

Informatyka 1 (EZ1E2008)

Politechnika Białostocka - Wydział Elektryczny
Elektrotechnika, semestr II, studia niestacjonarne I stopnia
Rok akademicki 2019/2020

Wykład nr 1 (06.03.2020)

dr inż. Jarosław Forenc

Dane podstawowe

- dr inż. Jarosław Forenc
- Politechnika Białostocka, Wydział Elektryczny,
Katedra Elektrotechniki, Energoelektroniki i Elektroenergetyki
ul. Wiejska 45D, 15-351 Białystok
WE-204
- e-mail: j.forenc@pb.edu.pl ■ tel. (0-85) 746-93-97
- <http://jforenc.prv.pl>
 - Dydaktyka - dodatkowe materiały do pracowni specjalistycznej
- konsultacje:
 - poniedziałek, godz. 09:00-10:00, WE-204
 - wtorek, godz. 11:00-12:00, WE-204
 - środa, godz. 09:00-10:00 + 12:15-13:00, WE-204
 - piątek, godz. 17:00-18:30, WE-204 (studia zaoczne)

Program wykładu (1/2)

1. Informacja analogowa i cyfrowa. Pozycyjne i niepozycyjne systemy liczbowe. Konwersje pomiędzy systemami liczbowymi.
2. Jednostki informacji cyfrowej. Kodowanie informacji. Kodowanie znaków.
3. Kodowanie liczb. Reprezentacja liczb w systemach komputerowych: stałoprzecinkowa i zmiennoprzecinkowa. Standard IEEE 754.
4. Programowanie w języku C. Deklaracje i typy zmiennych, operatory i wyrażenia arytmetyczne, operacje wejścia-wyjścia, operatory relacyjne i logiczne, wyrażenia logiczne, instrukcja warunkowa if, instrukcja switch, operator warunkowy, pętle (for, while, do .. while), tablice jednowymiarowe.
5. Architektura komputerów. Klasyfikacja systemów komputerowych (taksonomia Flynna). Architektura von Neumana i architektura harwardzka.

Program wykładu (2/2)

6. Budowa i zasada działania komputera. Procesor, pamięć wewnętrzna i zewnętrzna. Komunikacja z urządzeniami zewnętrznymi, interfejsy komputerowe. **Sprawdzian nr 1.**
7. System operacyjny. Funkcje i zadania systemu operacyjnego. Zarządzanie procesami, pamięcią i dyskami.
8. Sieci komputerowe. Technologie, protokoły, urządzenia. Zasada działania sieci Internet.
9. Algorytmy. Definicja algorytmu. Klasyfikacje i sposoby przedstawiania algorytmów. Rekurencja. Złożoność obliczeniowa. Sortowanie. Klasyfikacje algorytmów sortowania. Wybrane algorytmy sortowania.
10. **Sprawdzian nr 2.**

Literatura (1/3)

1. R. Kawa, J. Lembas: „Wykłady z informatyki. Wstęp do informatyki”. PWN, Warszawa, 2017.
2. W. Kwiatkowski: „Wprowadzenie do kodowania”. BEL Studio, Warszawa, 2010.
3. S. Gryś: „Arytmetyka komputerów w praktyce”. PWN, Warszawa, 2013.
4. W. Stallings: „Organizacja i architektura systemu komputerowego. Projektowanie systemu a jego wydajność”. WNT, Warszawa, 2004.
5. A.S. Tanenbaum: „Strukturalna organizacja systemów komputerowych”. Helion, Gliwice, 2006.
6. K. Wojtuszkiewicz: „Urządzenia techniki komputerowej. Część 1. Jak działa komputer? Część 2. Urządzenia peryferyjne i interfejsy”. PWN, Warszawa, 2011.

Literatura (2/3)

7. A.S. Tanenbaum, H. Bos: „Systemy operacyjne. Wydanie IV”. Helion, Gliwice, 2015.
8. W. Stallings: „Systemy operacyjne. Struktura i zasady budowy”. Mikom, Warszawa, 2006.
9. A.S. Tanenbaum, D.J. Wetherall: „Sieci komputerowe. Wydanie V”. Helion, Gliwice, 2012.
10. G. Coldwin: „Zrozumieć programowanie”. PWN, Warszawa, 2015.
11. S. Prata: „Język C. Szkoła programowania. Wydanie VI”. Helion, Gliwice, 2016.

Literatura (3/3)

12. P. Wróblewski: „Algorytmy, struktury danych i techniki programowania. Wydanie V”. Helion, Gliwice, 2015.
13. M. Sysło: Algorytmy. Helion, Gliwice, 2016.
14. B. Buczek: Algorytmy. Ćwiczenia. Helion, Gliwice, 2008.

