C++

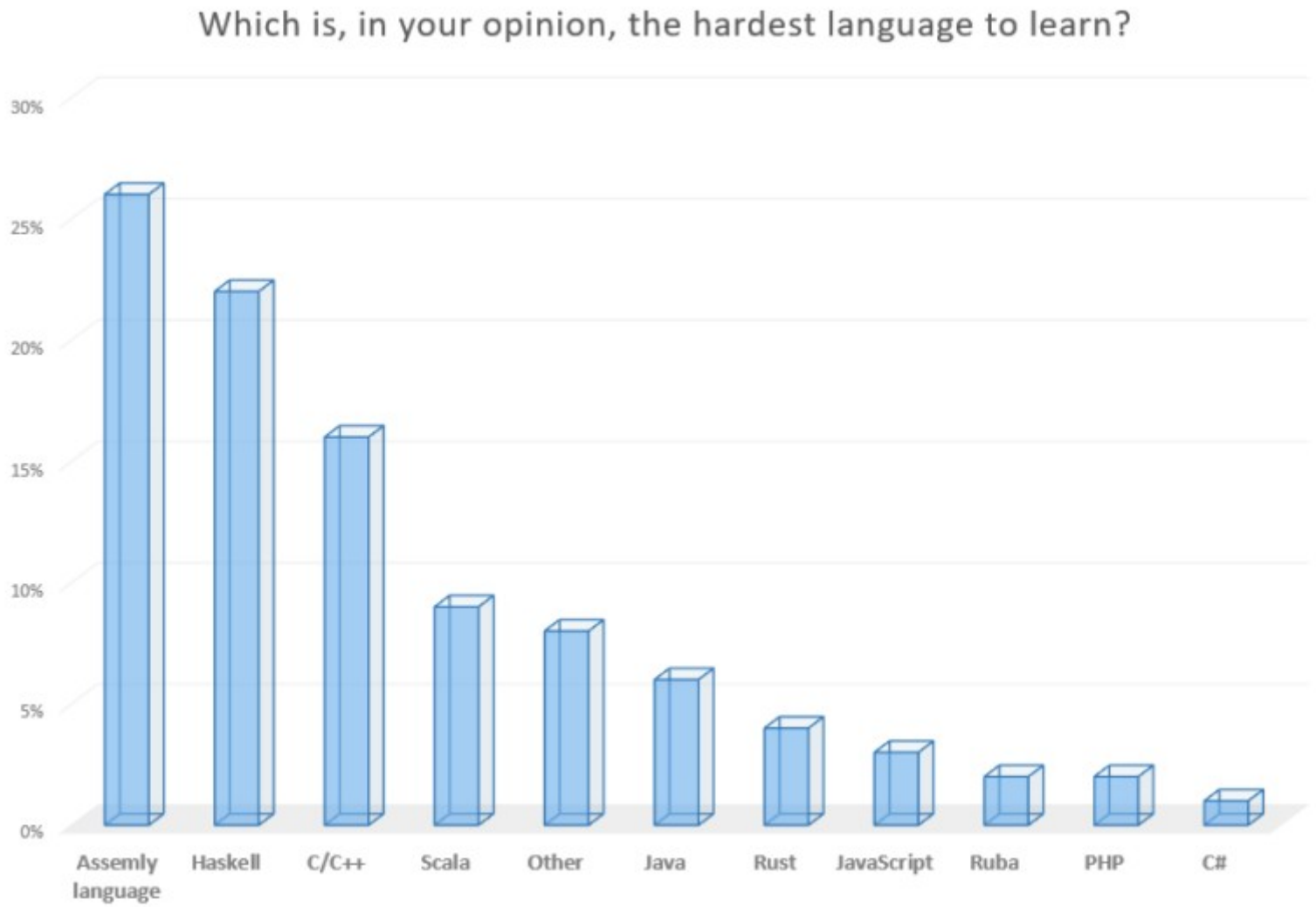
wprowadzenie - część II

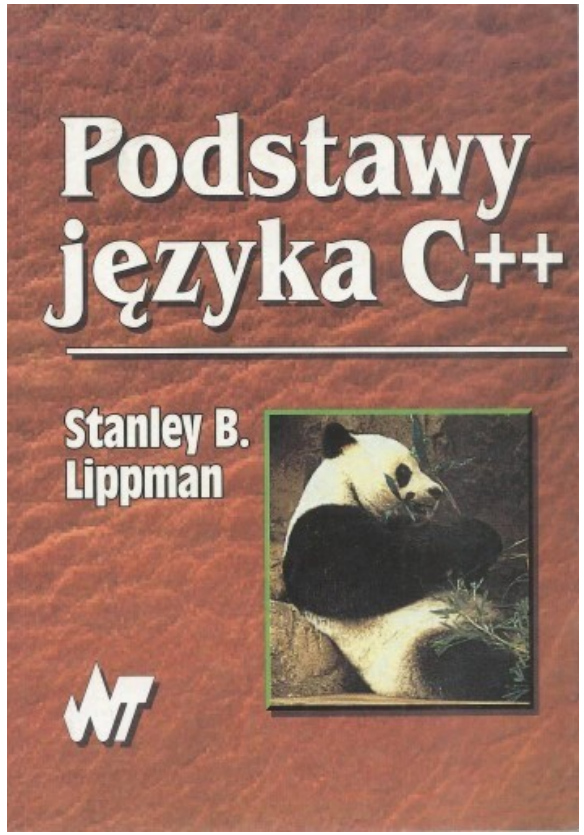
O czym będzie

- O podstawach...

- i nie tylko...

Trudność uczenia się – wynik ankiet





https://jaxenter.com/most-difficult-programming-languages-152590.html

Dyrektywy preprocesora

```
#include <iostream>
#include <przyklad.hpp>

#define QUESTION "What is your name?"
#define DIM 256


#define zmienna
