



Podstawy programowania (wykład)

Wprowadzenie Elementarz języka C – część 1

Tematyka wykładu

1. Wprowadzenie. Elementarz języka C - część 1
2. Elementarz języka C - część 2
3. Instrukcje
4. Operatory
5. Tablice
6. Wskaźniki
7. Łańcuchy znakowe
8. Struktury i unie. Pola bitowe
9. Funkcje - część 1
10. Funkcje - część 2 (m.in. makrodefinicje)
11. Rekurencja
12. Obsługa plików
13. Dynamiczne zarządzanie pamięcią
14. Omówienie zadań na zaliczenie
15. Zaliczenie wykładu

Literatura i źródła

Strona 3

1. R. Jones, I. Stewart: „*Sztuka programowania w języku C*”, WNT, Warszawa 1992. **(dla bardzo początkujących w programowaniu)**
2. G. Perry, D. Miller: „*Język C. Programowanie dla początkujących. Wydanie III*”, Helion, Gliwice 2016. **(dla początkujących w programowaniu)**
3. B. Kernighan, D. Ritchie: „*Język ANSI C*”, WNT, Warszawa 2001 („*Język ANSI C. Programowanie. Wydanie II*”, Helion, Gliwice, 2010).
4. S. Prata: „*Język C. Szkoła programowania*”, Helion, Gliwice 2016.
5. M. Tłuczek: „*Programowanie w języku C. Ćwiczenia praktyczne. Wydanie II*”, Helion, Gliwice 2011.
6. C. Delannoy: „*Ćwiczenia z języka C*”, WNT, Warszawa 1993.
7. C. Tondo, S. Gimpel: „*Język ANSI C. Ćwiczenia i rozwiązania*”, WNT, Warszawa 2004. **(rozwiązania zadań z podręcznika nr 3)**
8. M. Kotowski: „*Wysokie C*”, Lupus, Warszawa 1998. **(dla zaawansowanych)**
9. opis biblioteki standardowej: <https://cplusplus.com/reference/>
10. opis biblioteki standardowej: <https://en.cppreference.com/w/c/>

Wprowadzenie

- Dlaczego język C?
- Powstanie języka C
- Języki programowania a język C
- Język C i C++
- Popularność języka C
- Standardy języka C

Dlaczego język C?

Dlaczego warto poznać język programowania C?

- podstawa dla innych języków, które są na nim oparte (C++, Java, C#) - podobna składnia, instrukcje, operatory
- ścieżka nauki na studiach: C => C++ => Java => C#
- język wysokiego poziomu, ale „bliski” sprzętowi (*middle-level language*)
- kompilatory C tworzą wydajny, szybki kod wynikowy
- szerokie zastosowania (komputery, mikrokontrolery, systemy wbudowane)
- obecnie nadal jeden z najpopularniejszych języków programowania (indeks TIOBE, indeks PYPL)

Powstanie języka C

Strona 6



Dennis Ritchie i Ken Thompson, komputer PDP-11, 1972 rok

Powstanie języka C

Strona 7



Rozwijany w latach: 1969-1973.

Zaprojektowany w 1972 roku
w Bell Laboratories
(twórca Dennis Ritchie).