Zaliczenie wykładu - efekty uczenia się

EU1

identyfikuje i opisuje zasadę działania podstawowych elementów systemu komputerowego

EU2

opisuje podstawowe zadania systemu operacyjnego oraz strukturę sieci komputerowych

EU3

formułuje algorytmy komputerowe rozwiązujące typowe zadania inżynierskie występujące w elektrotechnice

- Szczegółowe zasady zaliczania znajdują się na stronie internetowej (<http://jforenc.prv.pl/dydaktyka.html>) oraz w systemie USOS

Terminy zajęć i zaliczeń

- Wykład nr 1 - 06.03.2020
- Wykład nr 2 - 13.03.2020
- Wykład nr 3 - 20.03.2020
- Wykład nr 4 - 27.03.2020
- Wykład nr 5 - 03.04.2020
- Wykład nr 6 - 24.04.2020
- Wykład nr 7 - 08.05.2020 (Sprawdzian nr 1: EU1)
- Wykład nr 8 - 15.05.2020
- Wykład nr 9 - 05.06.2020
- Wykład nr 10 - 19.06.2020 (Sprawdzian nr 2: EU2, EU3)
- Zaliczenie poprawkowe - sesja egzaminacyjna (EU1, EU2, EU3)

Zaliczenie wykładu

- Za każdy efekt uczenia się można otrzymać od 0 do 100 pkt.
- Na podstawie otrzymanych punktów wystawiana jest ocena:

Punkty	Ocena	Punkty	Ocena
91 - 100	5,0	61 - 70	3,5
81 - 90	4,5	51 - 60	3,0
71 - 80	4,0	0 - 50	2,0

- Prowadzący zajęcia może przyznawać dodatkowe punkty za aktywność na wykładzie
- Ocena końcowa wyznaczana jest na podstawie sumy punktów:

Punkty	Ocena	Punkty	Ocena
273 - 300	5,0	183 - 212	3,5
243 - 272	4,5	153 - 182	3,0
213 - 242	4,0	0 - 152	2,0

Plan wykładu nr 1

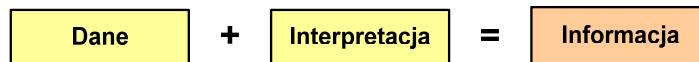
- Podstawowe pojęcia: informatyka i informacja
- Informacja analogowa i cyfrowa
- Systemy liczbowe
 - liczby i cyfry
 - systemy pozycyjne (dziesiętny, dwójkowy, szesnastkowy)
 - systemy niepozycyjne (rzymski)
- Język C
 - historia, struktura programu
 - kompilacja, zapis kodu
 - sekwencje sterujące, komentarze

Informatyka

- **Informatyka** (ang. computer science)
 - dziedzina nauki i techniki zajmująca się gromadzeniem, przetwarzaniem i wykorzystywaniem **informacji**
 - w języku polskim termin informatyka zaproponował w październiku 1968 r. prof. Romuald Marczyński na konferencji poświęconej „maszynom matematycznym”
 - wzorem nazwy były francuskie **informatique** i niemieckie **Informatik**
- **Informatykę** można rozpatrywać jako:
 - samodzielną dyscyplinę naukową
 - narzędzie wykorzystywane przez inne nauki
 - gałąź techniki
 - przemysł wytwarzający sprzęt (hardware) i oprogramowanie (software)

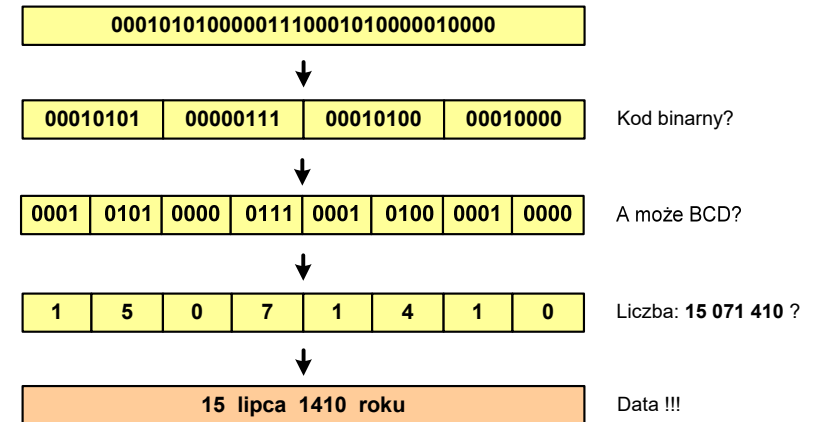
Informacja

- **Informatyka** (ang. computer science)
 - dziedzina nauki i techniki zajmująca się gromadzeniem, przetwarzaniem i wykorzystywaniem **informacji**
- **Informacja** - wielkość abstrakcyjna, która może być:
 - przechowywana w pewnych obiektach
 - przesyłana pomiędzy pewnymi obiektami
 - przetwarzana w pewnych obiektach
 - stosowana do sterowania pewnymi obiektami
- **Dane** - surowe fakty i liczby
- **Przetwarzanie danych** - logicznie powiązany zespół czynności pozwalających na uzyskanie z danych niezbędnych informacji



