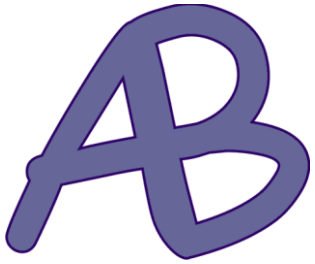
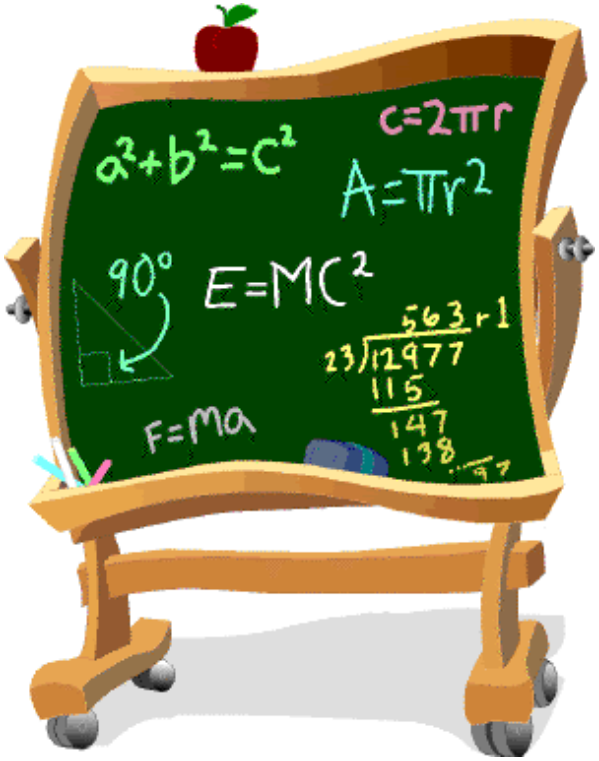