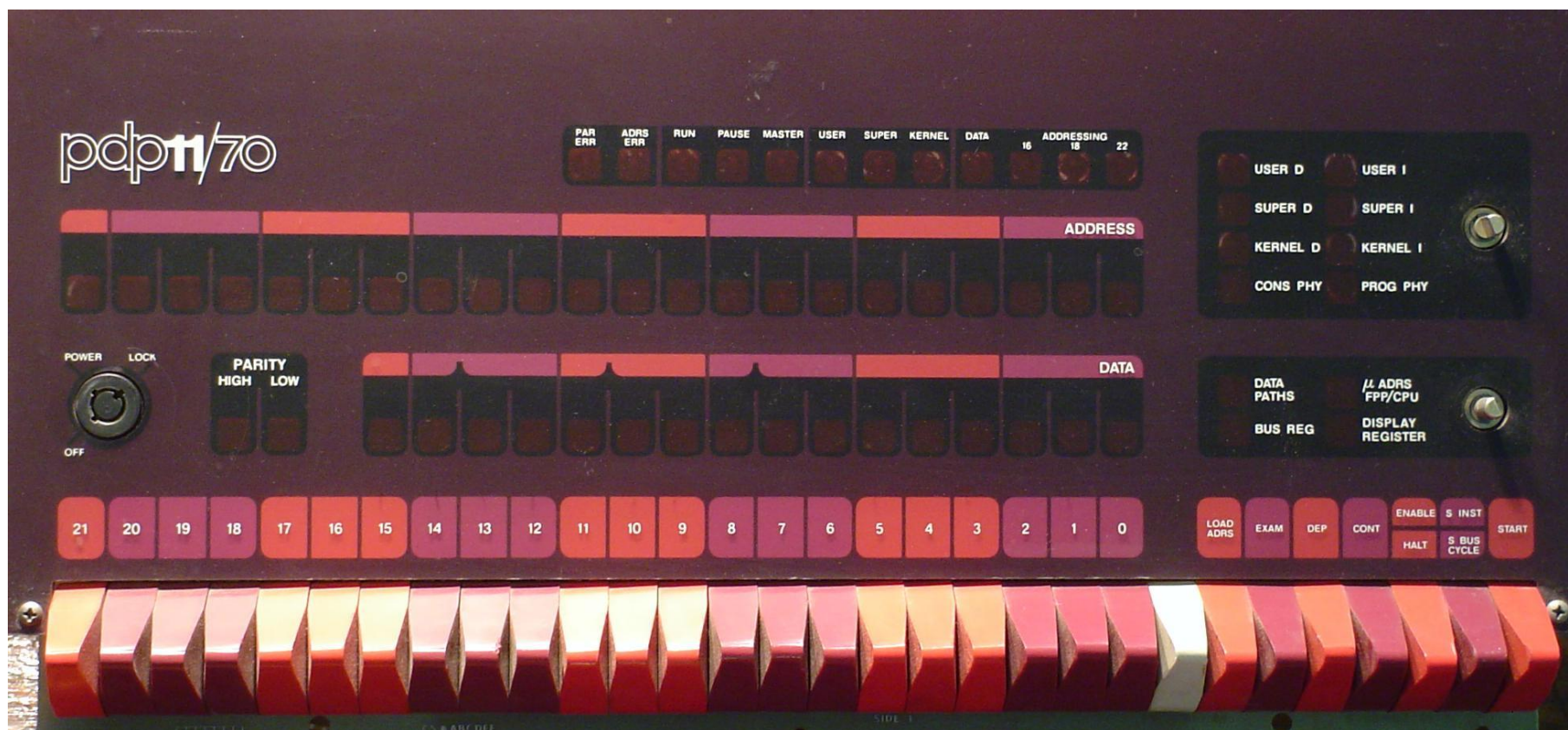
Zrealizowany pod systemem
operacyjnym UNIX
dla komputera DEC PDP-11.

Początkowo był językiem
oprogramowania systemowego
(powstał jako język do
programowania systemu UNIX).
Szybko zyskał popularność jako
uniwersalny język
programowania ogólnego
przeznaczenia.

