

# Podstawy programowania



Piotr Zierhoffer

Institute of Computer Science  
Poznań University of Technology

7 października 2012

# Opis przedmiotu

---

- Wstęp do zapisu algorytmów
- Podstawy języka C
- Obsługa środowiska Dev-C++



# Kontakt

---

piotr.zierhoffer@cs.put.poznan.pl **(w temacie [PP])**

[www.cs.put.poznan.pl/pzierhoffer](http://www.cs.put.poznan.pl/pzierhoffer)

konsultacje: 9:45–10:15 BTiCW, Piotrowo 2, pokój 2.7.2

telefon: (+48) 61 665 29 41



# Wymagania

---

- Zadania domowe



# Wymagania

---

- Zadania domowe
- Kolokwium



# Wymagania

---

- Zadania domowe
- Kolokwium
- Niezerowa aktywność



# Wymagania

---

- Zadania domowe
- Kolokwium
- Niezerowa aktywność
- **Obecność**



# Wymagania

---

- Zadania domowe
- Kolokwium
- Niezerowa aktywność
- **Obecność**
- **Samodzielność**

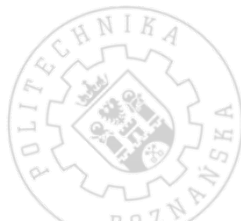




# Literatura

---

- Brian Kernighan, **Dennis Ritchie** — *Język ANSI C*
- Jerzy Grębosz — *Symfonia C++ Standard*
- Bruce Eckel — *Thinking in C*  
[<http://www.mindviewinc.com/CDs/ThinkingInC>]



# Algorytm

---

## Definition

**Algorytm** to skończony zbiór czynności, których wykonanie konieczne jest do osiągnięcia celu.



# Algorytm

---

## Definition

**Algorytm** to **skończony** zbiór **czynności**, których wykonanie konieczne jest do osiągnięcia **celu**.



# Algorytm

---

## Definition

**Algorytm** to **skończony** zbiór **czynności**, których wykonanie konieczne jest do osiągnięcia **celu**.

## Example

Gotowanie herbaty

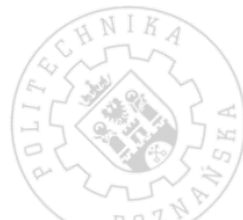
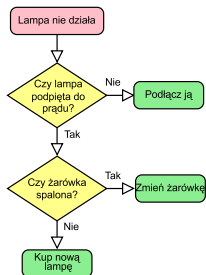
Dodawanie liczb

Szybka transformacja Fouriera



# Formy prezentacji

## 1. Schemat blokowy



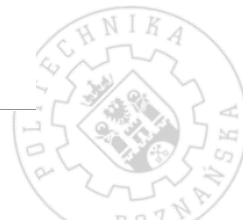
# Formy prezentacji

1. Schemat blokowy
2. Pseudokod

Lampa nie działa

## Algorytm DDMC

```
1: DDMC( $s, C, M$ )  
    $\Gamma(u)$  – sąsiedztwo wężła  $u$ .  
2: for each vertex  $v \in V$  do  
3:    $d(v) \leftarrow \infty$   
4:    $\pi(v) \leftarrow NULL$   
5: end for  
6:  $d(s) \leftarrow 0$   
7:  $S \leftarrow \emptyset$   
8:  $Q \leftarrow V$   
9: while  $Q \neq \emptyset$  do  
10:   $u = \text{PopMin}(Q)$   
11:   $S = S \cup \{u\}$   
12:  for each vertex  $v \in \Gamma(u)$  do  
13:    if  $d(v) > I_M(u)d(u) + w(u, v)$  then  
14:      if  $v \notin S$  then  
15:         $d(v) = I_M(u)d(u) + w(u, v)$   
16:         $\pi(v) = u$   
17:      end if  
18:    end if  
19:  end for  
20: end while
```



# Formy prezentacji

1. Schemat blokowy
2. Pseudokod
3. Kod programu

Lampa nie działa

## Algorytm DDMC

```
1: DDMC( $s, C, M$ )  
    $\Gamma(u)$  – sąsiedztwo wężła  $u$ .  
2: for each vertex  $v \in V$  do  
3:    $dl(v) \leftarrow \infty$   
4:   /* Park-Miller "minimal standard" 31 bit  
5:   ei * pseudo-random number generator, implemented  
6:    $d$  * with David G. Carta's optimisation: with  
7:    $S$  * 32 bit math and without division.  
8:    $Q$  */  
9:   w  
10:  long unsigned int rand31_next()  
11:  {  
12:    long unsigned int hi, lo;  
13:    lo = 16807 * (seed & 0xFFFF);  
14:    hi = 16807 * (seed >> 16);  
15:    lo += (hi & 0x7FFF) << 16;  
16:    lo += hi >> 15;  
17:    if (lo > 0x7FFFFFFF) lo -= 0x7FFFFFFF;  
18:    return ( seed = (long)lo );  
19:  }  
20: ei
```

