Politechnika Wrocławska

Wstęp do programowania

INP001213Wcl

rok akademicki 2018/19

semestr zimowy

Wykład 1

Karol Tarnowski

karol.tarnowski@pwr.edu.pl

A-1 p. 411B

Politechnika Wrocławska

Plan wykładów (1)

- Algorytmy i programy

- Proste typy danych
- Rozgałęzienia i iteracje
- Funkcje
- Tablice i wskaźniki
- Złożone typy danych

Politechnika Wrocławska

Plan wykładów (2)

- Rekurencja
- Złożoność czasowa algorytmów
- Zasada dziel i zwyciężaj
- Algorytmy porządkowania
- Przeszukiwanie z nawrotami
- Poprawność i skończoność algorytmów
- Złożoność i efektywność algorytmów
- Kolokwium zaliczeniowe
- Kolokwium poprawkowe

Politechnika Wrocławska

Plan prezentacji

- Co to jest algorytm?
- Co to jest program?
- Zapis algorytmów
- Pierwszy program w języku C

Na podstawie:

- M. M. Sysło, *Algorytmy*
- G. Perry, D. Miller, *Język C Programowanie dla początkujących*
- D. Harel, *Rzecz o istocie informatyki. Algorytmika*

Politechnika Wrocławska

Co to jest algorytm?

- Algorytm jest przepisem opisującym krok po kroku rozwiązanie problemu lub osiągnięcie jakiegoś celu
- Algorytmika to dziedzina zajmująca się algorytmami i ich właściwościami

M. M. Sysło, *Algorytmy*, WSiP, 2002

Algorytm

- Słowo algorytm wywodzi się od nazwiska arabskiego matematyka i astronoma. Muhammad ibn Musa al-Chuwarizmi żył na przełomie VIII i IX wieku, jest uznawany za prekursora metod obliczeniowych w matematyce

https://pl.wikipedia.org/wiki/Muhammad_ibn_Musa_al-Chuwarizmi

M. M. Sysło, *Algorytmy*, WSiP, 2002

Politechnika Wrocławska

Przykład algorytmu

KAKAOWE CIASTO BEZ PIECZENIA

POPULARNE CIASTA | SZYBKIE CIASTO | DESERY BEZ PIECZENIA | CIASTA | CIASTA CZEKOLADOWE | CIASTA KOSTKI | CZEKOLADA

Rewelacyjne ciasto bez pieczenia! Kakaowe hebatniki przełożone kakaową masą budyniową i dżemem porzeczkowym (można połączyć pół na pół z powidłami śliwkowymi).

https://www.kwestiasmaku.com/przepis/kakaowe-ciasto-bez-pieczenia

Przykład algorytmu

SKŁADNIKI

KREM CZEKOLADOWY

- 1 litr mleka
- 3 łyżki mąki pszennej
- 3 łyżki mąki ziemniaczanej
- 1 szklanka cukru
- 3 łyżki kakao
- 2 żółtka

ORAZ

- ok. 600 g kakaowych herbatników "petit beurre"
- 200 g masła
- 1 słoiczek dżemu porzeczkowego lub powideł śliwkowych
- kakao

PRZYGOTOWANIE

DODAJ NOTATKĘ

KREM CZEKOLADOWY

- Odlać 1 i 1/2 szklanki mleka i dokładnie wymieszać je (np. rózgą) z mąką pszenną i ziemniaczaną, cukrem, żółtkami oraz likierami jeśli ich używamy.
- Resztę mleka zagotować (dokładnie, aż zacznie kipieć), następnie wlewać do niego mieszankę mleka, mąki i żółtek, jednocześnie energicznie mieszając rózgą. Zagotować co chwilę mieszając.
- Po zagotowaniu gotowy budyń odstawić z ognia, przelać do czystej miski i całkowicie ostudzić (na wierzch można położyć folię spożywczą aby nie zrobił się kożuch).
- Miękkie masło ubijać przez ok. 3 minuty aż się napuszy, następnie stopniowo, w krótkich odstępach czasu, dodawać budyń ciągle ubijając.

https://www.kwestiasmaku.com/przepis/kakaowe-ciasto-bez-pieczenia

Przykład algorytmu

PRZEŁOŻENIE

- Formę o wymiarach ok. **20 x 30 cm** (może być większa) wys
masłem i wyłożyć papierem do pieczenia. Układać warstwami na
herbatniki i krem budyniowy, otrzymując 4 lub 5 takich warstw,
druga warstwa od dołu ma mieć zamiast kremu - dżem. Na wie
ma być cienka warstwa kremu.

- Ciasto oprószyć kakao i wstawić do lodówki na kilka godzin lu
noc. Ciasto można przygotować dzień wcześniej.

WSKAZÓWKI

Opcjonalnie do kremu można dodać 4 łyżki likieru pomarańczo
amaretto lub orzechowego lub 2 łyżki mocnego alkoholu.

https://www.kwestiasmaku.com/przepis/kakaowe-ciasto-bez-pieczenia

Politechnika Wrocławska

Algorytm Euklidesa

- Algorytm wyznaczania największego wspólnego dzielnika dwóch dodatnich liczb całkowitych został sformułowany w IV w. p.n.e. przez Euklidesa