...
...
#if defined zmienna
    // fragment do kompilacji jeśli
    // 'zmienna' jest zdefiniowane
#else
    // fragment do kompilacji jeśli
    // 'zmienna' nie jest zdefiniowane
#endif


#ifndef PLIKH_H
#define PLIKH_H

#endif
```

Hello World!

Typy danych i ich rozmiar

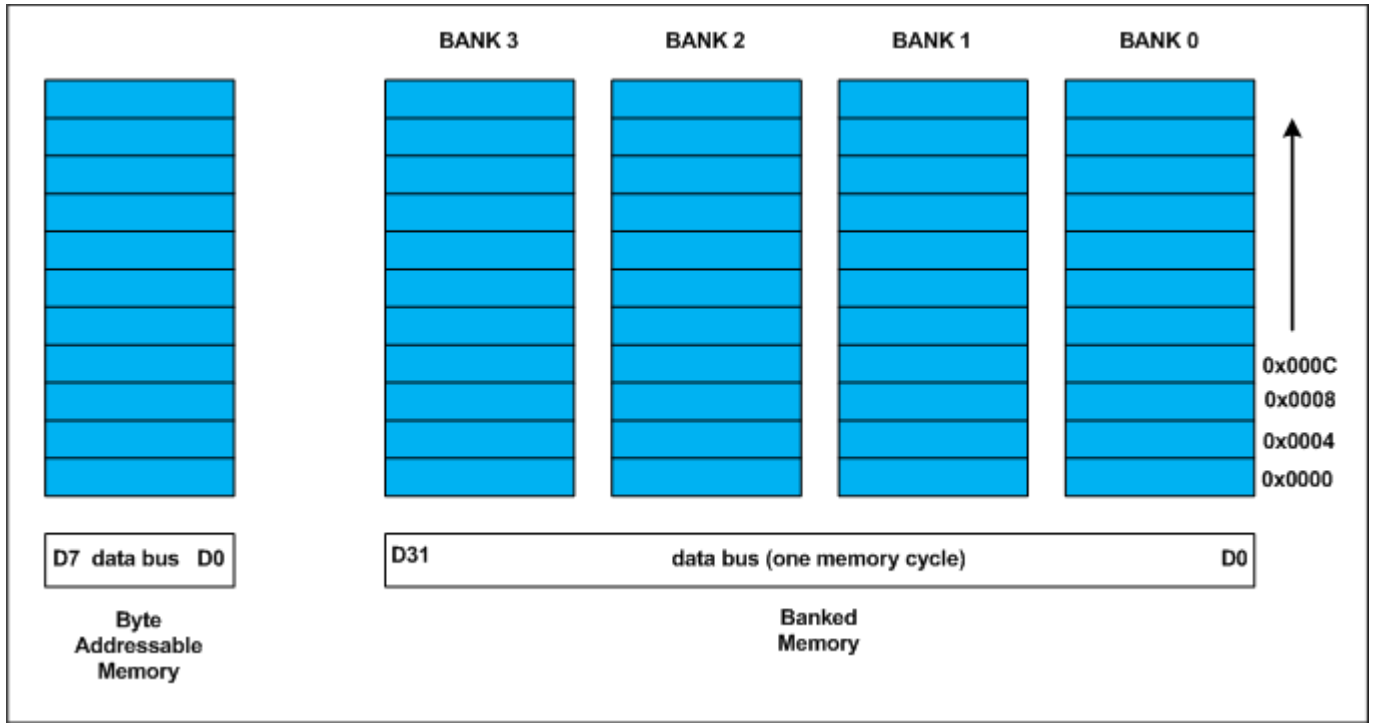
- char - unsigned char
- short - unsigned short
- int - unsigned int
- long - unsigned long