# Własności algorytmów

---

## 1. skończoność





# Własności algorytmów

---

1. skończoność
2. określoność



# Własności algorytmów

---

1. skończoność
2. określoność
3. postulat wejścia



# Własności algorytmów

---

1. skończoność
2. określoność
3. postulat wejścia
4. postulat wyjścia



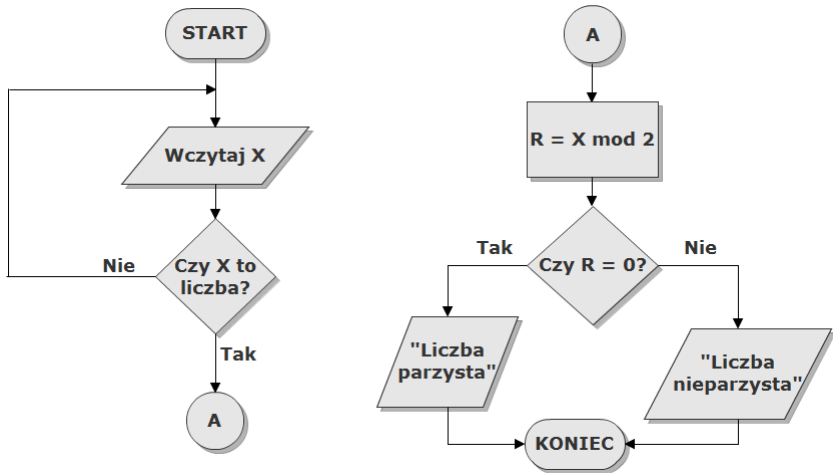
# Własności algorytmów

---

1. skończoność
2. określoność
3. postulat wejścia
4. postulat wyjścia
5. efektywność



# Badanie parzystości liczby



# Zadanie: algorytm zaparzania herbaty

---

Określ dane wejściowe

Zdefiniuj cel

Opracuj schemat



Zadanie: określ zależność dwóch liczb ( $>$ ,  $<$ ,  $=$ )

---

Jakie dane można przyjąć?

Jakie warunki trzeba zastosować?



## Zadanie: rozwiązanie równania liniowego

---

$$ax + b = 0$$

Zadane współczynniki





## Zadanie: rozwiązanie układu dwóch równań liniowych

---

$$a_1x + b_1y = c_1$$

$$a_2x + b_2y = c_2$$



# Zadanie domowe

---

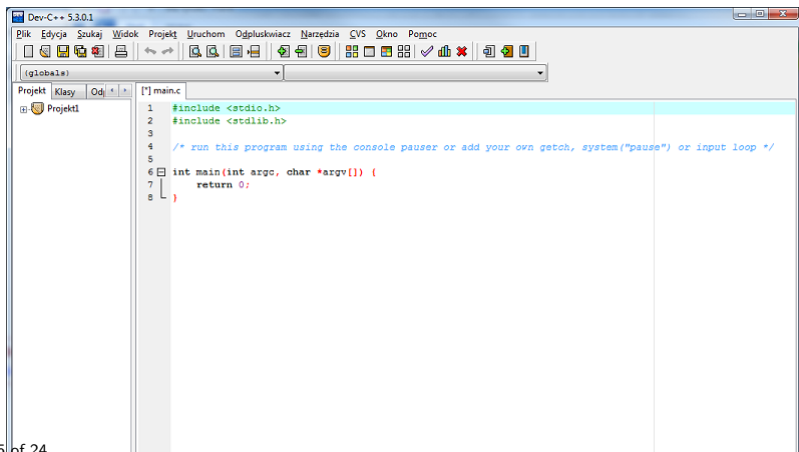
1. Algorytm Euklidesa
2. Obliczanie pierwiastków równania  $ax^2 + bx + c = 0$
3. Wyliczenie silni liczby



# Język C

Środowisko programistyczne Dev-C++.

<http://sourceforge.net/projects/orwelldevcpp/>



# Struktura programu

---

```
#include <stdio.h>

int main()
{
    int i;
    i = 5;
    //Ale o co tu chodzi?
    printf("Witaj swiecie!\n");
    return 0;
}
```



# Typy

---

char

int

short int

long int

long long int

const

float

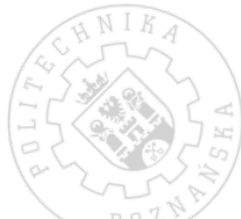
double

long double

void

signed

unsigned



# Operacje na zmiennych

---

```
int main()
{
    int a;
    int b = 5;
    a = b;
    b = b + 1;

    unsigned int u = 5u;

    float c = 0.5;
    c = b / c;

    char q = 'Q' - 2;
    return 0;
}
```



# Zadanie

---

Zapoznaj się z typem wyliczeniowym **enum**!



# Zakresy ważności zmiennych

---

```
int a = 3;  
{  
    int a = 4;  
    a++;  
}  
a = a + 1;
```





# Zakresy ważności zmiennych

---

```
int a = 3;
{
    int a = 4;
    a++;
}
a = a + 1;
```

```
int a = 3;
{
    a = 4;
    a++;
}
a = a + 1;
```



# Wypisywanie na ekran, wczytywanie danych

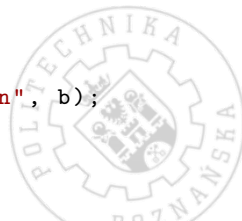
---

## Funkcje printf i scanf

```
#include <stdio.h>
int main()
{
    int a = 5;
    printf("Moja_liczba_to_%d\nPodaj_swoja!\n",
           a);

    int b;
    scanf("%d", &b);

    printf("Podana_liczba_to_%d\n", b);
    printf("Ta_liczba_szesnastkowo_to_0x%X\n", b);
}
```



# Zadania

---

1. Wczytaj dwie liczby całkowite i wypisz na ekran ich sumę, różnicę, iloczyn i iloraz. Wypróbuj z liczbą **0**!
2. Wypisz tekst na ekran stosując znaki specjalne `printf`.
3. Wypisz liczbę  $\pi$  z różną precyzją (nagłówek `math.h`, stała `M_PI`).



## Zadanie domowe

---

1. Zapoznaj się z możliwościami `printf` i `scanf`.
2. Wczytaj trzy liczby zmiennoprzecinkowe i wypisz różnicę każdej pary liczb.



Dziękuję za uwagę!

---