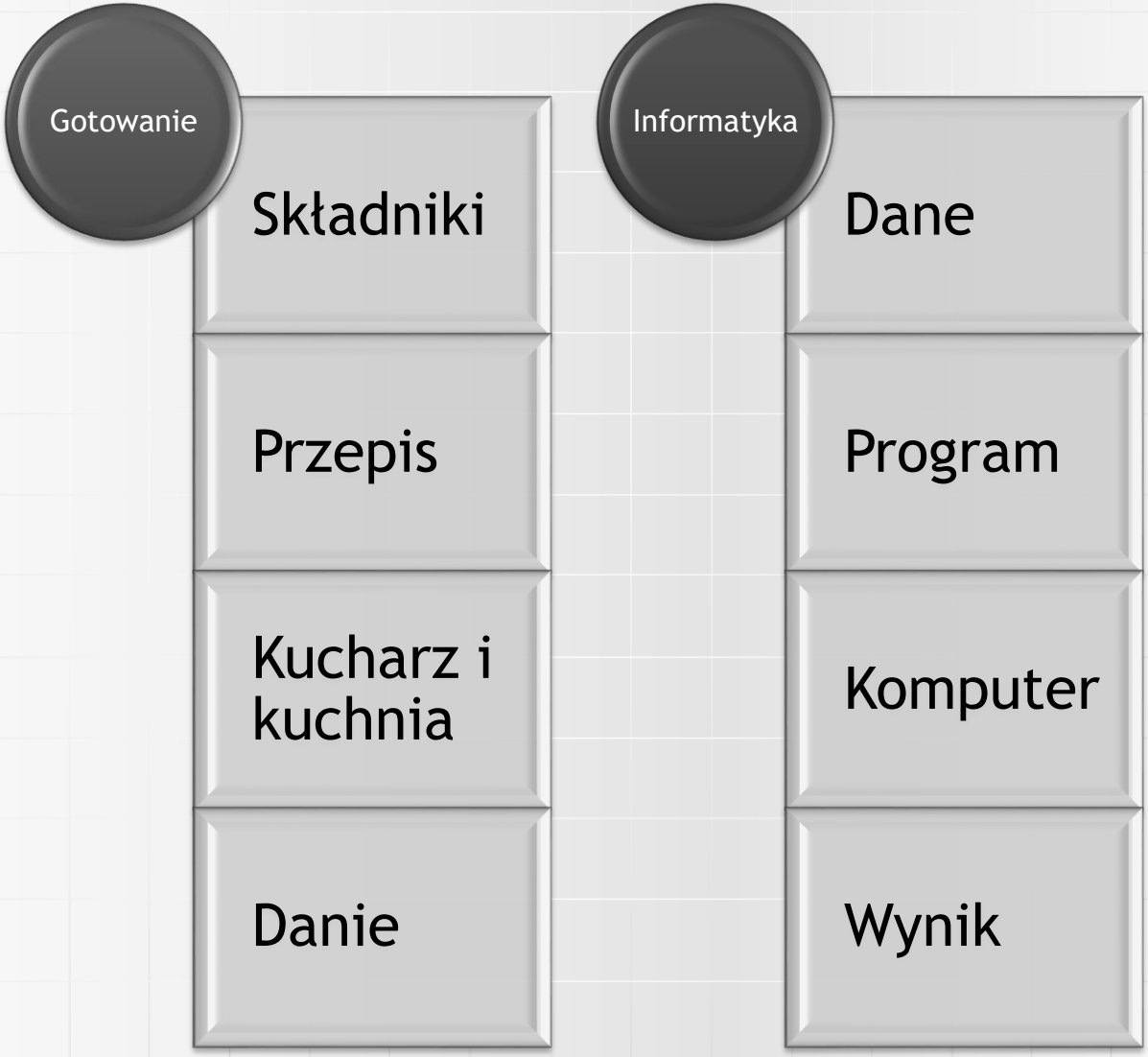
Politechnika Wrocławska

Co to jest program?

- Program to lista szczegółowych instrukcji przekazywanych komputerowi do realizacji określonych zadań

Politechnika Wrocławska

Program i przepis a algorytm



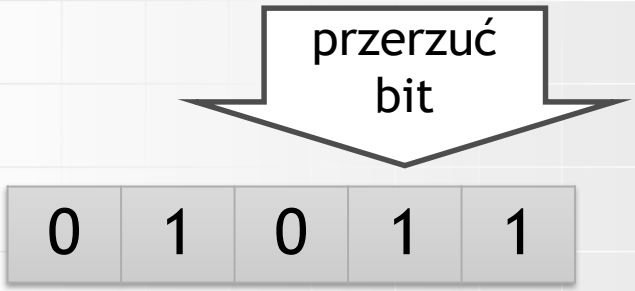
Politechnika Wrocławska

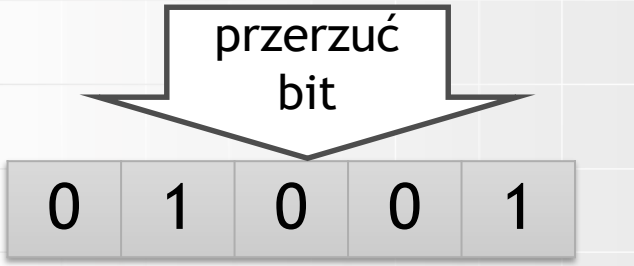
Komputer

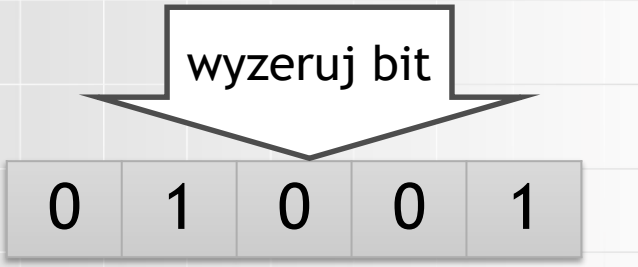
- O komputerach można myśleć, jako o zestawie przełączników – bitów.
- Każdy bit może znajdować się w jednym z dwóch stanów oznaczających 0 lub 1.
- Komputer może wykonać bezpośrednio niewielką liczbę prostych operacji (przerzucenie, wyzerowanie, sprawdzenie).

