

Funkcijos

Išmoksite:

- skaidyti algoritmus į smulkesnes, savarankiškas ir užbaigtas dalis;
- griežtai ir lakoniškai užrašyti programos dalis;
- užrašyti kreipinius į funkcijas, naudoti jas reiškiniuose

Sudarant C++ **funkciją** reikia numatyti:

- **pavadinimą** – pagal jį f-ja atpažįstama;
- **argumentus**, programavime vadinamus *formaliaisiais parametrais* (jie perduoda argumentų reikšmes funkcijai)
- **funkcijos veiksmus** (juos atliekant apsk. f-jos reikšmė)
- **funkcijos rezultatus**;
- **būdą, kuriuo rezultatas perduodamas iš funkcijos programai**

Funkcijos aprašymas (1)

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

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

Funkcijos aprašymas (2)

(natūraliojo skaičiaus n daliklių skaičius)

```
int DalSk (int n) {  
    int k = 0;  
    for (int d = 1; d <= n; d++)  
        if (n % d == 0)  
            k++;  
    return k;  
}
```

```

#include <iostream>
#include <iomanip>
#include <cstdlib>
using namespace std;
// -----funkcijų prototipai -----
double StPlotas(double x, double y);
// -----programos pradžia -----
int main ()
{
    // ----- kintamieji -----
    double a, b;
    double S;
    setlocale(LC_ALL, "Lithuanian");
    cout << "Įveskite pirmosios dalies matmenis a ir b ";
    cin >> a >> b;
    // -----skaičiavimai -----
    S = StPlotas(a, b) + StPlotas(2*a, 3*b);
    // ----- Rezultatas -----
    cout << "Sklypo plotas = " << S << " kv. m" << endl;
    return 0;
}
// -----funkcijų aprašai -----
double StPlotas(double x, double y) {
    double s = x * y;
    return s;
}

```