Powstanie języka C

Strona 8

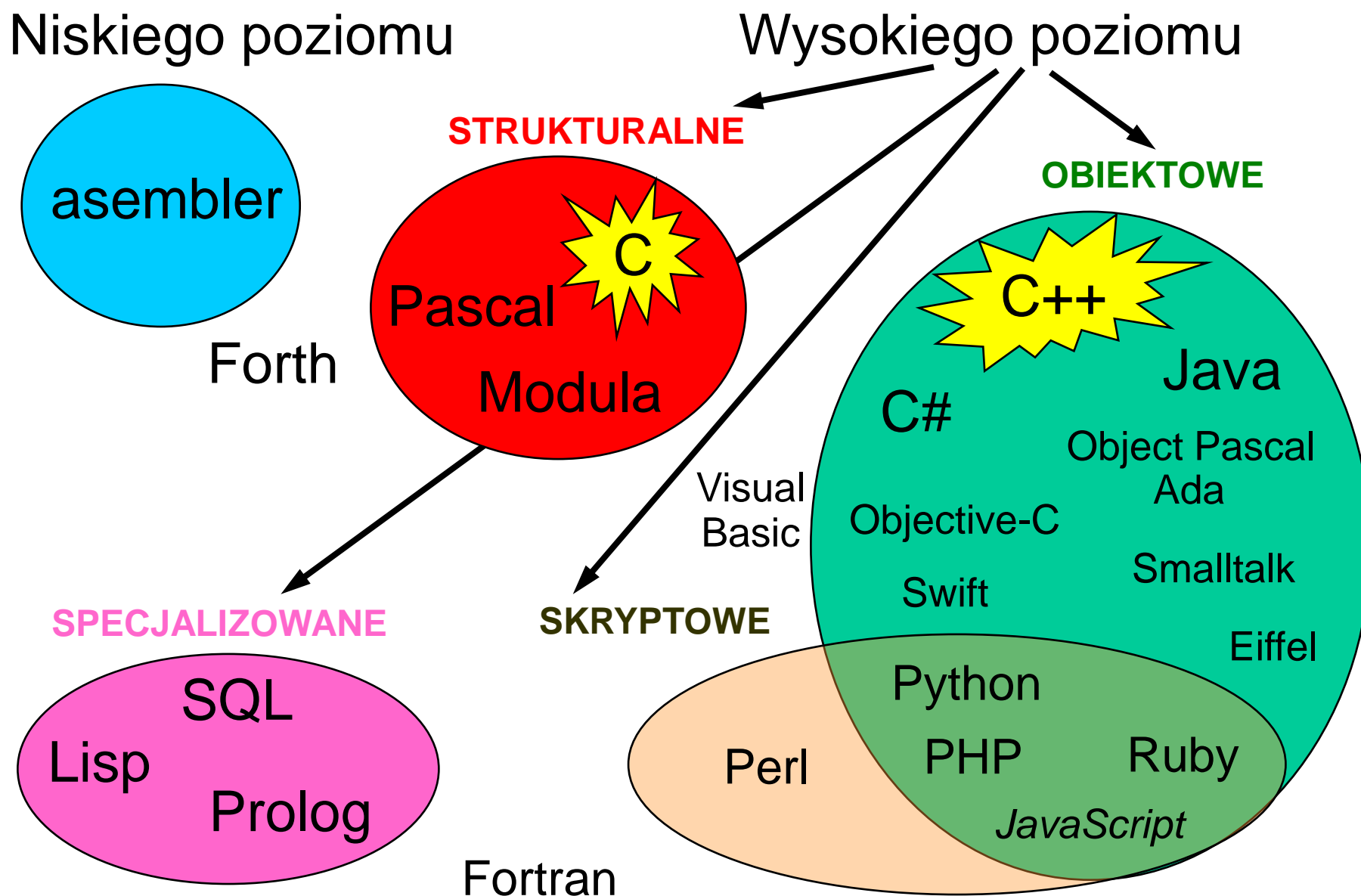
Zaglądamy do muzeum...



Źródło zdjęcia: Shieldforyoureyes Dave Fischer / Retro-Computing Society of Rhode Island institution QS:P195,Q18857750 (<https://commons.wikimedia.org/wiki/File:Pdp-11-70-panel.jpg>), „Pdp-11-70-panel“, <https://creativecommons.org/licenses/by-sa/3.0/legalcode>