Politechnika Wrocławska

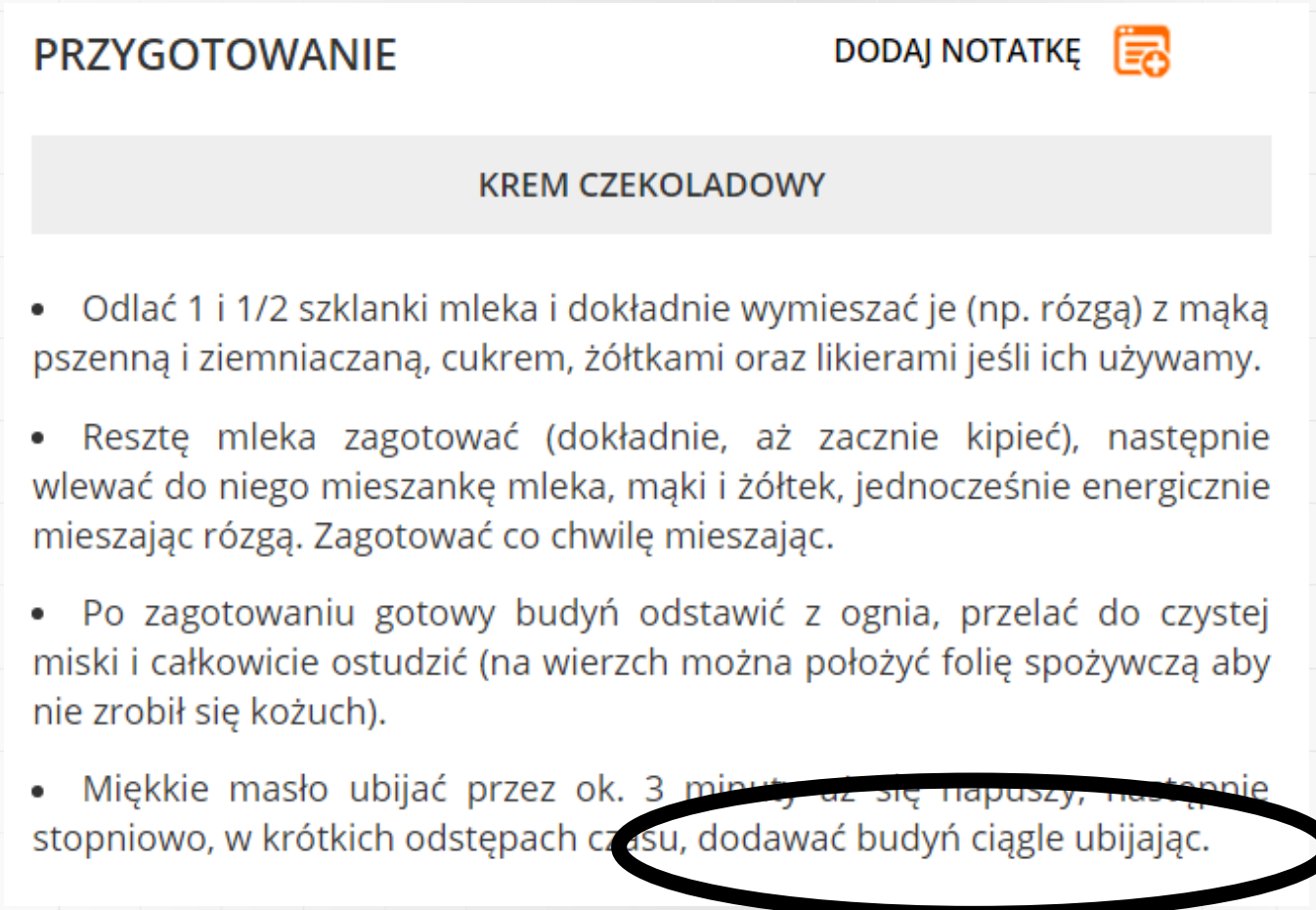
Komputer







Poziomy szczegółowości



PRZYGOTOWANIE

DODAJ NOTATKĘ

KREM CZEKOLADOWY

- Odlać 1 i 1/2 szklanki mleka i dokładnie wymieszać je (np. rózgą) z mąką pszenną i ziemniaczaną, cukrem, żółtkami oraz likierami jeśli ich używamy.
- Resztę mleka zagotować (dokładnie, aż zacznie kipieć), następnie wlewać do niego mieszankę mleka, mąki i żółtek, jednocześnie energicznie mieszając rózgą. Zagotować co chwilę mieszając.
- Po zagotowaniu gotowy budyń odstawić z ognia, przelać do czystej miski i całkowicie ostudzić (na wierzch można położyć folię spożywczą aby nie zrobił się kożuch).
- Miękkie masło ubijać przez ok. 3 minuty aż się napuszy, następnie stopniowo, w krótkich odstępach czasu, dodawać budyń ciągle ubijając.

- Przykład: mnożenie 528 × 46

Politechnika Wrocławska

Reprezentacje problemów

- Oblicz wartość funkcji $f(x) = \frac{x}{|x|}$

- Konieczne jest sformułowanie specyfikacji: warunki jakie powinny spełniać dane wejściowe, oraz jaki jest związek wyników z danymi.

- Dane: dowolna liczba rzeczywista x.
- Wynik: wartość funkcji f(x), jeśli x jest różne od zera i zero w przeciwnym przypadku

Reprezentacje problemów

Opis słowny

- Dane: dowolna liczba rzeczywista x.
- Wynik: wartość funkcji f(x), określonej wzorem

$$f(x) = \begin{cases} -1, & \text{dla } x < 0 \\ 0, & \text{dla } x = 0 \\ 1, & \text{dla } x > 0 \end{cases}$$

Reprezentacje problemów

Lista kroków

- Dane: dowolna liczba rzeczywista x.
- Wynik: wartość funkcji f(x), określonej wzorem

Krok 0. Wczytaj wartość danej x.

