Programowanie C++

Wykład 1 (27.02.2017)

Sprawy organizacyjne

- Kontakt: Tomasz.Kazimierczuk@fuw.edu.pl
- 1h wykładu + 2h ćwiczeń / tydzień
- Ocena na podstawie punktów (skala standardowa)
  - 2 kolokwia przy komputerach (2 x 25%)
  - Test z wykładu (25%)
  - Egzamin z zadań (25%)
  - Aktywność na ćwiczeniach (10%)
- Uwaga: surowa punktacja. Na kolokwium program nie kompilujący się dostaje 0 punktów. Na teście punkty ujemne.
- Strona do wykładu: http://www.fuw.edu.pl/~tkaz/teaching/programowanie2017/

Sprawy organizacyjne

- Ćwiczenia
- Wykład
- **Internet!!!**

C++ jest językiem kompilowanym

1. Tworzymy kod źródłowy w pliku tekstowym

```cpp
#include <iostream>
using namespace std;
int main() {

  cout << "Hello world";

  return 0;
}
```

program.cpp

2. Kompilujemy program

```
$ g++ program.cpp -o program
```

3. Uruchamiamy program

```
$ ./program
Hello world
```

Logowanie na Primusa

Dom —— ssh ——> primus.okwf.fuw.edu.pl

putty 

```
ssh login@tempac.fuw.edu.pl
ssh login@primus.okwf.fuw.edu.pl
```

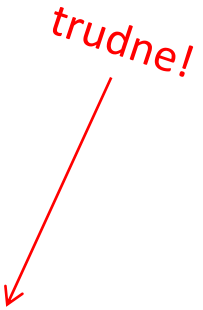
Kompilator: g++

Edytor: nano, pico

Wikipedia

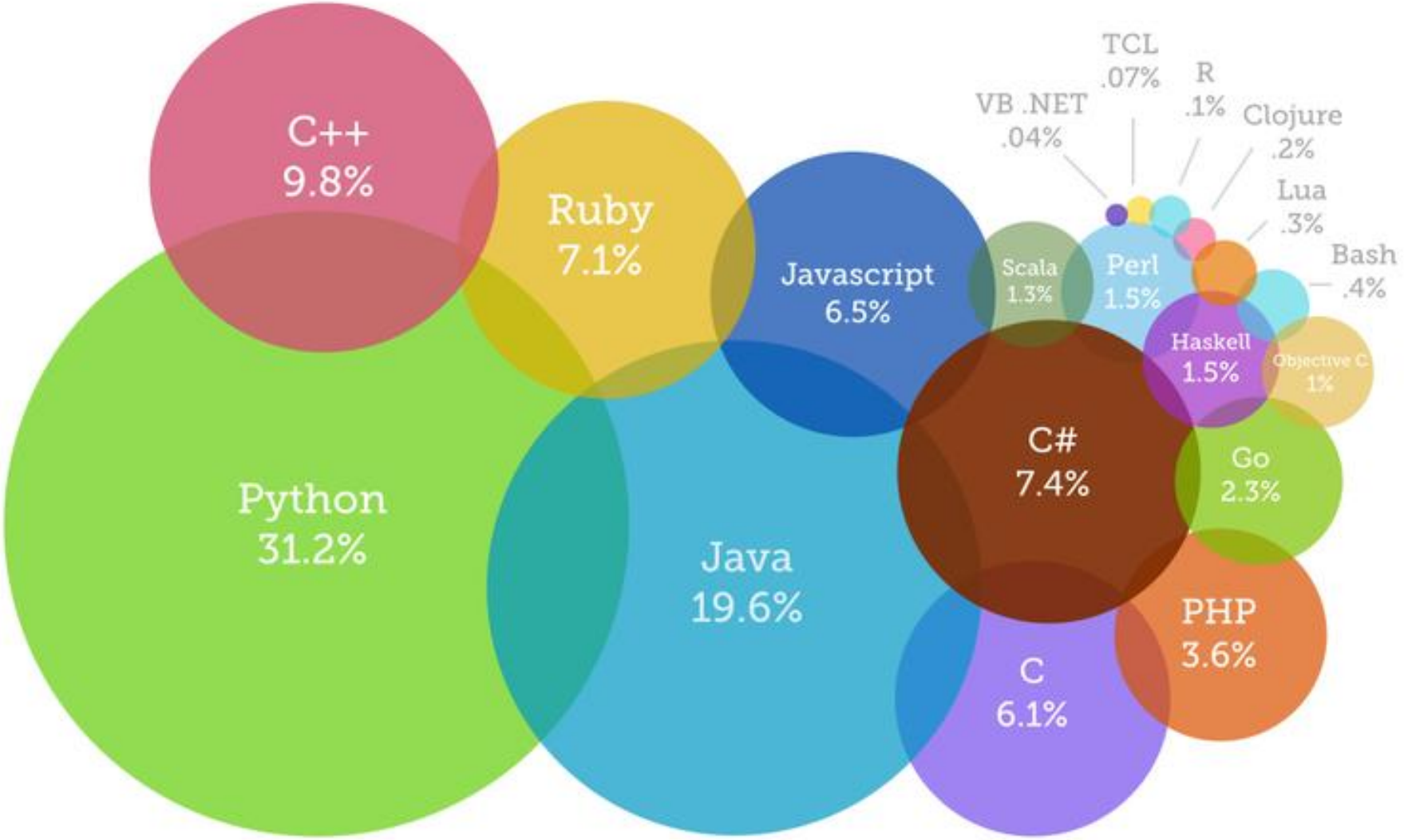
- **Programowanie komputerów** – proces projektowania, tworzenia, testowania i utrzymywania kodu źródłowego programów komputerowych (…)