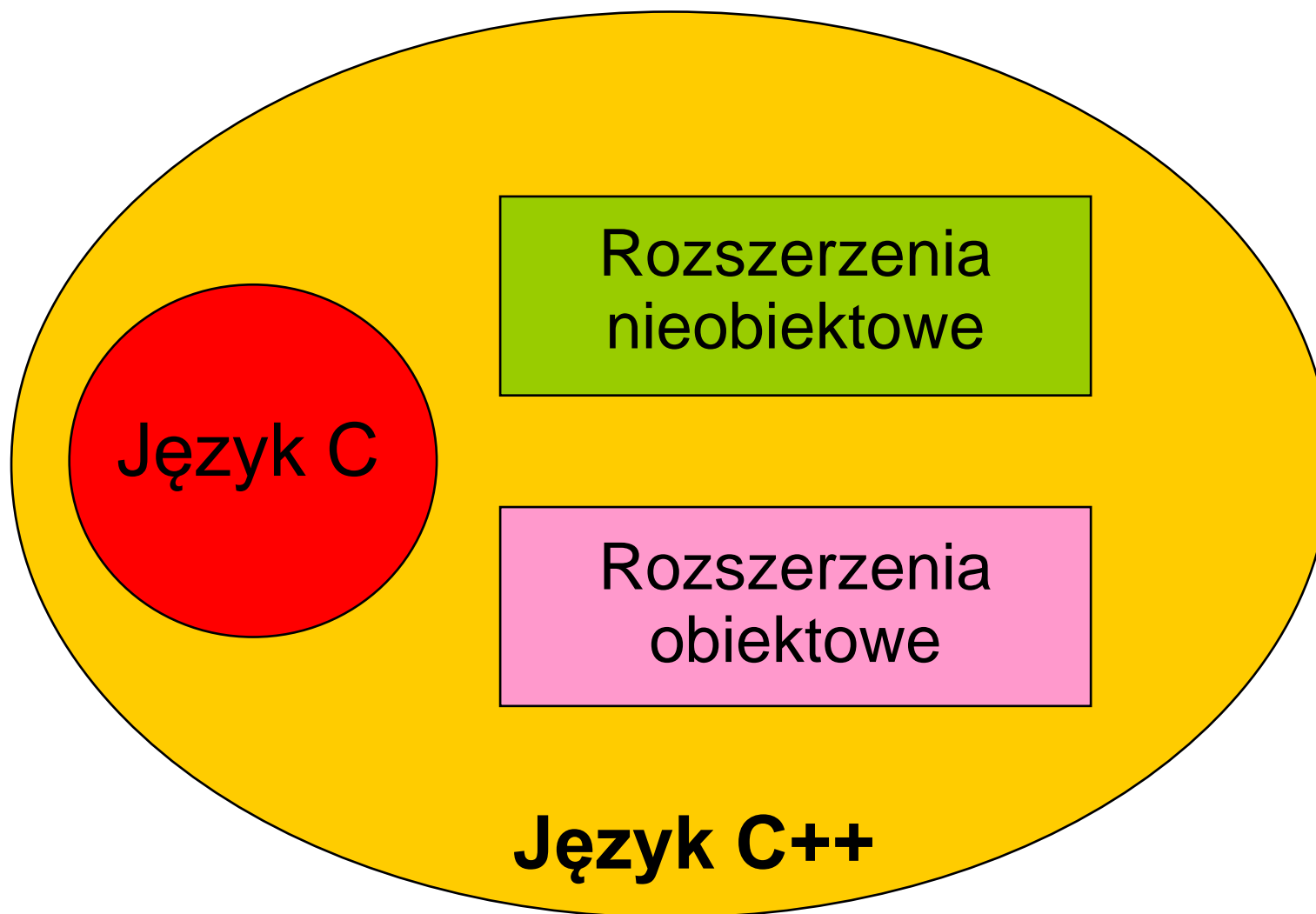
Języki programowania a język C

Strona 9



Języki C i C++

Strona 10



C++ to nadzbiór języka C.

C to podzbiór języka C++.

Każdy program w języku C jest programem w języku C++.

Rozszerzenia nieobiektywne C++

Strona 11

Elementy nieobiektywne języka C++:

Czasami „przemycane” do programów tworzonych w języku C, co wymaga użycia kompilatora C++

- ❖ Nowe mechanizmy WE/WY: strumienie `cout`, `cin` *de facto to obiekty*
- ❖ Nowy symbol komentarzy: `//` (komentarz wierszowy)
- ❖ Dowolne rozmieszczenie deklaracji
- ❖ Wyrażenia arytmetyczne w inicjalizacji: `int a = 2*n-1;`
- ❖ Przekazywanie argumentów do funkcji przez referencję

```
void zwieksz(int &a) { a++; }
```
- ❖ Argumenty domniemane funkcji

```
void ustaw(int x, int y = 2, int z = 4)
```
- ❖ Przeciążanie funkcji
- ❖ Dynamiczne zarządzanie pamięcią: `new`, `delete`
- ❖ Funkcje otwarte `inline`

Popularność języka C

www.tiobe.com

www.tiobe.com/tiobe-index/

Indeks TIOBE określa popularność języków programowania.