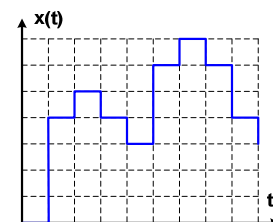
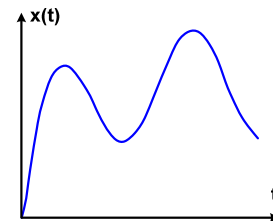
Informacja

- Co oznaczają poniższe dane?



Informacja analogowa i cyfrowa

- **Sygnal analogowy**
 - może przyjmować dowolną wartość z ciągłego przedziału (nieskończonego lub ograniczonego zakresem zmienności)
 - wartości mogą zostać określone w każdej chwili czasu dzięki funkcji matematycznej opisującej dany sygnał
- **Sygnal cyfrowy**
 - dziedzina i zbiór wartości są dyskretne
 - sygnał ciągły, który może zmieniać swoją wartość tylko w określonych chwilach czasu i może przyjmować tylko określone wartości



Informacja analogowa i cyfrowa

- **Zalety sygnałów cyfrowych:**
 - odporne na zakłócenia
 - powtarzalne (np. kopia filmu na DVD i VHS)
 - możliwość przesyłania na duże odległości
 - możliwość szyfrowania sygnału (kryptografia)
 - niższe koszty przetwarzania
- **Wady sygnałów cyfrowych:**
 - ograniczenie częstotliwości próbkowania (sygnał analogowy zamieniony na cyfrowy i ponownie na analogowy nie jest już tym samym sygnałem)

Liczby i cyfry

- **Liczba** - pojęcie abstrakcyjne, abstrakcyjny wynik obliczeń, wartość
 - umożliwia wyrażenie wyniku liczenia przedmiotów oraz mierzenia wielkości
- **Cyfra** - umowny znak (symbol) stosowany do zapisu liczby
 - liczba znaków służących do zapisu jest zależna od **systemu liczbowego** i przyjętego sposobu zapisu
 - system dziesiętny - 10 znaków
 - system szesnastkowy - 16 znaków
 - system rzymski - 7 znaków
- **Cyfry rzymskie**

I	V	X	L	C	D	M
1	5	10	50	100	500	1000

Liczby i cyfry

- **Cyfry arabskie** (pochodzą z Indii)
 - arabskie, standardowe europejskie
- indyjsko-arabskie
- wschodnio-indyjsko-arabskie
- W niektórych systemach jako cyfry stosowane są litery, np.

1	2	3	4	5	6	7	8	9	0
---	---	---	---	---	---	---	---	---	---

१	२	३	४	५	६	७	८	९	०
1	2	3	4	5	6	7	8	9	0

१	२	३	४	५	६	७	८	९	०
1	2	3	4	5	6	7	8	9	0

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Liczby i cyfry

- Inne przykłady zapisu cyfr i liczb:

cyfry etruskie

I·Λ·X·XX·ΛXX·↑·*(C)·⊕(⊕)
1 5 10 20 25 50 100 1000

cyfry grecko-jońskie

α	β	γ	δ	ε	ς	ζ	η	θ
1	2	3	4	5	6	7	8	9
ι	κ	λ	μ	ν	ξ	ο	π	ρ
10	20	30	40	50	60	70	80	90
ρ	σ	τ	υ	φ	χ	ψ	ω	Ͱ
100	200	300	400	500	600	700	800	900
Ϡ	β	Ϟ	Ϝ	Ϛ	ϛ	ϙ	Ϙ	ϕ
1000	2000	3000	4000	5000	6000	7000	8000	9000

Ϡ Ϟ Ϟ Ϛ ϛ

cyfry w pisowni chińskiej

jeden	一	sześć	六
dwa	二	siedem	七
trzy	三	osiem	八
cztery	四	dziewięć	九
pięć	五	dziesięć	十
zero	另		

ucziechinskiego.com

Liczby i cyfry

- Inne przykłady zapisu cyfr i liczb:

liczby w piśmie klinowym (Babilończycy)