Wikipedia

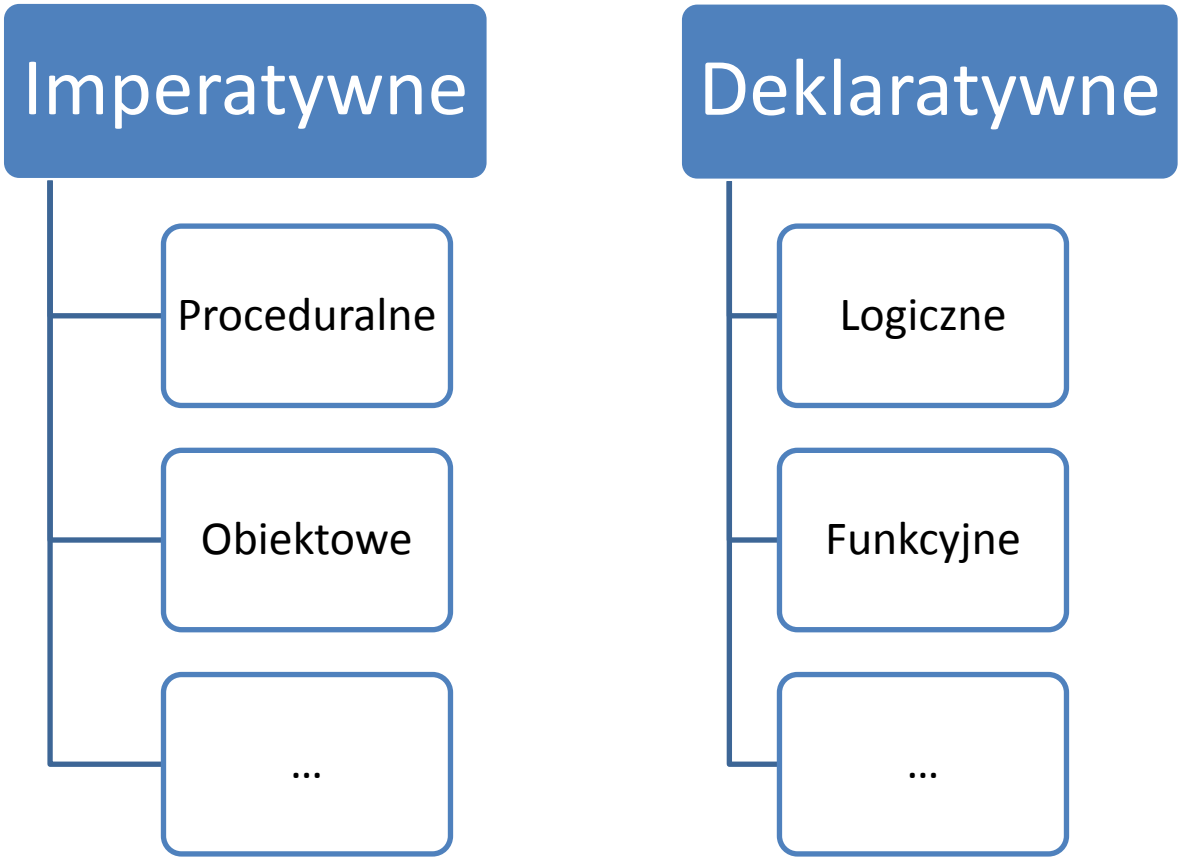


- **Programowanie komputerów** – proces projektowania, tworzenia, testowania i utrzymywania kodu źródłowego programów komputerowych (…)

Most Popular Coding Languages of 2015



Paradygmaty programowania



Przykład z Internetu

1. Z lodówki weź 10 jajek - połóż na stole ocalałe 7, wytrzyj podłogę, następnym razem uważaj!

2. Weź sporą miskę i wbij jajka rozbijając je o brzeg naczynia.

3. Wytrzyj podłogę, następnym razem bardziej uważaj! W naczyniu mamy 5 żółtek.

4. Weź mikser i wstaw do niego skrzydełka i zacznij ubijać jajka.

5. Wstaw od nowa skrzydełka do miksera, tym razem do oporu. Zacznij ubijać.

6. Umyj twarz, ręce i plecy. W naczyniu pozostały 2 żółtka,a dokładnie tyle potrzeba na szarlotkę.

13. Potnij jabłka w kostkę pamiętając, że potrzebujemy 2 jabłka, więc nie wolno zjeść więcej niż połowę! Przemyj jodyną palec wskazujący i środkowy.