Jest tworzony na podstawie liczby zapytań zawierających nazwę języka, zadawanych przez użytkowników w wyszukiwarkach internetowych.

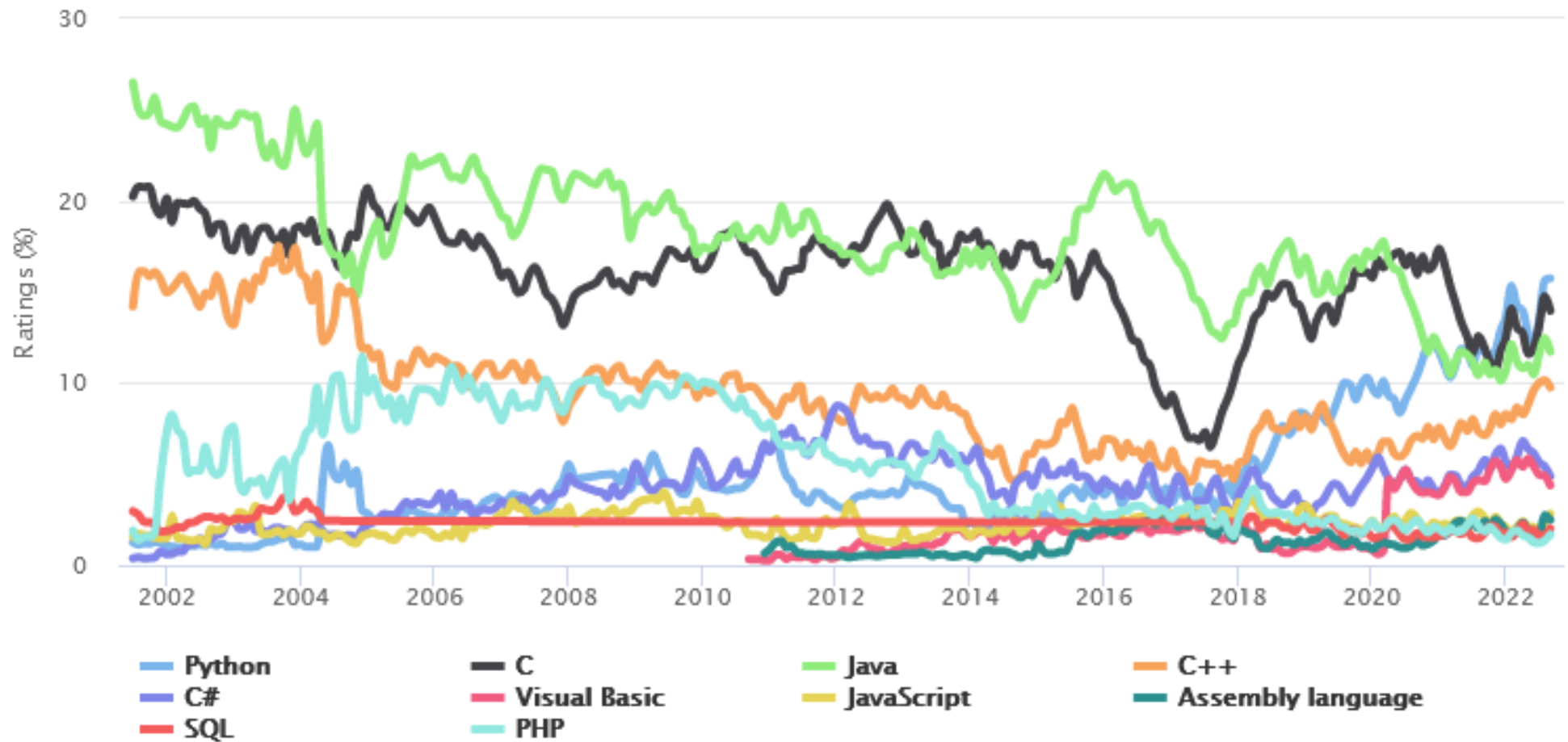
Język	TIOBE Index (październik 2020)
C	16,95%
Java	12,56%
Python	11,28%
C++	6,94%
C#	4,16%

