

# Wstęp do programowania

## - wykład 10

dr Piotr Jastrzębski

# Tablice w C++

Cechy:

- ▶ mają stały rozmiar
- ▶ są modyfikowalne (można zmienić poszczególne elementy)
- ▶ mogą przechowywać elementy tylko jednego typu
- ▶ elementy posiadają pozycję

## Deklaracja

```
Typ nazwa_tablicy [liczba elementów];
```

np.

```
int tab[4];  
char tab2[3];  
float tabf[5];
```

```
#include <iostream>

using namespace std;

int main()
{
    int a[10];
    for (int i = 0; i < 10; ++i)
    {
        a[i] = i + 1;
    }
    int b[10]={ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
    return 0;
}
```

## Problemy:

- ▶ bezpośrednio tablice nie posiadają informacji o swoim rozmiarze
- ▶ kompilacja umożliwia operowanie poza zadeklarowanym obszarem tablicy
- ▶ ograniczona ilość komend względem języka Python
- ▶ brak możliwości przypisywanie tabel (obejście za pomocą wskaźników, których nie będzie omawiać)
- ▶ nowe podejście do tablic `vector`, `array`

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

int main()
{
    int tab[4];
    for (int i = 0; i < 4; i++)
    {
        cout << tab[i] << endl;
    }
    return 0;
}
```

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

int main()
{
    int tab[4] = { 3,5 };
    for (int i = 0; i < 4; i++)
    {
        cout << tab[i] << endl;
    }
    return 0;
}
```

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

int main()
{
    int tab[4] = { 3,5,1,4,6,7 };
    for (int i = 0; i < 4; i++)
    {
        cout << tab[i] << endl;
    }
    return 0;
}
```

Ten kod w większości wypadków się nie kompiluje(!).



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

int main()
{
    int tab[] = { 3,5,1,4,6,7 };
    for (int i = 0; i < 6; i++)
    {
        cout << tab[i] << endl;
    }
    return 0;
}
```

Problematyczna konstrukcja:

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

int main()
{
    int tab[] = { 3,5,1,4,6,7 };
    for (int i = 0; i < sizeof tab / sizeof(int); i++)
    {
        cout << tab[i] << endl;
    }
    return 0;
}
```

## Stałe tablice

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

int main()
{
    const int tab[4] = { 1,-3,4,5 };
    tab[2] = 5;
    return 0;
}
```

Zmiana wartości nie jest możliwa!

```
#include <iostream>
#define ROZMIAR 3
using namespace std;

int main()
{
    int tab[ROZMIAR] = { 3,-4,3 };
    for (int i = 0; i < ROZMIAR; i++)
    {
        cout << tab[i] << endl;
    }
    return 0;
}
```

```
#include <iostream>
#define ROZMIAR 3
using namespace std;

int main()
{
    int tab[ROZMIAR] = { 3,-4,3 };
    cout << tab[-1] << endl;
    cout << tab[ROZMIAR] << endl;
    return 0;
}
```

# Tablice a funkcje

- ▶ mogą być jedynie argumentem
- ▶ przekazując tablicę - nie przekazujemy jej rozmiaru
- ▶ nie może być typem zwracanym

https:

[//gist.github.com/pjastr/ac3c9a75e60c07d39c3921269c54feef](https://gist.github.com/pjastr/ac3c9a75e60c07d39c3921269c54feef)

# Klasyczne zadania

- ▶ zwrócenie najmniejszej/największej wartości
- ▶ sprawdzenie ile elementów spełnia określony warunek
- ▶ sortowanie
- ▶ ...