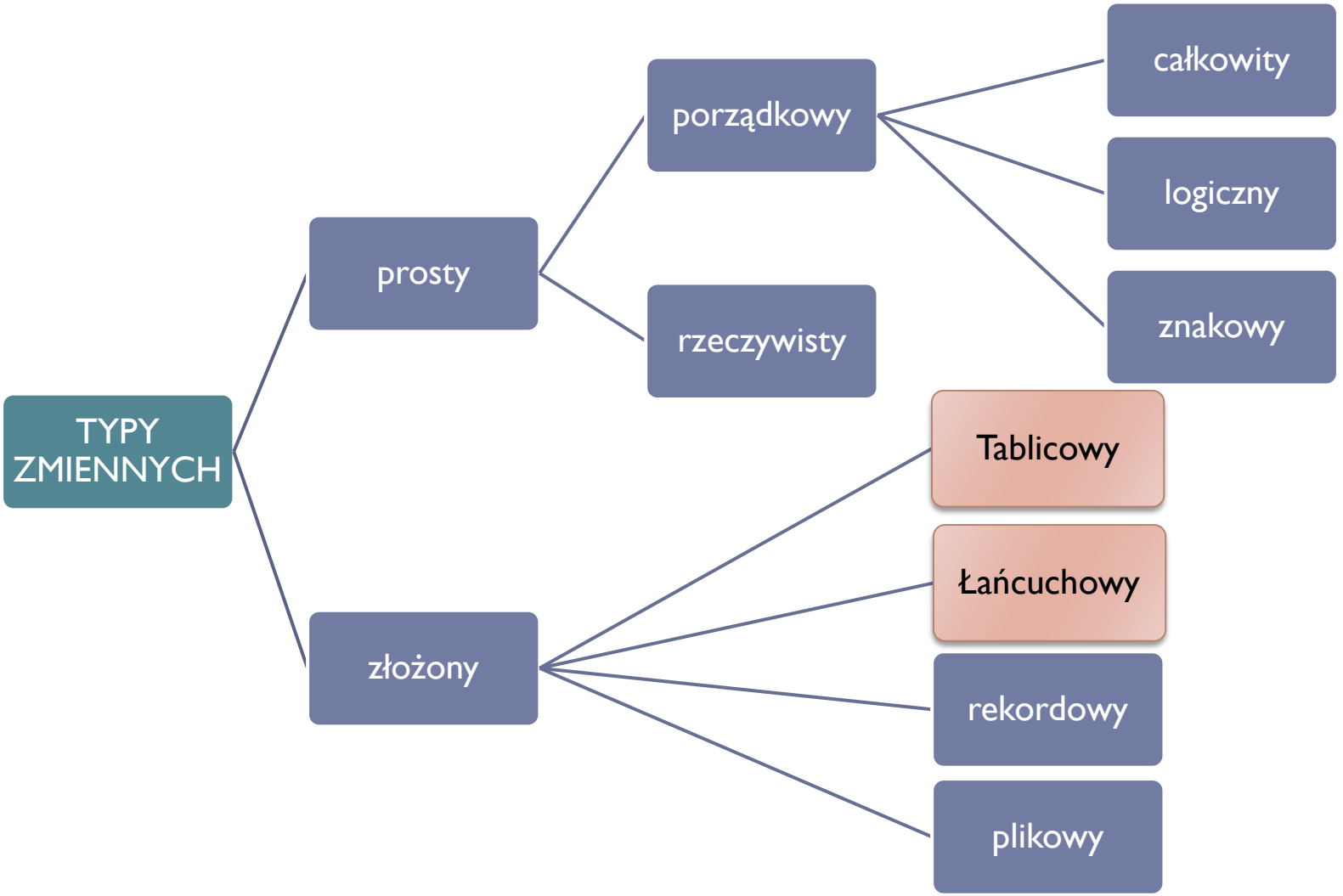
Wykład: 4

Tablice statyczne

Tablice



Klasyfikacja zmiennych statycznych



Rodzaje tablic

Tablice jednowymiarowe (wektory)

- są zespołem określonej liczby zmiennych o wspólnej nazwie, które ponumerowano liczbami naturalnymi – każda z nich ma przypisany na stałe tzw. indeks,
- mogą przechowywać nie większą od ich długości liczbę elementów zbioru danych jednakowego typu.

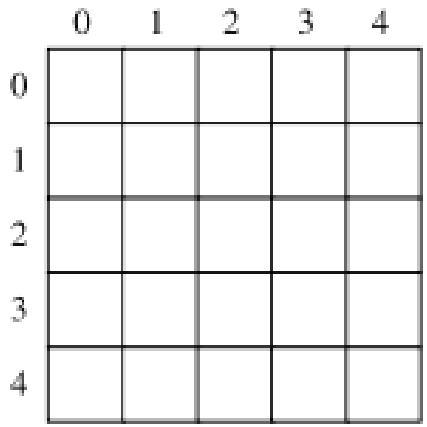
tab								- nazwa tablicy
34	56	32	-8	45	2	...	13	- wartości
tab[0]	tab[1]	tab[2]	tab[3]	tab[4]	tab[5]	...	tab[n-1]	- pole – nazwa[indeks]

W zapisie symbolicznym T[6[oznacza 6 zmienną w tablicy T Indeks może być określony przez bezpośrednie podanie wartości w odwołaniu do elementu tablicy, np. T[6], lub użycie nazwy zmiennej o typie zgodnym z indeksem, np. T[X] Zmienną X nazywamy wtedy zmienną indeksową i wskazanie elementu tablicy wymaga odczytania jej aktualnej wartości.

Rodzaje tablic

Tablice dwu – i więcej wymiarowe (macierze)

- są zespołem określonej liczby zmiennych o wspólnej nazwie, które oznaczono dwoma lub więcej indeksami,
- mogą przechowywać nie większą od ich rozmiaru liczbę elementów zbioru danych jednakowego typu.

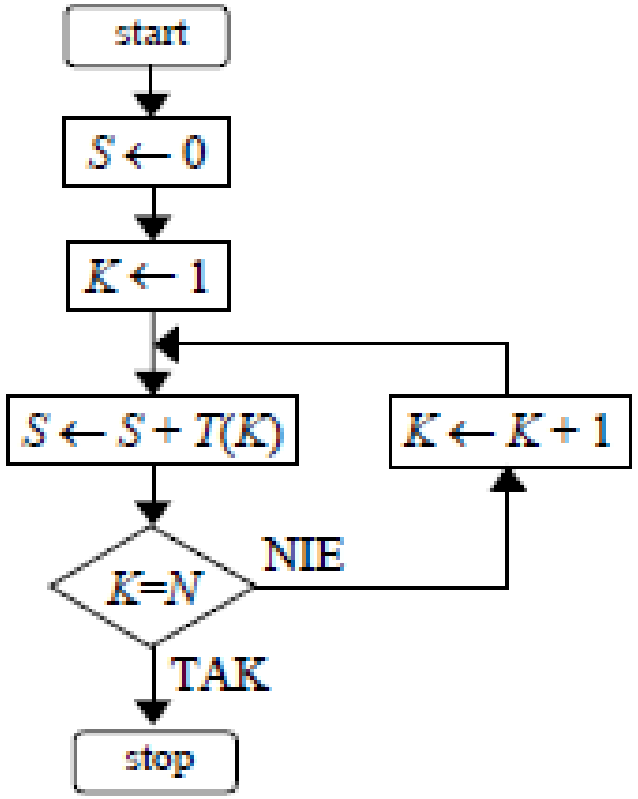


W zapisie symbolicznym W[3][5] oznacza zmienną w tablicy W położoną umownie na przecięciu 3 wiersza i 5 kolumny.