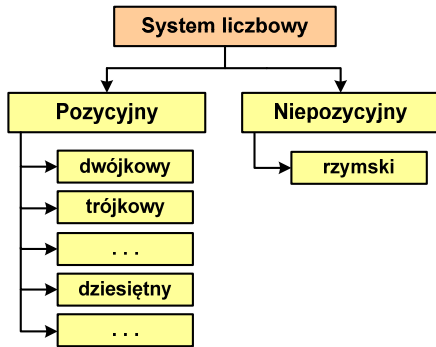
1	2	3	4	5	6	7	8	9
10	20	30	40	50	60	70	80	90
100	200	300	400	500	600	700	800	900
1000	2000	3000	4000	5000	6000	7000	8000	9000

system prekolumbijski

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19

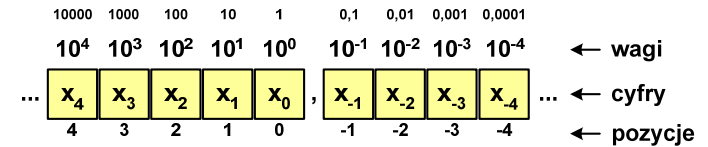
Systemy liczbowe

- System liczbowy - zbiór zasad umożliwiających zapis liczb za pomocą cyfr oraz wykonywanie działań na tych liczbach

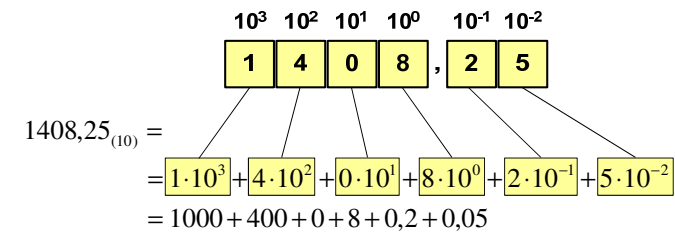


- Pozycyjny** - znaczenie cyfry jest zależne od miejsca (pozycji), które zajmuje ona w liczbie
 - system dziesiętny - liczba 111 (każda cyfra ma inne znaczenie)
- Niepozycyjny** - znaczenie cyfry jest niezależne od miejsca położenia w liczbie
 - system rzymski - liczba III

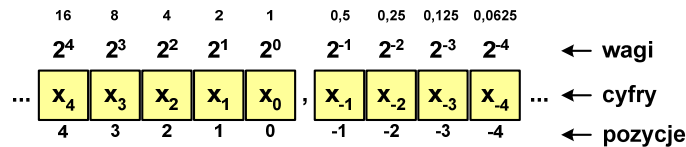
System dziesiętny (ang. decimal)



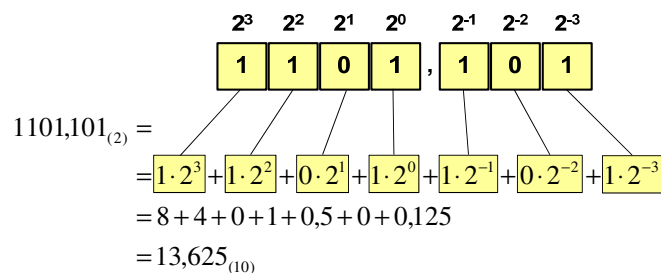
- p - podstawa systemu pozycyjnego, D - zbiór dozwolonych cyfr
- $p = 10$, $D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$



System dwójkowy (ang. binary)

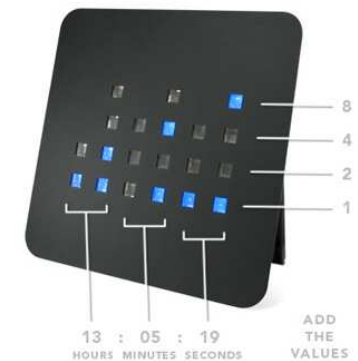


- w systemie dwójkowym: $p = 2$, $D = \{0, 1\}$



System dwójkowy - zastosowania

- Powszechnie używany w informatyce, technice cyfrowej



System szesnastkowy (ang. hexadecimal)

- System heksadecymalny
- $p = 16$, $D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$
- Powszechnie używany w informatyce - jeden bajt można zapisać za pomocą tylko dwóch cyfr szesnastkowych

$$3A5D_{(16)} = 3 \cdot 16^3 + 10 \cdot 16^2 + 5 \cdot 16^1 + 13 \cdot 16^0 = 14941_{(10)}$$

- Sposoby zapisu liczb w systemie szesnastkowym:

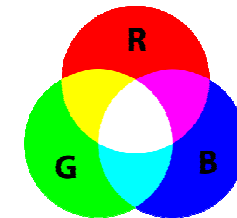
3A5Dh	0x3A5D	#3A5D
3A5D ₍₁₆₎	3A5D ₁₆	3A5D _{hex}
(3A5D) _{hex}	(3A5D) ₁₆	\$3A5D

System szesnastkowy - zastosowania

- Zapis 24-bitowego koloru RGB (Red-Green-Blue), 16 mln kolorów
- Każda barwa przyjmuje wartość z zakresu: $0..255_{(10)}$, $00..FF_{(16)}$