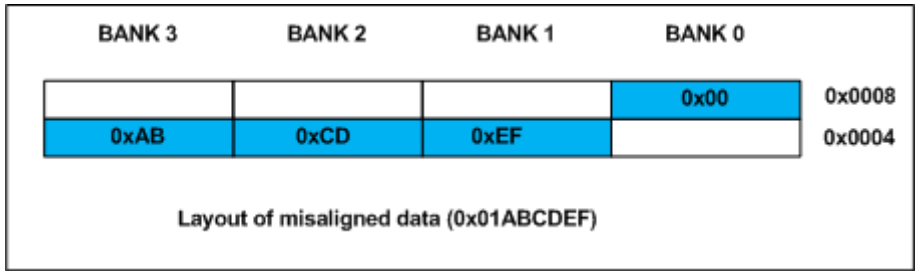
- float
- double
- long double

Struktury i ich rozmiar

- sizeof(...)
- sumowanie łatwe nie jest :P

Wyrównanie pamięci





Wyrównanie pamięci

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13
ExtraS	c	padding			s.total				s.a				s.b				d	padding		
ExtraP	c	p.total					p.a				p.b				d					

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24
ExtraS	c	padding			s.total				s.a				s.b				d	padding			c	padding			s.total				s.a				s.b				d
ExtraP	c	p.total					p.a				p.b				d		c	p.total					p.a				p.b				d								

Instrukcja warunkowa if

- Iloczyn logiczny - &&
- Suma logiczna - ||
- Negacja - !

- Prawo de Morgana

  !(a && b) = !a || !b

  !(a || b) = !a && !b

```
if (warunek)
{
    instrukcja_1;
    instrukcja_2;
    ...
    instrukcja_n;
}
else
{
    instrukcja_1;
    instrukcja_2;
    ...
    instrukcja_n;
}
```

Instrukcja warunkowa switch

- warto używać #define
- lub komentarzy
- czytelność przede wszystkim
- problem zmiennych

- default
- break

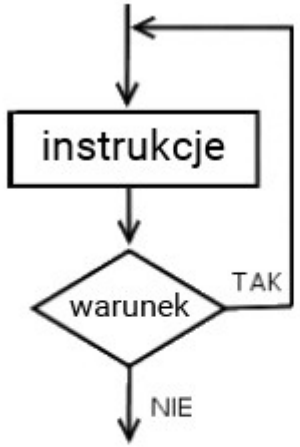
```
switch( zmienna )
{
case wartosc_1:
    //jakiś kod
    break;

case wartosc_2:
    //jakiś kod
    break;

    //...
case wartosc_n:
    //jakiś kod
    break;

default:
    //jakiś kod
    break;
}
```

Pętla for

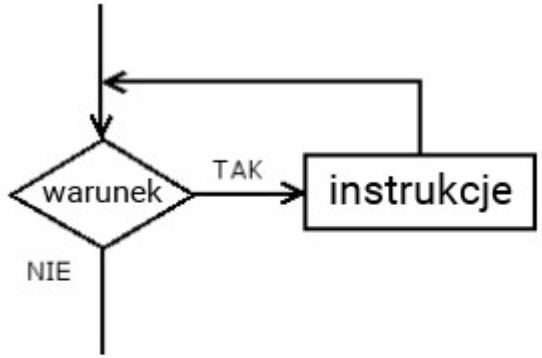
- break, continue

```
for(int i = 1; i <= 10; i++)
{
    [instrukcje, które będą się powtarzać]
}

for(int i = 10; i >= 1; i--)
{
    [instrukcje, które będą się powtarzać]
}

for(int i = 1; i <= 10; i = i + 2)
{
    cout << endl << i;
}

for(int i=1; i<=10; i=i+2); // średnik zmienił działanie programu!
{
    cout << endl << i;      // wykona się zawsze i to dokładnie jeden raz!
}
```

Pętla while, oraz do...while



to jest pętla do..while, gdyż najpierw jest instrukcja, zaś potem sprawdzamy warunek



to jest pętla while, bo najpierw sprawdzamy jest warunek i tylko jeśli jest spełniony wykona się instrukcja

```
while(strzal != liczba)
{
    [instrukcje realizujące odgadywanie liczby]
}

do
{
    [instrukcje realizujące odgadywanie liczby]
} while(strzal != liczba);
```

Praktyczne zastosowanie pętli while

Pętlę **while** w praktyce stosuje się zawsze tam, gdzie mamy potrzebę przetwarzania nieokreślonej liczby danych. Przykłady:

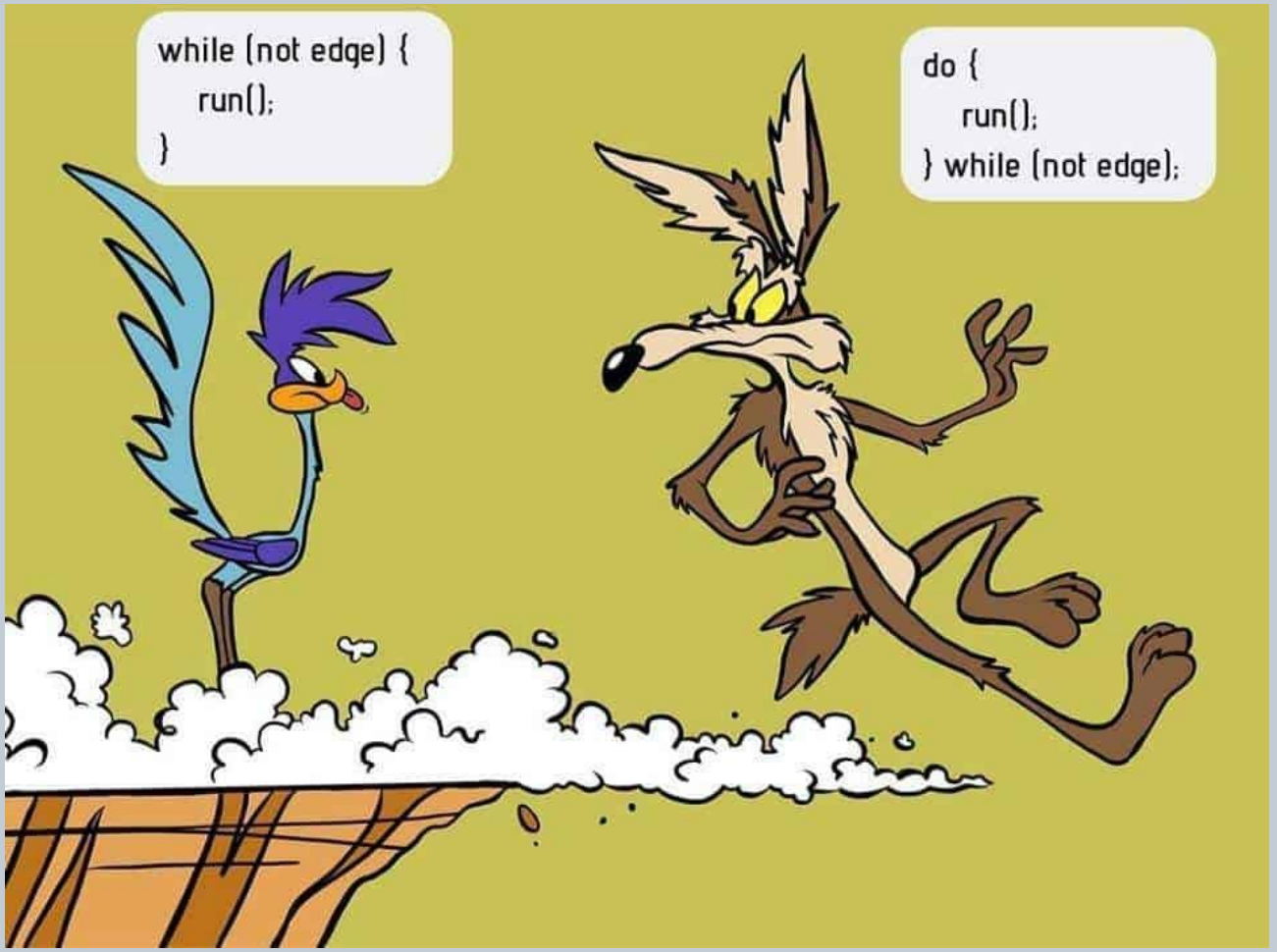
- chcesz wczytać dane do tablicy, ale liczbę wprowadzanych danych podaje np. użytkownik;
- chcesz odczytać sekwencyjnie zawartość całego pliku;
- chcesz zakończyć pętlę np. tylko wtedy gdy zostanie wciśnięty określony klawisz.

Funkcje

```
typ_zwracanej_wartosci nazwa_funkcji (typ_1 nazwa_1 /*,...*/, typ_n nazwa_n )
{
    return zwracana_wartosc;
}

void to_jest_moja_funkcja()
{
    //INFO: tu kod funkcji
}
```

Ciekawostki - prezentacja

PYTANIA ?



radoslaw.roszczyk@ee.pw.edu.pl

https://github.com/rroszczyk/cpp

https://github.com/rroszczyk/python