Obsługa tablicy jednowymiarowej

Algorytm sumowania N liczb zapamiętanych w tablicy T

1. $S \leftarrow 0$ (ustalenie początkowej wartości sumy);
2. $K \leftarrow 1$ (ustalenie początkowej wartości zmiennej indeksowej);
3. wykonaj co następuje N razy:
   - 3.1. $S \leftarrow S + T(K)$;
   - 3.2. $K \leftarrow K + 1$.



Tablice

Tworzenie jednowymiarowych tablic zmiennych - za deklaracją zmiennej podamy liczbę elementów.

```
typ_zmiennej nazwa_zmiennej [liczba_elementow];
```

liczba_elementów - musi być wartością stałą dosłowną, lub stałą const

Np.:

```
int Tablica[ 10 ];
```

Lub:

```
const STALA = 10;
int Tablica[ STALA ];
```

Tablice

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

int main()
{
    int liczbyCalkowite[5];
        // Definicja tablicy pieciu liczb całkowitych
    double liczbyRzeczywiste[3];
        // Definicja tablicy trzech liczb zmiennoprzecinkowych
    return 0;
}
```

Tablice

Tablice można tworzyć z:

- ✓ typów fundamentalnych (z wyjątkiem void),
- ✓ typów wyliczeniowych (enum),
- ✓ wskaźników,
- ✓ innych tablic;
- ✓ obiektów typu zdefiniowanego przez użytkownika (czyli klasy),
- ✓ wskaźników do pokazywania na składniki klasy.

Tablice

Numeracja elementów tablicy zaczyna się od zera.

[0][1][2]⋯[*n*−1]

indeksy komórek n-*elementowej tablicy*

Jeśli zdefiniujemy tablicę:

```
int tab[5];
```

jest to tablica pięciu elementów typu int. Poszczególne elementy tej tablicy to:

```
tab[0]  tab[1]  tab[2]  tab[3]  tab[4]
```

Inicjalizacja tablic

Inicjalizacja tablicy - nadanie wartości początkowych w momencie definicji tablicy.

Np.:

```
int tab[6] = {2, 3, 5, 7, 11, 13 } ;
```

Jest równoznaczne z:

```
tab[0]=2;  tab[1]=3;   tab[2]=5;
tab[3]=7;  tab[4]=11;  tab[5]=13;
```

Uwaga: zapis: cout << tab[6];
odnosi się do nieistniejącego elementu tablicy.

Inicjalizacja tablic

Przy zapisie:

int tab[] = {2, 3, 5, 7, 11, 13 };

kompilator „domyśli się”, że chodzi tablicę 6-cio elementową.

Przy zapisie:

```
int tab[6] = {2, 3, 5};
```

pierwsze trzy elementy zostaną zainicjalizowane podanymi wartościami, pozostałe zerami:

```
tab[0]=2;  tab[1]=3;  tab[2]=5;
tab[3]=0;  tab[4]=0;  tab[5]=0;
```

Inicjalizacja tablic

Uwaga: Code::Blocks dopuszcza konstrukcję:

```
typ_elementów nazwa_tablicy[zmienna];
```

pozwala ona na tworzenie statycznych tablic o liczbie elementów podanej w zmiennej. Na przykład:

```
int n;
cin >> n;
double a[n];
```

Nie jest to standardowe rozwiązanie i może nie być przenośne na inne kompilatory C++.

Obsługa tablicy jednowymiarowej

```
for (i=0; i < rozmiar; i++) ……….…;
```

Przykład:
Dana jest 6-cio elementowa tablica,
Wypisz jeśli wartość w tablicy jest parzysta

```
int tab[6] = {1,2,3,4,5,6};
for(int i=0;i<5;i++)
     if(tab[i]%2==0)
        cout<<tab[i]<<" ";
```

Inicjalizacja tablic

Przykład: Lotto – losowanie 6 liczb bez powtórzeń z zakresu 1-49

```
int main()
{
    srand(time(NULL));
    bool powt = false;
    int tab[6];
    int i = 0, x;
    do
    {
        powt = false;
        x = rand() % 48 + 1;
        for(int j=0; j<i; j++)
           {
           if(x==tab[j]) powt = true;
           }
        if (powt==true) continue;
        tab[i] = x;
        i++;
    } while (i < 6);

    for (int j = 0; j < 6; j++)
        cout << tab[j] << " ";
    return 0;
}
```

Przekazywanie tablic do funkcji

NAZWA TABLICY

jest równocześnie

ADRESEM JEJ ZEROWEGO ELEMENTU

Dla: int tab[10];

zapis: tab jest równoznaczny z &tab[0]

Przekazując tablicę do funkcji w rzeczywistości przekazujemy funkcji wskaźnik do tej tablicy.

Przekazywanie tablic do funkcji

Tablicy nie można przesłać przez wartość.

Można tak przesłać pojedyncze jej elementy, ale nie całość.

Mamy funkcję o nagłówku:

```
void funkcja   (float tab[]);
```

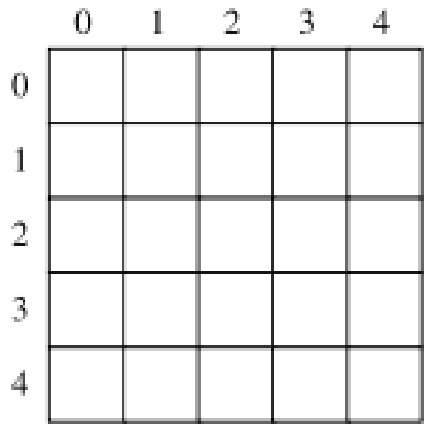
która spodziewa się jako argumentu: tablicy liczb typu float, Taką funkcję wywołujemy na przykład tak:

```
float tablica[4]={ 7, 8.1, 4, 4.12};
funkcja (tablica);
```

Tablice wielowymiarowe

Tablice dwu – i więcej wymiarowe (macierze)

- są zespołem określonej liczby zmiennych o wspólnej nazwie, które oznaczono dwoma lub więcej indeksami,
- mogą przechowywać nie większą od ich rozmiaru liczbę elementów zbioru danych jednakowego typu.



W zapisie symbolicznym ***W*(3, 5)** oznacza zmienną w tablicy ***W*** położoną umownie na przecięciu 3. wiersza i 5. kolumny.

Tablice wielowymiarowe

W języku C++ tablice wielowymiarowe to tablice, których elementami są inne tablice.

```
int tab_2D[n][m];
```

Definicja ta oznacza: tab_2D jest tablicą n-elementową, z których każdy jest m-elementową tablicą (liczb typu int).

Uwaga: zapis ~~int tab_2D[n , m]~~ jest błędny.

Tablice wielowymiarowe

Przykład:

```
int tab_2D[4][3];
```

Gdzie: tab_2D jest tablicą 4-elementową, z których każdy jest 3-elementową tablicą liczb typu int.

[4][3]	0	1	2
0	[0] [0]	[0] [1]	[0] [2]
1	[1] [0]	[1] [1]	[1] [2]
2	[2] [0]	[2] [1]	[2] [2]
3	[3] [0]	[3] [1]	[3] [2]

Tablice wielowymiarowe

Elementy takie umieszczane są kolejno w pamięci komputera tak, że najszybciej zmienia się najbardziej skrajny prawy indeks.

Stąd, inicjalizacja zbiorcza:

```
int tab[3][2]  =  {1,2,3,4,5,6};
```

spowoduje, że elementom tej tablicy zostaną przypisane wartości początkowe tak, jakbyśmy to robili grupą instrukcji:

```
tab[0][0]   = 1;
tab[0][1]   = 2;
tab[1][0]   = 3;
tab[1][1]   = 4;
tab[2][0]   = 5;
tab[2][1]   = 6;
```

Tablice wielowymiarowe

Obsługa tablicy dwuwymiarowej:

```
#include <iostream>

using namespace std;
int tab[5][3] = { 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};
int main()
{

    for (int j=0; j<5; j++)
    {

      for (int i=0; i<3; i++)
       cout << tab[j][i>] << "   ";
       cout << "\n";
    }
    return 0;
}
```

```
1   2   3
4   5   6
7   8   9
10   11   12
13   14   15
```

Tablice wielowymiarowe

Przykład: tabliczka mnożenia

```
int main()
{
    int tab[11][11];
    for (int j = 0; j < 11; j++)
        for (int i = 0; i < 11; i++)
            tab[j][i] = j * i;

    for (int j = 0; j < 11; j++)
    {
        for (int i = 0; i < 11; i++)
            cout << tab[j][i] << "\t";
        cout << endl;
    }
    return 0;
}
```

Wypełnienie tablicy

Wypisanie zawartości tablicy

Literatura:

W prezentacji wykorzystano przykłady i fragmenty:

- Gręboszl J. : ***Symfonia C++, Programowanie w języku C++ orientowane obiektowo***, Wydawnictwo Edition 2000.
- Jakubczyk K.: *Turbo Pascal i Borland C++ Przykłady*, Helion.

Warto zajrzeć także do:

- Sokół R. : ***Microsoft Visual Studio 2012 Programowanie w Ci C++,*** Helion.
- Kerninghan B. W., Ritchie D. M.: ***język ANSI C***, Wydawnictwo Naukowo Techniczne.

Dla bardziej zaawansowanych:

- Gręboszl J. : ***Pasja C++,*** Wydawnictwo Edition 2000.
- Meyers S.: ***język C++ bardziej efektywnie***, Wydawnictwo Naukowo Techniczne