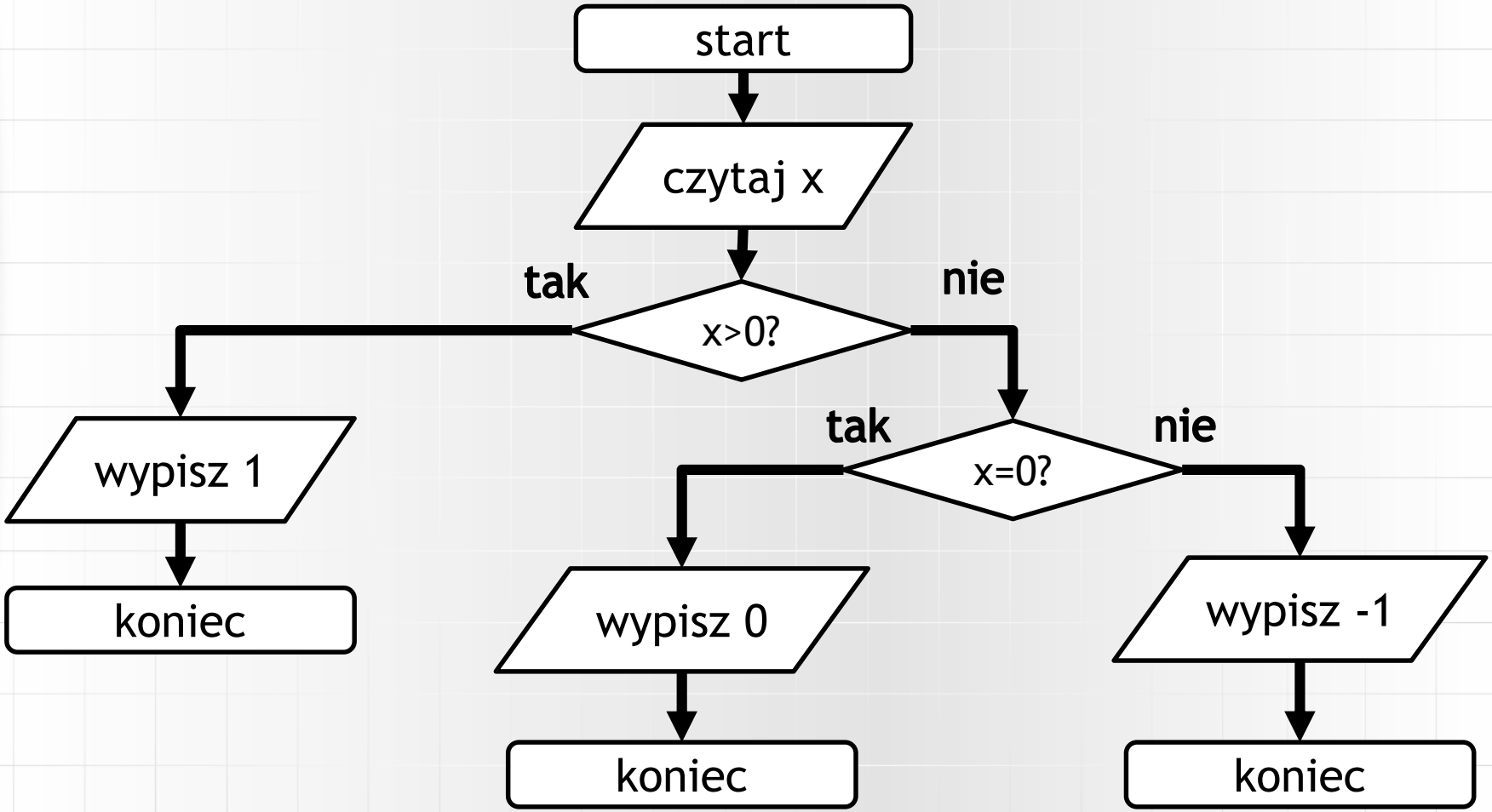
Krok 1. Jeśli x > 0, wynikiem jest 1. Zakończ algorytm.

Krok 2. {*W tym przypadku* $x \leq 0$} Jeśli x = 0, wynikiem jest 0. Zakończ algorytm.

Krok 3. {*W tym przypadku* $x < 0$} Wynikiem jest -1. Zakończ algorytm.

Reprezentacje problemów

Schemat blokowy



Reprezentacje problemów

Program w języku C

Start here | reprezentacja_algorytmu.c

```c
//reprezentacja_algorytmu.c

#include <stdio.h>

int main()
{
    float x;
    printf("Podaj x:"); scanf("%f",&x);

    if(x>0)
        printf("f(x) = 1");
    else //w tym przypadku x <= 0
        if(x==0)
            printf("f(x) = 0");
        else //w tym przypadku x < 0
            printf("f(x) = -1");
    return 0;
}

```

Reprezentacje problemów

Różnorodność języków programowania

- http://www.99-bottles-of-beer.net

Language C

(standard version)

Date: 04/20/05
Author: Bill Wein
URL: n/a
Comments: 4
Info: n/a
Score: ★★★ (2.81 in 124 votes)

```
/*
 * 99 bottles of beer in ansi c
 *
 * by Bill Wein: bearheart@bearnet.com
 *
 */

#define MAXBEER (99)

void chug(int beers);

main()
{
register beers;

for(beers = MAXBEER; beers; chug(beers--))
  puts("");

puts("\nTime to buy more beer!\n");

exit(0);
}

void chug(register beers)
{
char howmany[8], *s;

s = beers != 1 ? "s" : "";
printf("%d bottle%s of beer on the wall,\n", beers, s);
printf("%d bottle%s of beeeeer . . . ,\n", beers, s);
printf("Take one down, pass it around,\n");

if(--beers) sprintf(howmany, "%d", beers); else strcpy(howmany, "No more");
s = beers != 1 ? "s" : "";
printf("%s bottle%s of beer on the wall.\n", howmany, s);
}
```

Language Matlab

(vectorized version)

Date: 04/20/05
Author: Rich Stein
URL: n/a

Language Perl

(bottled by Acme::EyeDrops)

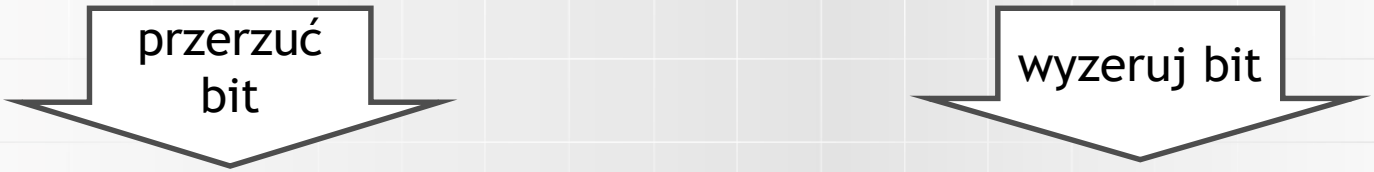
Date: 06/04/05
Author: Andrew Savige
URL: n/a
Comments: 76
Info: http://search.cpan.org/dist/Acme-EyeDrops/lib/Acme/EyeDrops.pm
Score: ★★★½ (3.51 in 2731 votes)

```
   ''=~(       '(?{'       .('`'       |'%')       .('['       ^'-')
   .(``'       |'!')       .('`'       |',')       .'"'.       '\\$'
   .'=='       .('['       ^'+')       .(``'       |'/')       .('['
   ^'+')       .'||'       .(';'       &'=')       .(';'       &'=')
   .';-'       .'-'.       '\\$'       .'=;'       .('['       ^'()')
   .('['       ^'.')       .('`'       |'"')       .('!'       ^'+')
  .'_\\{'     .'(\\$'      .';=('.     '\\$=|'     .'\|".(     '`'^'.'
 ).((``')|   '/').').'    .'\\"'.+(   '{'^'[').   (``'|'"')   .(``'|'/'
).('['^'/') .('['^'/').  (``'|',').(  ``'|('%')).  '\\".\\"'.(  '['^('(')).
'\\"'.('['^  '#').'!!--'  .'\\$=.\\"'  .('{'^'[').  (``'|'/').(  ``'|'\&").(
'{'^"\[").(  ``'|"\"").(  ``'|"\%").(  ``'|"\%").(  '[^(')')).  '\\").\\"'.
('{'^'[').(  ``'|"\/").(  ``'|"\.").(  '{'^"\[").(  '['^"\/").(  ``'|"\(").(
``'|"\%").(  '{'^"\[").(  '['^"\,").(  ``'|"\!").(  ``'|"\,").(  ``'|(',')).
'\\"\\}'.+(  '['^"\+").(  '['^"\)").(  ``'|"\)").(  ``'|"\.").(  '['^('/')).
'+_,\\",'.(  '{'^('[')).  ('\\$;!').(  '!'^"\+").(  '{'^"\/").(  ``'|"\!").(
``'|"\+").(  ``'|"\%").(  '{'^"\[").(  ``'|"\/").(  ``'|"\.").(  ``'|"\%").(
'{'^"\[").(  ``'|"\$").(  ``'|"\/").(  '['^"\,").(  ``'|('.')).  ','.(('{')^
'[').("\["^  '+').("\`"|  '!').("\["^  '(').("\["^  '(').("\{"^  '[').("\`"|
')').("\["^  '/').("\{"^  '[').("\`"|  '!').("\["^  ')').("\`"|  '/').("\["^
'.').("\`"|  '.').("\`"|  '$').("\,".(  '!'^('+')).  '\\",_,\\"'  .'!'.("\!"^
'+').("\!"^  '+').'\\"'.  ('['^',').(  ``'|"\(").(  ``'|"\)").(  ``'|"\,").(
``'|('%')).  '++\\$="})'  );$:=('.')^  '~';$~='@'|  '(';$^=')'^  '[';$/=``';
```

