykinių (programavimo) gebėjimų vertinimas</b>			
2.1	Duomenų tipų naudojimas.	Programoje naudojami duomenys atitinka užduoties sąlygą (realieji ir (ar) sveikieji skaičiai). Pagal sąlygą reikia naudoti abiejų tipų duomenis, o naudojami tik vieno tipo duomenys.	2 1	<b>2 (2)</b>
2.2	Naudojami veiksmai.	Teisingai atliekami veiksmai pradiniam duomenims įvesti. Teisingai atliekami veiksmai rezultatams išvesti.	1 1	<b>2 (2)</b>
2.3	Programoje naudojamos konstrukcijos.	Teisingas (teisingi) priskyrimo sakiny (sakiniai). Teisinga nuosekli veiksmų seka. Teisingas veiksmų šakojimas (sąlyginis sakiny). Veiksmų šakojimo sakinyje (sąlyginiame sakinyje) yra klaidų. Teisinga veiksmų kartojimo (ciklo) antraštinė dalis. Veiksmų kartojimo sakinyje (cikle) yra klaidų. Programoje naudojamos konstrukcijos neteisingos.	1 1 2 1 2 1 0	<b>6 (6)</b>
2.4	Algoritmų naudojimas.	Programoje naudojamas algoritmas (algoritmai), tinkantis pasirinktai problemai (užduočiai) spręsti. Programoje naudojamas algoritmas (algoritmai) tik iš dalies tinka pasirinktai problemai (užduočiai) spręsti. Programoje naudojamas algoritmas (algoritmai) netinka pasirinktai problemai (užduočiai) spręsti.	6 1-3 0	<b>6 (6)</b>

Eil. Nr.	Vertinimo kriterijai*	Kriterijaus aprašas	Vertinimas	
<b>2</b>	<b>Dalykinių (programavimo) gebėjimų vertinimas</b>			
2.5	Programoje atliekami aritmetiniai veiksmai.	Teisingai atliekami veiksmai su sveikaisiais skaičiais. Veiksmuose su sveikaisiais skaičiais yra klaidų. Teisingai atliekami veiksmai su realiaisiais skaičiais. Veiksmuose su realiaisiais skaičiais yra klaidų. Visi programoje atliekami veiksmai neteisingi.	<b>4</b> 1–3 <b>4</b> 1–3 0	<b>4**</b> <b>(4)</b>
2.6	Programos derinimas ir testavimas.	Programai derinti ir testuoti tinkamai parinkti kontroliniai duomenys. Tik dalis parinktų kontrolinių duomenų tinka programai derinti ir testuoti. Programa nebuvo derinama ir testuojama.	<b>2</b>  1 0	<b>2 (1)</b>
2.7	Programos struktūra.	Tinkama antraštė. Gerai aprašyti kintamieji. Išskirta pagrindinė programos dalis. Yra programos pabaiga.	<b>1</b> <b>1</b> <b>1</b> <b>1</b>	<b>4 (3)</b>
2.8	Programavimo kultūra.	Programos kintamųjų vardai parinkti prasmingai. Tik dalis kintamųjų pavadinti prasmingai. Programos veiksmai tinkamai komentuojami lietuvių (gimtąja) kalba. Programos veiksmai nėra komentuojami. Programa parašyta laikantis programavimo kultūros reikalavimų ir bendrųjų rašybos taisyklių. Programoje yra keletas neesminių programavimo kultūros reikalavimų arba bendrųjų rašybos taisyklių pažeidimų. Programa parašyta nesilaikant programavimo kultūros reikalavimų ar (ir) bendrųjų rašybos taisyklių.	<b>1</b> 0 <b>1</b> 0 <b>2</b> 1 0	<b>4 (2)</b>
<b>Vertinimas taškais</b>			<b>30</b>	<b>(26)</b>

\* Atsižvelgiant į mokinio darbo ypatumus, galima keisti kriterijus ir jų vertinimą iki 10 proc. taškų.

\*\* Jei naudojami ir sveiko, ir realaus tipo duomenys, bendras įvertinimas neturi viršyti 4 taškų.