14. Jedyne pozostałe jabłko pocięte w kostkę wrzuć do naczynia z ciastem, pozbieraj z podłogi pozostałe kawałki i przemyj wodą.

15. Wymieszaj wszystkie składniki w naczyniu mikserem, umyj lodówkę bo jak zaschnie to nie domyjesz!

16. Przelej ciasto do foremki, wstaw do piekarnika.

17. Po godzinie, jeśli nie widać żadnych zmian włącz piekarnik.

Przykład 2

```
function GreatestCommonDivisor(a, b: Int64): Int64;
  begin
    if (a < 0) then a := -a;
    if (b < 0) then b := -b;
    if (a = 0) then Exit(b);
    if (b = 0) then Exit(a);
    while not (a = b) do
    begin
      if (a > b) then
       a := a - b
      else
       b := b - a;
    end;
    Result := a;
  end;
```

Przykład: programowanie deklaratywne

```
SELECT Student.Imie, Student.Nazwisko, Ocena.Ocena
  FROM Student
  JOIN Ocena ON Student.id = Ocena.StudentId
  WHERE Ocena.Przedmiot = Mechanika kwantowa
   AND Ocena.Ocena > 2;
```

Przykład 2: programowanie funkcyjne

```
(define (sieve n)
  (define (aux u v)
    (let ((p (car v)))
      (if (> (* p p) n)
        (let rev-append ((u u) (v v))
          (if (null? u) v (rev-append (cdr u) (cons (car u) v))))
        (aux (cons p u)
          (let wheel ((u '()) (v (cdr v)) (a (* p p)))
            (cond ((null? v) (reverse u))
                  ((= (car v) a) (wheel u (cdr v) (+ a p)))
                  ((> (car v) a) (wheel u v (+ a p)))
                  (else (wheel (cons (car v) u) (cdr v) a)))))))))
  (aux '(2)
    (let range ((v '()) (k (if (odd? n) n (- n 1))))
      (if (< k 3) v (range (cons k v) (- k 2))))))
```

Programowanie imperatywne

- Kolejne instrukcje zmieniają stan (np. wartości zmiennych)

- Podstawowe typy zmiennych:

  `int` – liczba całkowita, np. 2235

  `double` – liczba rzeczywista, np. 3.14

  `char` – pojedynczy znak ASCII (litera)

  `bool` – wartość logiczna (prawda lub fałsz)

Prześledźmy program:

Standardowy początek

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

int main() {
   int a = 0;
   int b = 0;

   a = 8;
   b = 9 + a;
   b = b + b;


   cout << a << " " << b << "\n";

   return 0;
}
```

Wypisywanie wartości zmiennych

Standardowy koniec

Wyrażenia w C++

Operator: + (np. a + b)

Wynik: suma wartości *a* i *b*

Efekt uboczny: brak

Operator: = (np. a = b)

Wynik: wartość *b*

Efekt uboczny: zmienna *a* przyjmuje wartość *b*

Operator: ++ (np. a++)

Wynik: wartość *a*

Efekt uboczny: zmienna *a* zwiększa się o 1

a = 3; b = 5;

b = (a = (a++))

Ile wynosi a?
Ile wynosi b?

b = a = b+a++

A teraz?

Precedence	Operator	Description	Associativity
1	::	Scope resolution	Left-to-right
2	a++ a--	Suffix/postfix increment and decrement	
	type() type{}	Functional cast	
	a()	Function call	
	a[]	Subscript	
	. ->	Member access	
3	++a --a	Prefix increment and decrement	Right-to-left
	+a -a	Unary plus and minus	
	! ~	Logical NOT and bitwise NOT	
	(type)	C-style cast	
	*a	Indirection (dereference)	
	&a	Address-of	
	sizeof	Size-of[note 1]	
	new new[]	Dynamic memory allocation	
	delete delete[]	Dynamic memory deallocation	
4	.* ->*	Pointer-to-member	Left-to-right
5	a*b a/b a%b	Multiplication, division, and remainder	
6	a+b a-b	Addition and subtraction	
7	<< >>	Bitwise left shift and right shift	
8	< <=	For relational operators < and ≤ respectively	
	> >=	For relational operators > and ≥ respectively	
9	== !=	For relational operators = and ≠ respectively	
10	a&b	Bitwise AND	
11	^	Bitwise XOR (exclusive or)	
12	|	Bitwise OR (inclusive or)	
13	&&	Logical AND	
14	||	Logical OR	
15	a?b:c	Ternary conditional[note 2]	Right-to-left
	throw	throw operator	
	=	Direct assignment (provided by default for C++ classes)	
	+= -=	Compound assignment by sum and difference	
	*= /= %=	Compound assignment by product, quotient, and remainder	
	<<= >>=	Compound assignment by bitwise left shift and right shift	
	&= ^= |=	Compound assignment by bitwise AND, XOR, and OR	
16	,	Comma	Left-to-right

Porównywanie

- ==, !=, >, <, >=, <=

- Wykorzystywane głównie w instrukcjach kontrolnych, np.

```
if (a == b)
    cout << "Zmienne a i b sa rowne";


while (a > b)
{
    b++;
    a--;
}
```