Reprezentacje problemów

Języki programowania a język maszynowy

- Język maszynowy zestaw podstawowych instrukcji, jakie dany procesor potrafi wykonać



- Program zapisany w języku programowania jest przetwarzany na język maszynowy

Politechnika Wrocławska

Co jest potrzebne do pisania programów w języku C?

Compiler	Author	Microsoft Windows	Unix-like	Other OSs	License type
AMPC	Axiomatic Solutions Sdn Bhd	No	Yes	Yes	Proprietary
Dev C++	Dev Inc.	Yes	No	No	Proprietary
Aztec C	Manx Software Systems	No	No	CP/M, CP/M-86, DOS, Classic Mac OS	Proprietary
Amsterdam Compiler Kit	Andrew Tanenbaum and Ceriel Jacobs	No	Yes	Yes	BSD
CCS C Compiler	CCS, Inc.	Yes	Yes	Yes	Proprietary
Ch	SoftIntegration, Inc	Yes	OS X, FreeBSD, Linux, Solaris, HP-UX, AIX, Qnx	Yes	Freeware
Clang	LLVM Project	Yes	Yes	Yes	BSD
CodeWarrior	Metrowerks	Yes	Yes	Classic Mac OS	Proprietary
CParser/libFirm	Matthias Braun, Christoph Mallon and Michael Beck	Yes	Yes	Yes	GPL
DeSmet-C	C-Ware Corporation	No	No	DOS	GPL
Digital Mars	Digital Mars	Yes	No	No	Proprietary
Dignus Systems/C	Dignus, L.L.C	Yes (host)	Yes (host)	Z/Architecture	Proprietary
Edison Design Group	Edison Design Group	Yes	Yes	Yes	Proprietary
GCC C	GNU Project	MinGW, Cygwin	Yes	IBM mainframe, AmigaOS, VAX/VMS, RTEMS	GPL
Hippo-C	Hippopotamus Software, Haba Systems	No	No	Classic Mac OS, Atari ST	Proprietary
IAR C/C++ Compilers	IAR Systems	Yes	No	No	Proprietary
Interactive C	KISS Institute for Practical Robotics	Yes	Unix, OS X, Linux, IRIX, Solaris, SunOS	No	Freeware
LabWindows/CVI	National Instruments	Yes	Yes	Yes	Proprietary
Lattice C	Lifeboat Associates	No	Yes	DOS, OS/2, Commodore, Amiga, Atari ST, Sinclair QL	Proprietary
lcc	Chris Fraser and David Hanson	Yes	Yes	Yes	Freeware (source code available for non-commercial use)
Mac C	Consulair	No	No	Classic Mac OS	Proprietary
Mark Williams C	Mark Williams Company	Yes	Coherent	Yes	Proprietary
Micro-C Compiler (mcc)	Dunfield Development Services	No	No	DOS	Freeware (source code available)
Micro C Compiler (mcc)	Roshan Singh	Yes	Yes	Yes	Freeware (source code available for non-commercial use)
MikroC Compiler	Mikroelektronika	Yes	Yes	Yes	Proprietary
MPW C	Apple	No	No	Classic Mac OS	Proprietary
Neatcc	Ali Gholami Rudi	No	Yes	No	BSD
Nwcc	Nils Weller	No	Yes	No	BSD
Open64	AMD SGI Google HP Intel Nvidia PathScale Tsinghua University and others	No	Yes	Yes	GPL
Open Watcom	Sybase and SciTech Software	Yes	Linux	OS/2, DOS	Sybase Open Watcom Public License
Orange C compiler	David Lindauer	Yes	No	DOS	BSD

Co jest potrzebne do pisania programów w języku C?