#FF48B8



System szesnastkowy - zastosowania

- Zapis 24-bitowego koloru RGB (Red-Green-Blue), 16 mln kolorów
- Kolory w dokumentach HTML:

```
<BODY bgcolor="#336699" text="#000000" link="#FFFF00"  
vlink="#33FFFF" alink="#FF0000">
```

ARCHIWUM	Studia stacjonarne:
ENGLISH	Poniedziałek: 12:15 - 14:00 Informatyka 1 - wykład , sem. 2 ED, WE-Aula II
	Wtorek: 08:30 - 10:00 Informatyka 1 - prac. , sem. 2 ED, gr. PS3, WE-110 12:15 - 13:45 Informatyka 1 - prac. , sem. 2 ED, gr. PS2, WE-110 14:15 - 15:45 Informatyka 1 - prac. , sem. 2 ED, gr. PS1, WE-110

System szesnastkowy - zastosowania

- 48-bitowy adres fizyczny urządzenia (MAC - Media Access Control)

88 : AD : D2 : 09 : 41 : 3B

producent numer egzemplarza

- <http://hwaddress.com>

HWaddress	Home	Countries	Companies	88:AD:D2	Search
OUI	MAC range	Company			
88-AD-D2	88-AD-D2-00-00-00 - 88-AD-D2-FF-FF-FF	Samsung Electronics Co.,Ltd			

Przykład systemu niepozycyjnego - system rzymski

- W systemie rzymskim posługujemy się siedmioma znakami:
 $I - 1$ $V - 5$ $X - 10$ $L - 50$ $C - 100$ $D - 500$ $M - 1000$
- Za pomocą dostępnych symboli można określić liczby od 1 do 3999
- System **addytywny** - wartość liczby określa się na podstawie sumy wartości cyfr, np.
 - II ($1 + 1 = 2$), XXX ($10 + 10 + 10 = 30$)
 - CLX ($100 + 50 + 10 = 160$), $MMXII$ ($1000 + 1000 + 10 + 1 + 1 = 2012$)
- Wyjątkiem od powyższej zasady są liczby do opisu których używa się odejmowania, np.
 - IV ($5 - 1 = 4$), IX ($10 - 1 = 9$), XL ($50 - 10 = 40$), XC ($100 - 10 = 90$)
- Stosowany w łacińskiej części Europy do końca Średniowiecza
- Niewygodny w prowadzeniu nawet prostych działań arytmetycznych, brak ułamków

Przykład systemu niepozycyjnego - system rzymski

- Zasady tworzenia liczb:
 - zestawiamy odpowiednie znaki od oznaczającego liczbę największą do oznaczającego liczbę najmniejszą
 $XVI = 10(X) + 5(V) + 1(I) = 16$
 - jeżeli składnik liczby, którą piszemy, jest wielokrotnością liczby nominalnej, wtedy zapisywany jest z użyciem kilku następujących po sobie znaków
 $CCC = 100(C) + 100(C) + 100(C) = 300$
 - dodatkowo należy zachować zasadę nie pisania czterech tych samych znaków po sobie, lecz napisać jeden znak wraz ze znakiem oznaczającym wartość większą o jeden rząd liczbowy
 $CD = 500(D) - 100(C) = 400$

Przykład systemu niepozycyjnego - system rzymski

- Zasady odczytu liczb:
 - cyfry jednakowe są dodawane
 $MMM = 1000(M) + 1000(M) + 1000(M) = 3000$
 - cyfry mniejsze stojące przed większymi są odejmowane od nich
 $CDXCIV = 500(D) - 100(C) + 100(C) - 10(X) + 5(V) - 1(I) = 494$
 - cyfry mniejsze stojące za większymi są do nich dodawane
 $MDCLX = 1000(M) + 500(D) + 100(C) + 50(L) + 10(X) = 1660$

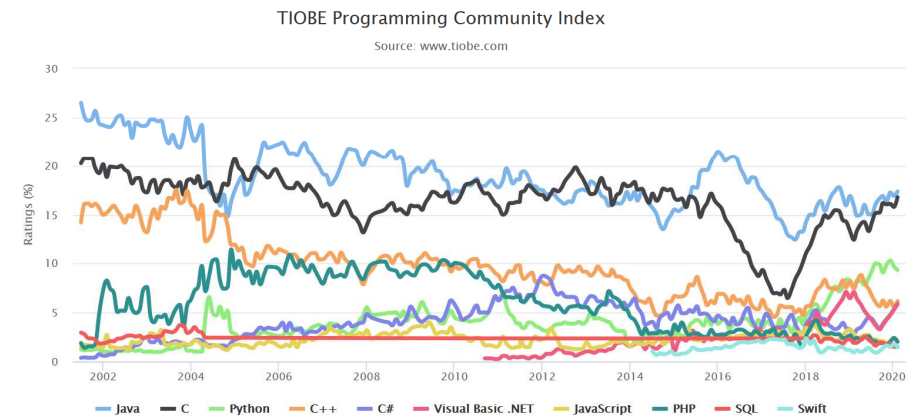
Język C - krótka historia (1/2)

