

# Masyvas

Tai viena pagrindinių duomenų struktūrų, naudojamų programavime. Ši struktūra leidžia laikyti atmintyje vienu vardu daug to paties tipo reikšmių.

# Tikslai:

- išmokti aprašyti masyvą;
- suteikti reikšmes masyvo elementams;
- masyvo elementų reikšmes naudoti skaičiavimuose

# Masyvas

- *Masyvas* – viena pagrindinių duomenų struktūrų, naudojamų programavimo kalbose.
- Ši struktūra leidžia laikyti atmintyje (vienoje vietoje) vienu vardu daug to paties tipo reikšmių.
- Pavyzdžiui, žinomos vieno mėnesio pokalbių telefonu trukmės minutėmis. Reikia apskaičiuoti, kiek iš viso minučių prakalbėta, ilgiausio pokalbio trukmę, ilgiausių pokalbių skaičių ir kt.
- Rašant programą bet kurio mėnesio pokalbių telefonu trukmėms apdoroti, negalima žinoti, kiek pokalbių bus tą mėnesį. Todėl negalima iš anksto numatyti, kiek kintamųjų reikia pokalbių trukmėms laikyti. Be to, atskirų pokalbių trukmėms naudojant skirtingus kintamuosius, programa bus gremėzdiška ir neuniversali (kito mėnesio duomenims apdoroti programą reikės taisyti).
- Masyve visas reikšmes galima laikyti vienu vardu. Reikšmių skaičius masyve gali būti skirtingas.

# Pagrindinės sąvokos

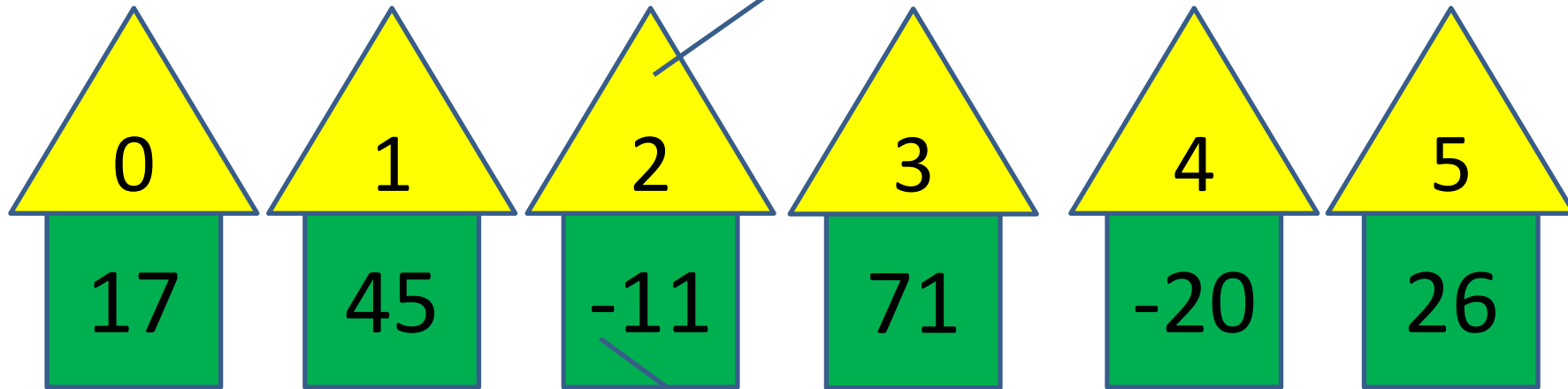
- **Masyvas** – įvardyta atminties sritis, kurioje laikomos masyvo elementų reikšmės;
- **Elementas** – vienai masyvo reikšmei skirta vieta kompiuterio atmintyje;
- **Reikšmės tipas** – visos masyvo elementų reikšmės yra to paties tipo;
- **Reikšmė** – tai kas įrašyta į masyvo elementą;
- **Indeksas** – masyvo elemento numeris, pradedant skaičiuoti nuo masyvo pradžios.

Masyvo reikšmės  
tipas

Masyvo elementų skaičius

`int Mas[6]`

Masyvo indeksas



Masyvo reikšmė

Pradinė indekso reikšmė yra lygi nuliui, paskutinė – vienetu mažesnė už masyvo ilgį

# Masyvų įvairovė

- **Loginio** (*bool*) tipo masyvai
- **Sveikųjų ir realiųjų skaičių** masyvai
- **Simbolių** (*char*) masyvai
- **Simbolių eilučių** masyvai
- **Struktūrų** masyvai

# Svarbu:

- Elemento indeksas gali būti sveikasis skaičius, sveikojo tipo kintamasis arba reiškinys, kurio rezultatas yra sveikasis skaičius. Pavyzdžiui:

```
A[12]; A[i]; A[i + 2]; B[k + 6 * i];
```

- Masyvo elementų reikšmių tipas gali būti bazinis (int, double, char ir kt.) arba programuotojo sukurtas.
- Masyve galima laikyti ne daugiau reikšmių, negu nurodyta jo apraše. Paskutinio elemento indeksas – vienetu mažesnis už masyvo dydį. Todėl rašant programas labai svarbu sekti, kad indekso reikšmė visuomet būtų intervale  $[0, \text{masyvoDydis} - 1]$ .
- Programoje galima aprašyti ir naudoti keletą skirtingo tipo ir skirtingo dydžio masyvų. Jų vardai programos bloke turi būti skirtingi.
- Masyvo dydis apraše nurodo kompiliatoriui, kiek vietos atmintyje reikia skirti masyvo elementų reikšmėms.

# Masyvo aprašymas

```
int atstumas[25];  
double pinigai [12];
```

```
const int CIlgis = 100;  
double M[CIlgis];
```

# Reikšmių suteikimas masyvo elementams

```
int T[7] = {5, 8, 0, -7, -11, -1, 4}
```

```
int P[] = {15, 18, -11, -1}
```

```
T[0] = 5;
```

```
T[1] = 8;
```

```
T[2] = 0;
```

```
.....
```

```
T[6] = 4;
```



# Masyvo užpildymas duomenimis iš tekstinio failo

```
# include <iostream>
# include <cstdlib>
# include <fstream>
const int CIlgis = 25;
const char FV[] = "k1.txt";
using namespace std;
int main ()
{
    int U[CIlgis];    int n;
    ifstream fd (FV);
        fd >> n;
    for (int i = 0; i < n; i++)
        fd >> U[i];
    for (int i = 0; i < n; i++)
        cout << U[i] << " ";
    cout << endl;
    return 0;
}
```

```
5
180 185 176 169 189
```

# Skaitymo funkcija

```
void Skaityti(char fv[], int D[], int & k)
{
    ifstream fd(fv);
    fd >> k;
    for (int i = 0; i < k; i++)
        fd >> D[i];
    fd.close();
}
```

# Sumos skaičiavimo funkcija

```
int Suma(int A[], int n)
{
    int suma = 0;
    for (int i = 0; i < n; i++)
        suma = suma + A[i];
    return suma;
}
```