- Code::Blocks (www.codeblocks.org)

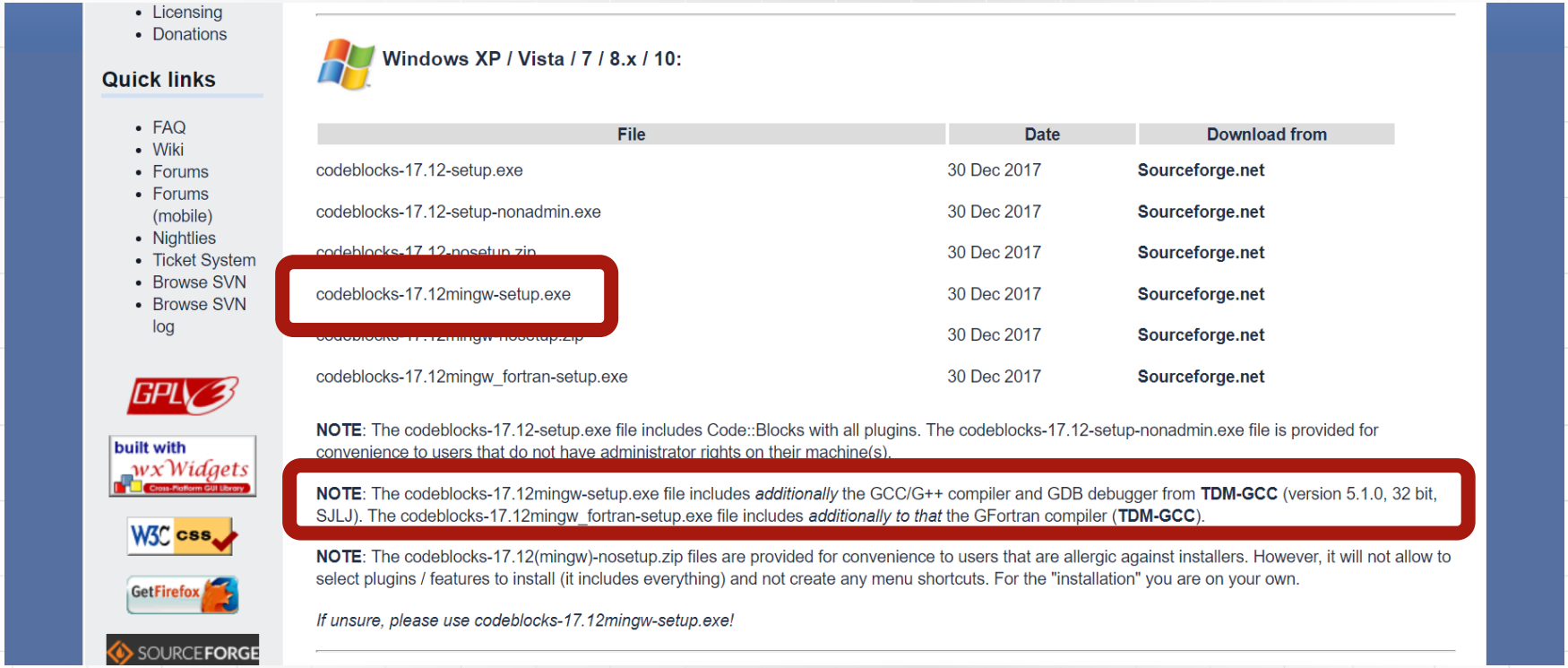
- Wieloplatformowe darmowe środowisko programistyczne o otwartym kodzie źródłowym dla programistów C, C++, Fortrana

Co jest potrzebne do pisania programów w języku C?

Politechnika Wrocławska

Co jest potrzebne do pisania programów w języku C?

Co jest potrzebne do pisania programów w języku C?



- Licensing
- Donations

Quick links

- FAQ
- Wiki
- Forums
- Forums (mobile)
- Nightlies
- Ticket System
- Browse SVN
- Browse SVN log

Windows XP / Vista / 7 / 8.x / 10:

File	Date	Download from
codeblocks-17.12-setup.exe	30 Dec 2017	**Sourceforge.net**
codeblocks-17.12-setup-nonadmin.exe	30 Dec 2017	**Sourceforge.net**
codeblocks-17.12-nosetup.zip	30 Dec 2017	**Sourceforge.net**
codeblocks-17.12mingw-setup.exe	30 Dec 2017	**Sourceforge.net**
[illegible]	30 Dec 2017	**Sourceforge.net**
codeblocks-17.12mingw_fortran-setup.exe	30 Dec 2017	**Sourceforge.net**

NOTE: The codeblocks-17.12-setup.exe file includes Code::Blocks with all plugins. The codeblocks-17.12-setup-nonadmin.exe file is provided for convenience to users that do not have administrator rights on their machine(s).

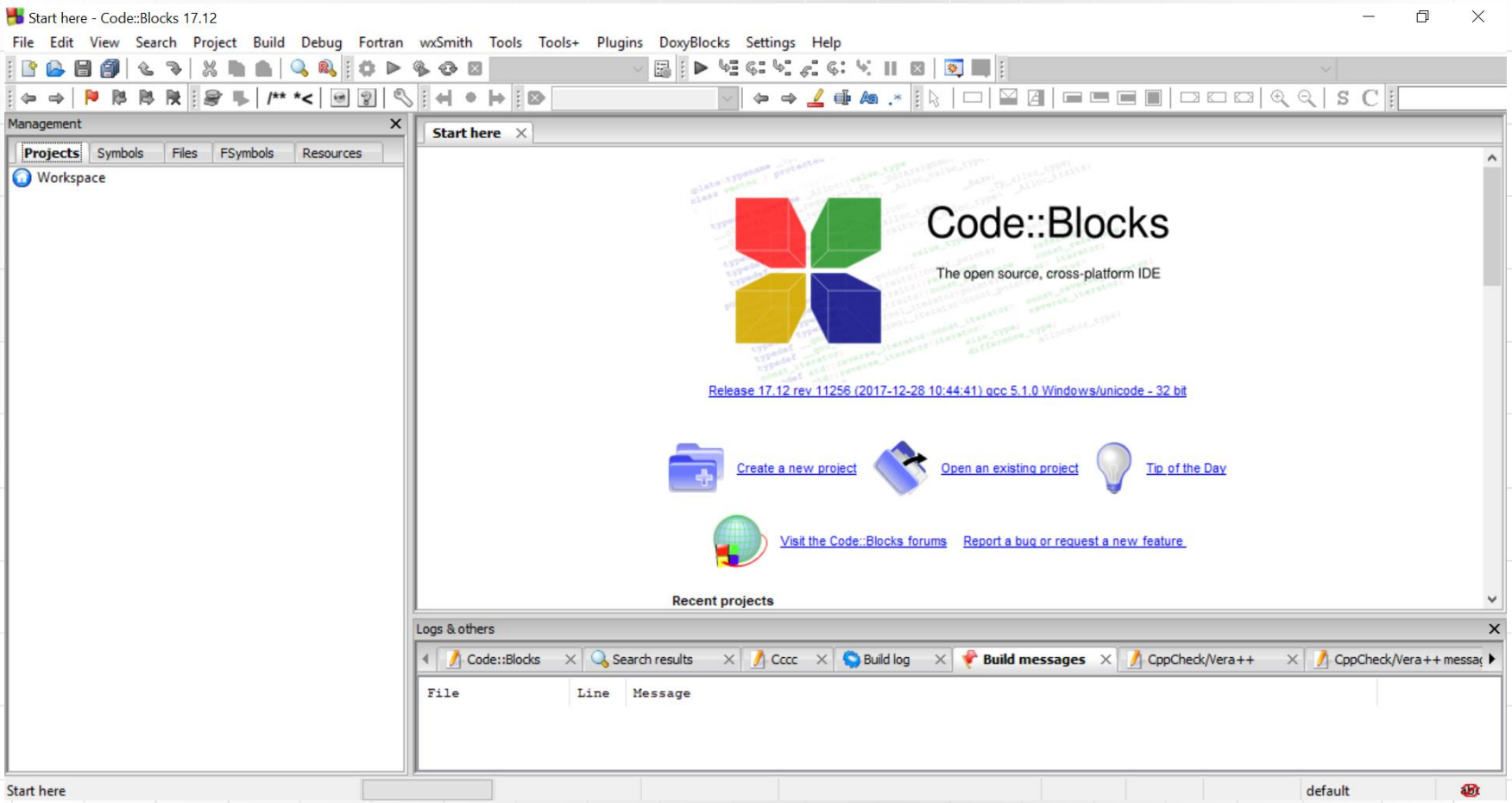
NOTE: The codeblocks-17.12mingw-setup.exe file includes *additionally* the GCC/G++ compiler and GDB debugger from **TDM-GCC** (version 5.1.0, 32 bit, SJLJ). The codeblocks-17.12mingw_fortran-setup.exe file includes *additionally to that* the GFortran compiler (**TDM-GCC**).