- **1969** - język BCPL - Martin Richards, University Mathematical Laboratories, Cambridge
- **1970** - język B - Ken Thompson, adaptacja języka BCPL dla pierwszej instalacji systemu Unix na komputer DEC PDP-7
- **1972** - język NB (New B), nazwany później C - Dennis Ritchie, Bell Laboratories, New Jersey, system Unix na komputerze DEC PDP-11
 - 90% kodu systemu Unix oraz większość programów działających pod jego kontrolą napisane w C
- **1978** - książka „The C Programming Language” (Kernighan, Ritchie), pierwszy podręcznik, nieformalna definicja standardu (**K&R**)

Język C - krótka historia (2/2)

- 1989 - standard ANSI X3.159-1989 „Programming Language C” (ANSI C, C89)
- 1990 - adaptacja standardu ANSI C w postaci normy ISO/IEC 9899:1990 (C90)
- 1999 - norma ISO/IEC 9899:1999 (C99)
- 2011 - norma ISO/IEC 9899:2011 (C11)
- 2018 - norma ISO/IEC 9899:2018 (C18 lub C17)

Język C - TIOBE Programming Community Index



Język C - pierwszy program

- Niesformatowany plik tekstowy o odpowiedniej składni i mający rozszerzenie `.c`
- Kod najprostszego programu:

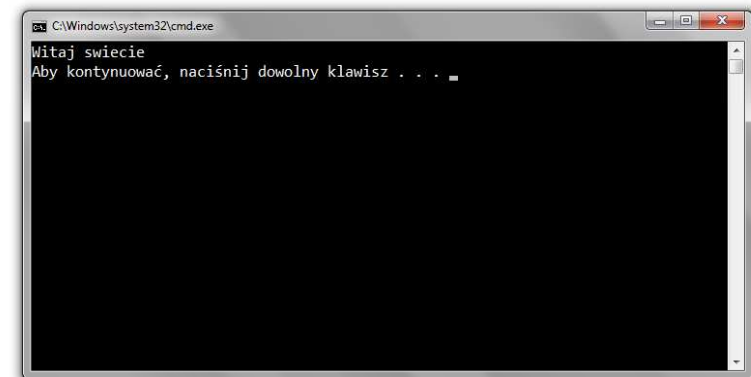
```
#include <stdio.h>

int main(void)
{
    printf("Witaj swiecie\n");
    return 0;
}
```

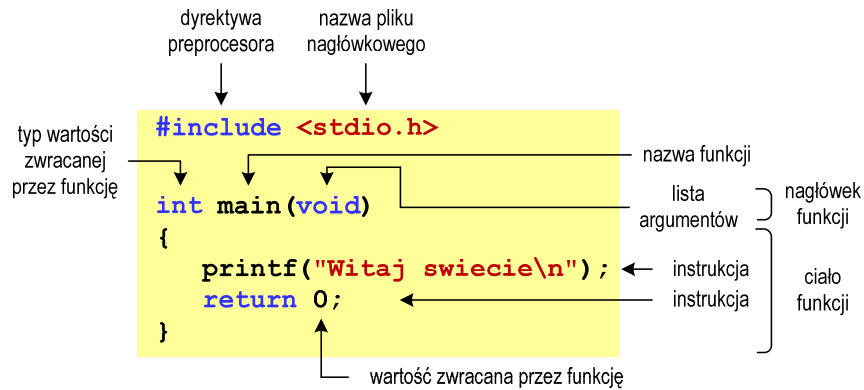
- Program konsolowy - wyświetla w konsoli tekst `Witaj swiecie`

Język C - pierwszy program

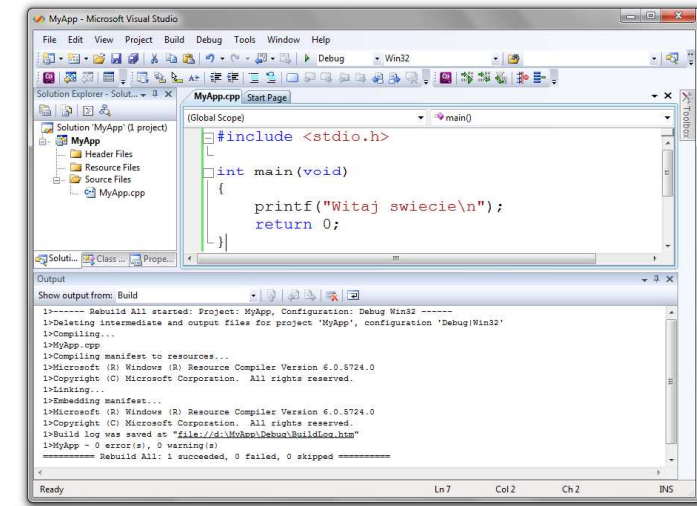
- Wynik uruchomienia programu:



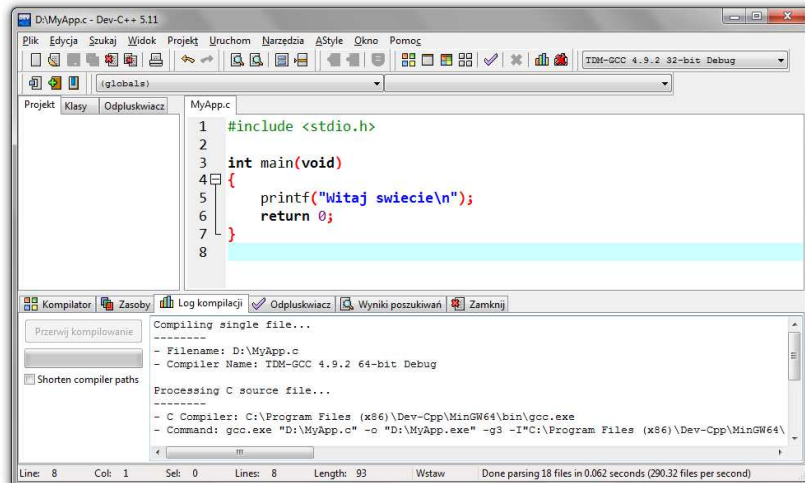
Język C - struktura programu



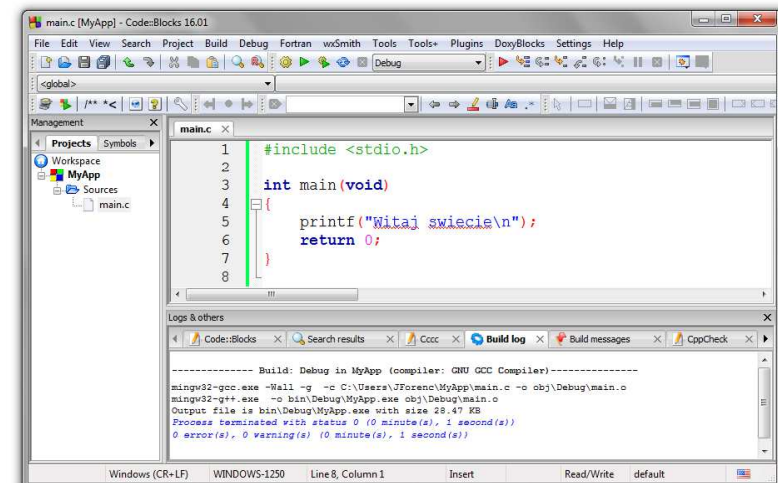
Microsoft Visual Studio 2008



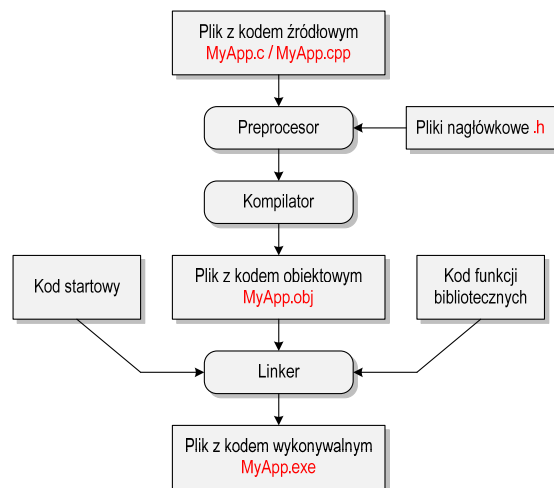
Dev-C++ 5.11



Code::Blocks 16.01



Język C - kompilacja programu



Język C - zapis kodu programu

- Sposób zapisu kodu programu wpływa tylko na jego przejrzystość, a nie na kompilację i wykonanie
- W takiej postaci program także skompiluje się:

```
#include <stdio.h>
int main(void) {printf("Witaj swiecie\n"); return 0; }
```

- W Microsoft Visual Studio 2008 można automatycznie sformatować kod źródłowy programu - **Ctrl + K + F**
- Język C rozróżnia wielkość liter - poniższy kod nie skompiluje się:

```
#include <stdio.h>
int Main(void) {printf("Witaj swiecie\n"); return 0; }
```

Język C - Wyświetlanie tekstu (printf)

- Znak przejścia do nowego wiersza `\n` może pojawić w dowolnym miejscu łańcucha znaków

```
printf("Witaj swiecie\n");
```