Język programowania	TIOBE Index (wrzesień 2022)
Python	15,74%
C	13,96%
Java	11,72%
C++	9,76%
C#	4,88%

Popularność języka C

Strona 13

TIOBE Programming Community Index

Source: www.tiobe.com

Źródło rysunku: www.tiobe.com/tiobe-index/, wrzesień 2022

Popularność języka C

Strona 14

Średnie miejsca zajmowane przez poszczególne języki w ciągu roku

Programming Language	2022	2017	2012	2007	2002	1997	1992	1987
Python	1	5	8	7	12	28	-	-
C	2	2	1	2	2	1	1	1
Java	3	1	2	1	1	16	-	-
C++	4	3	3	3	3	2	2	6
C#	5	4	4	8	14	-	-	-
Visual Basic	6	14	-	-	-	-	-	-
JavaScript	7	8	10	9	8	24	-	-
Assembly language	8	10	-	-	-	-	-	-
SQL	9	-	-	-	7	-	-	-
PHP	10	7	6	5	6	-	-	-
Prolog	24	32	33	27	17	21	12	3
Lisp	33	31	13	16	13	10	4	2
Pascal	270	114	16	22	99	9	3	5
(Visual) Basic	-	-	7	4	4	3	6	4

Źródło danych: www.tiobe.com/tiobe-index/, wrzesień 2022

Popularność języka C

Strona 15

Worldwide, Oct 2022 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	28.3 %	-1.8 %
2		Java	17.2 %	-0.9 %
3		JavaScript	9.69 %	+0.4 %
4		C#	7.2 %	-0.2 %
5		C/C++	6.45 %	-0.5 %
6		PHP	5.39 %	-0.9 %
7		R	4.03 %	+0.3 %
8	↑↑↑	TypeScript	2.71 %	+1.1 %
9	↓	Objective-C	2.16 %	+0.2 %
10	↑↑	Go	2.08 %	+0.5 %
11	↓	Swift	2.05 %	+0.4 %
12	↓↓↓	Kotlin	1.81 %	+0.1 %
13	↑↑↑↑	Rust	1.57 %	+0.8 %
14	↓	Matlab	1.52 %	+0.1 %
15		Ruby	1.13 %	+0.1 %

Indeks PYPL

PopularitY
of
Programming Language

Źródło danych: <https://pypl.github.io/PYPL.html> , wrzesień 2022

Standardy języka C

Strona 16

Nazwa standardu	Rok	Opis
pierwsza wersja	1972	powstanie języka C (następca języka B z 1969)
K&R C	1978	standard Kernighan i Ritchie z podręcznika „ <i>C Programming Language</i> ” (wydanie polskie „ <i>Język C</i> ”)
ANSI C (C89)	1989	ANSI X3.159-1989 "Programming Language C"
ISO C (C90)	1990	ANSI C jako norma ISO/IEC 9899:1990
C99	1999	norma ISO/IEC 9899:1999
C11	2011	norma ISO/IEC 9899:2011
C17	2017–18	korekta C11, norma ISO/IEC 9899:2018
C2x		nowy standard planowany do ogłoszenia

Elementarz języka C

Część 1 – zagadnienia podstawowe

- Nazwy (identyfikatory)
 - Słowa kluczowe
 - Komentarze
 - Typy danych
 - Deklaracje zmiennych i stałych
 - Instrukcje
 - Operatory
 - Priorytety operatorów
 - Wyrażenia arytmetyczne
 - Wyrażenia logiczne
- } *zapowiedź*

Nazwy (identyfikatory)

Strona 18

Poprawna nazwa w języku C to ciąg znaków składający się z liter, cyfr lub znaku podkreślenia `_` (*underline*), rozpoczynający się od litery (ewentualnie od znaku `_`).

`pole``a11``x_1``obwod_kola``_wynik`~~`2litry`~~~~`obwód_koła`~~

UWAGA: w języku C rozróżniane są duże i małe litery.

Dlatego:

`liczba``LICZBA``LicznBA`

to w języku C trzy różne nazwy.