NOTE: The codeblocks-17.12(mingw)-nosetup.zip files are provided for convenience to users that are allergic against installers. However, it will not allow to select plugins / features to install (it includes everything) and not create any menu shortcuts. For the "installation" you are on your own.

If unsure, please use codeblocks-17.12mingw-setup.exe!

Politechnika Wrocławska

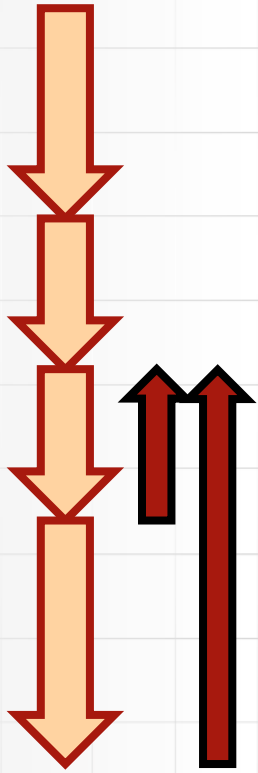
Co jest potrzebne do pisania programów w języku C?



Politechnika Wrocławska

Proces programowania

1. Dokładne określenie planowanej funkcjonalności programu
2. Napisanie kodu programu
3. Skompilowanie programu
4. Sprawdzenie, czy program nie zawiera błędów, i ewentualnie ich naprawienie
5. Wykonanie programu i sprawdzenie, czy działa prawidłowo (zgodnie z oczekiwaniami), ewentualna poprawa



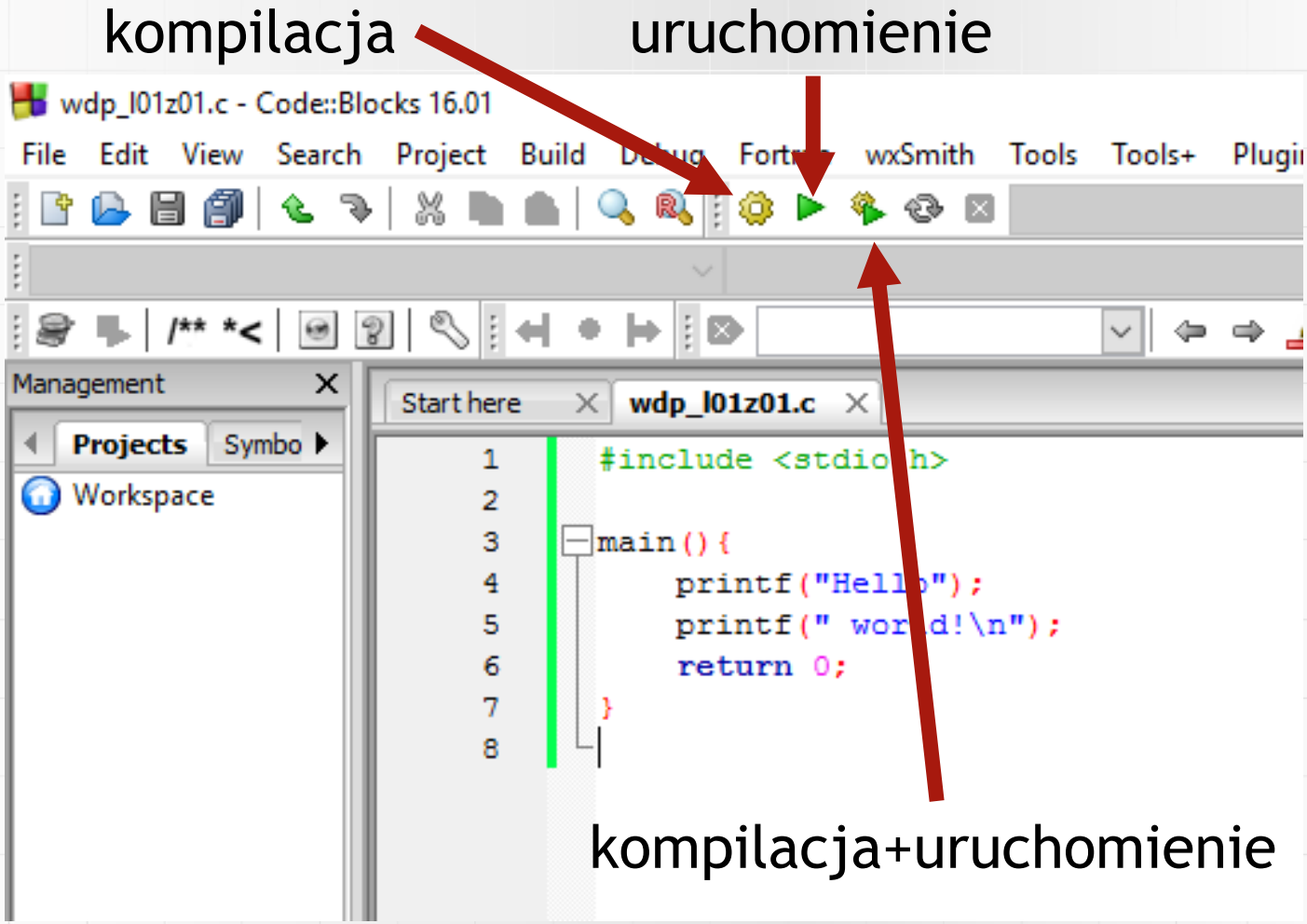
Politechnika Wrocławska

Przykład kodu

```
#include <stdio.h>

main(){
    printf("Hello");
    printf(" world!\n");
    return 0;
}
```