Witaj swiecie

```
printf("Witaj\nswiecie\n");
```

Witaj
swiecie

```
printf("Witaj ");
printf("swiecie");
printf("\n");
```

Witaj swiecie

Język C - Sekwencje sterujące

- Istnieją także inne sekwencje sterujące (ang. escape sequence)

Opis znaku	Zapis w printf()
Alarm (ang. alert), głośniczek wydaje dźwięk	<code>\a</code>
Backspace	<code>\b</code>
Wysunięcie strony (ang. form feed)	<code>\f</code>
Przejście do nowego wiersza (ang. new line)	<code>\n</code>
CR - Carriage Return (powrót na początek wiersza)	<code>\r</code>
Tabulacja pozioma (odstęp) (ang. horizontal tab)	<code>\t</code>
Tabulacja pionowa (ang. vertical tab)	<code>\v</code>

Język C - Wyświetlenie znaków specjalnych

- Niektóre znaki pełnią specjalną funkcję i nie można wyświetlić ich w tradycyjny sposób

Opis znaku	Znak	Zapis w printf()
Cudzysłów	"	\"
Apostrof	'	\'
Ukośnik (ang. backslash)	\	\\
Procent	%	%%

Sciezka dostępu: "C:\dane\plik.txt"

```
printf("Sciezka dostepu: \"C:\\dane\\plik.txt\"\\n");
```

Język C - Wyświetlenie znaku o podanym kodzie

- Można wyświetlić dowolny znak podając jego kod w systemie ósemkowym lub szesnastkowym

Znaczenie	Zapis
Znak o podanym kodzie ASCII (system ósemkowy)	\0oo
Znak o podanym kodzie ASCII (system szesnastkowy)	\xhh

```
printf("\127\151\164\141\152\040");  
printf("\x73\x77\x69\x65\x63\x69\x65\x21\x0A");
```

Witaj swiecie!

Język C - Wyświetlenie tekstu

```
#include <stdio.h>  
  
int main(void)  
{  
    printf("-----\n");  
    printf("| Punkty | Ocena |\n");  
    printf("-----\n");  
    printf("| 91-100 | 5,0 |\n");  
    printf("| 81-90 | 4,5 |\n");  
    printf("| 71-80 | 4,0 |\n");  
    printf("| 61-70 | 3,5 |\n");  
    printf("| 51-60 | 3,0 |\n");  
    printf("| 0-50 | 2,0 |\n");  
    printf("-----\n");  
  
    return 0;  
}
```

Punkty	Ocena
91-100	5,0
81-90	4,5
71-80	4,0
61-70	3,5
51-60	3,0
0-50	2,0

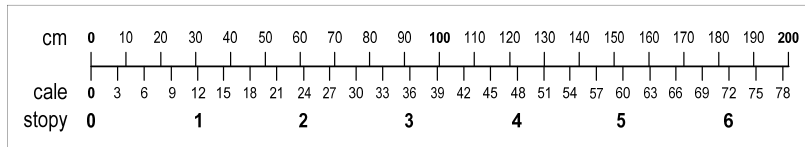
Język C - Komentarze

- Komentarze są pomijane podczas kompilacji

```
/*  
Nazwa: MyApp.cpp  
Autor: Jarosław Forenc, Politechnika Białostocka  
Data: 06-03-2020 18:30  
Opis: Program wyświetlający tekst "Witaj swiecie"  
*/  
  
#include <stdio.h> // zawiera deklarację printf()  
  
int main(void) // nagłówek funkcji main()  
{  
    printf/* funkcja */("Witaj swiecie\n");  
  
    return 0;  
}
```

Przykład: zamiana wzrostu w cm na stopy i cale

- Wybrane jednostki długości w brytyjskim systemie miar:
 - 1 cal (inch) [in] = 2,54 [cm]
 - 1 stopa (foot) [ft] = 12 cali = 30,48 [cm]



- 1 jard (yard) [yd] = 3 stopy = 91,44 [cm]
- 1 furlong [fur] = 660 stóp = 201,168 [m]
- 1 mila (mile) [mi] = 8 furlongów = 1609,344 [m]

Koniec wykładu nr 1

Dziękuję za uwagę!
(następny wykład: 13.03.2020)

Przykład: zamiana wzrostu w cm na stopy i cale

```
#include <stdio.h>

int main(void)
{
    float cm;      /* wzrost w cm */
    float stopy;   /* wzrost w stopach */
    float cale;    /* wzrost w calach */

    printf("Podaj wzrost w cm: ");
    scanf("%f", &cm);

    stopy = cm / 30.48f;
    cale = cm / 2.54f;

    printf("%f [cm] = %f [ft]\n", cm, stopy);
    printf("%f [cm] = %f [in]\n", cm, cale);

    return 0;
}
```

```
Podaj wzrost w cm: 175
175.000000 [cm] = 5.741470 [ft]
175.000000 [cm] = 68.897636 [in]
```