Słowa kluczowe

Strona 19

Słowa kluczowe w standardzie ANSI C / ISO C

auto	else	register	union
break	enum	return	unsigned
case	extern	short	void
char	float	signed	volatile
const	for	sizeof	while
continue	goto	static	asm
default	if	struct	near
do	int	switch	far
double	long	typedef	huge

poza standardem

Komentarze

Komentarz to tekst opisujący kod źródłowy programu. Komentarze są ignorowane przez kompilator podczas kompilacji kodu. Można je również stosować do „wyłączania” fragmentów kodu z działania.

W języku ANSI C komentarz rozpoczyna się sekwencją znaków `/*` i kończy sekwencją znaków `*/`.

Pomiędzy tymi znakami może znajdować się dowolna liczba znaków lub wierszy komentarza.

Przykłady:

```
/* Program rozwiązuje równanie kwadratowe */
```

```
/* To jest komentarz  
   umieszczony  
   w kilku wierszach */
```

```
d = b*b - 4*a*c; /* komentarz na końcu wiersza */
```

```
obwod = 2 * pi /* stała PI */ * r;
```

Typy danych

Dane w programach przechowywane są w postaci zmiennych i stałych.

W języku C wyróżniamy następujące **typy proste (wbudowane w język)**:

- stałoprzecinkowe: **char**, **short**, **int**, **long**, **long long**
- zmiennoprzecinkowe: **float**, **double**, **long double**
- typ znakowy **char**
- typ nieokreślony **void**
- **brak typu danych dla wartości logicznych**
(stosuje się typ **int**, prawda to wartość niezerowa, fałsz to wartość zerowa),
typ logiczny **bool** przyjmujący wartości **true**, **false** dostępny od standardu C99 (biblioteka **stdbool.h**)
- **brak typu danych dla łańcuchów znakowych (napisów)**
(stosuje się tablice znakowe lub typ wskaźnikowy **char***)

typy arytmetyczne (liczbowe)

od C99

Typy danych

Typy stałoprzecinkowe

typ	liczba bitów N	zakres wartości $-2^{N-1} \dots 2^{N-1} - 1$
char	8	-128 ... 127
short	16	-32 768 ... 32 767
int	16	-32 768 ... 32 767
	32	-2 147 483 648 ... 2 147 483 647
long	32	-2 147 483 648 ... 2 147 483 647
long long	64	-9 223 372 036 854 775 808 ... 9 223 372 036 854 775 807

od C99

Typ **short** nazywany jest także **short int**.

Typ **long** nazywany jest także **long int**.

Typ **long long** nazywany jest także **long long int**.

Typy danych

Typy stałoprzecinkowe

Domyślnie, wszystkie typy stałoprzecinkowe opisują wartości ze znakiem (**signed**) reprezentowane binarnie w kodzie U2.

Użycie kwalifikatora **unsigned** oznacza wartości bez znaku reprezentowane binarnie w naturalnym kodzie dwójkowym NKB.

typ	liczba bitów N	zakres wartości $0 \dots 2^N - 1$
unsigned char	8	0 ... 255
unsigned short	16	0 ... 65 535
unsigned int	16	0 ... 65 535
	32	0 ... 4 294 967 295
unsigned long	32	0 ... 4 294 967 295
unsigned long long	64	0 ... 18 446 744 073 709 551 615

Typy danych

Typy zmiennoprzecinkowe

typ	liczba bitów	liczba cyfr znaczących	zakres wartości
float	32	7	$3.4 \cdot 10^{-38} \dots 3.4 \cdot 10^{38}$
double	64	15	$1.7 \cdot 10^{-308} \dots 1.7 \cdot 10^{308}$
long double	80	19	$3.4 \cdot 10^{-4932} \dots 1.1 \cdot 10^{4932}$

Wszystkie typy zmiennoprzecinkowe opisują wartości ze znakiem.

Nie jest dozwolone stosowanie kwalifikatorów **signed** i **unsigned**.

Deklaracje zmiennych i stałych

Deklaracje zmiennych i stałych

Deklaracja:

- nadaje nazwę,
- określa typ przechowywanej wartości,
- rezerwuje odpowiedni obszar pamięci.