Przykład kodu



Politechnika Wrocławska

Przykład kodu - uwagi

- starannie wpisuj kod źródłowy
- dodawaj puste wiersze i wcinaj sekcje kodu
- polecenia i funkcje standardowe pisze się małymi literami

Politechnika Wrocławska

Funkcja `main()`

- przykłady funkcji:

  `main()` `calcIt()` `printf()` `strlen()`

- przykłady poleceń:

  `return` `while` `int` `if` `float`

Politechnika Wrocławska

Funkcja `main()`

- wykonywanie programu **zawsze** zaczyna się od funkcji **`main()`**
- program **musi** zawierać funkcję **`main()`**
- program może zawierać więcej funkcji
- bezpośrednio za napisem **`main()`** znajduje się otwierający nawias klamrowy (**{**) wyznaczający początek funkcji
- koniec funkcji wyznacza zamykający nawias klamrowy (**}**)

Politechnika Wrocławska

Funkcja `main()`

- prawie zawsze program będzie pobierał i drukował dane
- pomaga w tym dyrektywa `#include <stdio.h>`

Politechnika Wrocławska

Rodzaje danych

- znaki
- liczby całkowite
- liczby zmiennopozycyjne

Politechnika Wrocławska

Rodzaje danych - znaki

- każda litera, cyfra, spacja, symbol specjalny to jakiś znak

```
A a 7 % ! \ + ] (
```

- dane znakowe umieszcza się między apostrofami

```
'A' 'a' '7' '-'
```

Politechnika Wrocławska

Rodzaje danych - znaki

- w apostrofach można umieszczać pojedyncze znaki, a nie ich ciągi

- przykłady danych nie będących prawidłowymi znakami

```
'Wstęp do programowania'
'WPPT'
```

Politechnika Wrocławska

Rodzaje danych - łańcuchy

- ciąg znaków nazywamy łańcuchem

- łańcuchy umieszcza się w cudzysłowie

```
"Wstęp do programowania"
"WPPT"
"Hello"
" world!\n"
```

Politechnika Wrocławska

Rodzaje danych - sekwencje specjalne

- w języku C istnieje kilka dwuznakowych kombinacji interpretowanych jako pojedynczy znak, np. `'\n'`

Politechnika Wrocławska

Rodzaje danych - liczby

- liczby całkowite – nie mają części ułamkowej

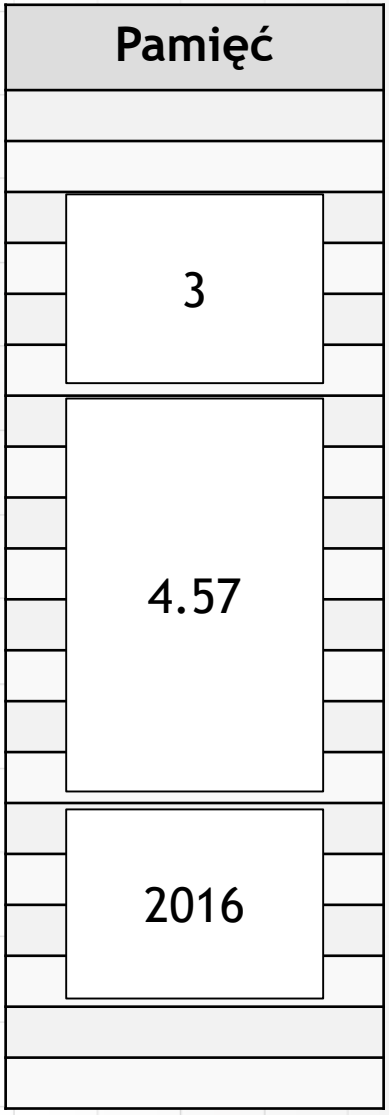
```
12  61  0  -51
```

- liczby zmiennopozycyjne

```
12.61  0.51  0.0  -10.48  -2.016
```

- wybór liczb zależy od rodzaju danych, na których pracuje program

Rodzaje danych - liczby



- liczby zmiennopozycyjne zajmują więcej miejsca, niż liczby całkowite
- zajęta pamięć nie zależy od wartości liczby

Politechnika Wrocławska

Przykład

Start here | wdp_l01p02.c

```
1   #include <stdio.h>
2
3   main(){
4       printf("To jest program w %c.\n",'C');
5       printf("Laboratorium %d z WdP.\n",1);
6       printf("WdP na %.1f ",99.9);
7       printf("procent.\n");
8       return 0;
9   }
10
```

funkcja **`main()`** to jedyna funkcja w tym programie napisana przez programistę

Przykład

Start here | wdp_l01p02.c

```
1   #include <stdio.h>
2
3   main(){
4       printf("To jest program w %c.\n",'C');
5       printf("Laboratorium %d z WdP.\n",1);
6       printf("WdP na %.1f ",99.9);
7       printf("procent.\n");
8       return 0;
9   }
10
```

treść (ciało) funkcji **`main()`**
objęta jest nawiasami klamrowymi

Przykład

Start here | wdp_l01p02.c

```
1   #include <stdio.h>
2
3   main(){
4       printf("To jest program w %c.\n",'C');
5       printf("Laboratorium %d z WdP.\n",1);
6       printf("WdP na %.1f ",99.9);
7       printf("procent.\n");
8       return 0;
9   }
10
```

w programie wywołano funkcję **`printf()`**

Politechnika Wrocławska

Podsumowanie

- Co to jest algorytm?
- Co to jest program?
- Zapis algorytmów
- Pierwszy program w języku C