Deklaracja zmiennej:

```
char znak;  
int i, j, k;  
float x, y;
```

Inicjalizacja zmiennej:

```
char znak = 'X';  
int i = 0;  
float pi = 3.14;
```

Deklaracja stałej:

```
const char LITERA = 'a';  
const int N = 10;  
const float PI = 3.14;
```

Deklaracje zmiennych i stałych

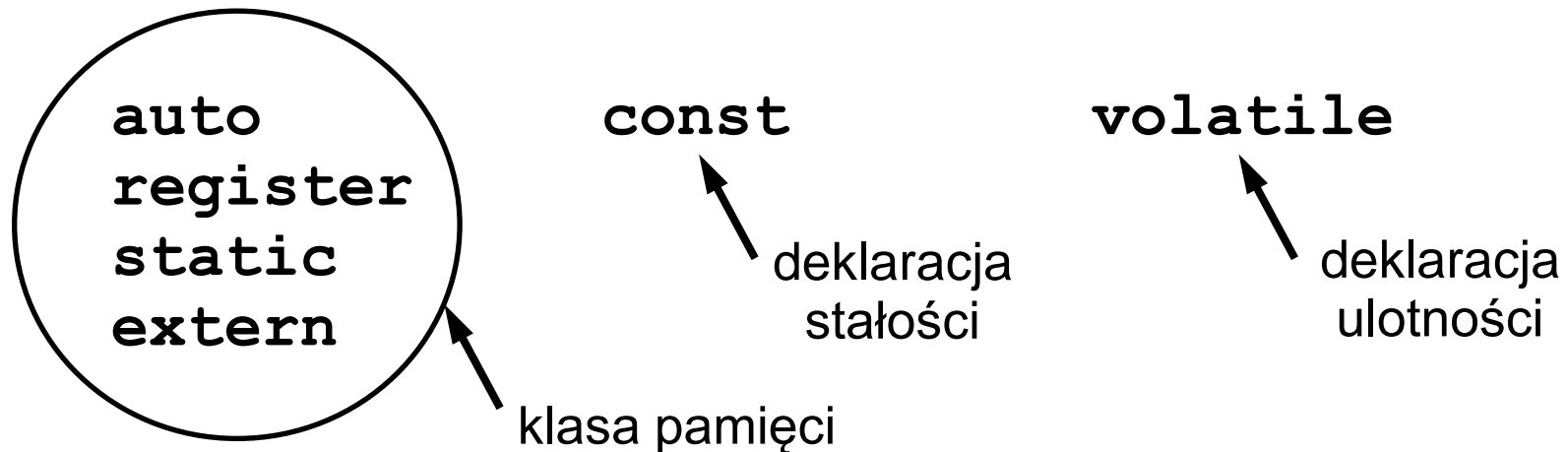
Strona 26

Rodzaje zmiennych

Zmienne globalne (tworzone na zewnątrz funkcji) są dostępne w całym programie od miejsca deklaracji. **Są automatycznie zerowane.**

Zmienne lokalne (tworzone wewnątrz funkcji) są tworzone tylko na czas działania funkcji (po jej wykonaniu są usuwane), są dostępne tylko wewnątrz funkcji. **Po utworzeniu przyjmują przypadkowe wartości.**

W deklaracjach zmiennych możliwe jest stosowanie następujących kwalifikatorów umieszczanych przed nazwą typu zmiennej:



Instrukcje

- ❖ prosta
- ❖ złożona { }

Cecha języka C:
mały zestaw instrukcji

- ❖ pusta ;
- ❖ warunkowa prosta **if**
i warunkowa złożona **if-else**
- ❖ wyboru **switch**
- ❖ iteracyjne (pętle): **for**, **while**, **do-while**
- ❖ przerwania pętli **break**
- ❖ kontynuacji pętli **continue**
- ❖ skoku **goto**
- ❖ powrotu z funkcji **return**

Operatory

Zestaw operatorów języka C

Cecha języka C:
rozbudowany zestaw operatorów

- jednoargumentowe
- dwuargumentowe
- trójargumentowy `?:`
- wieloargumentowy `()`

arytmetyczne `+` `-` `*` `/` `%`

zwiększania i zmniejszania `++` `--`

relacji `<` `<=` `>` `>=` `==` `!=`

logiczne `&&` `||` `!`

bitowe `&` `|` `^` `<<` `>>` `~`

przypisania `=` `+=` `-=` `*=` `/=` `%=`

`&=` `|=` `^=` `<<=` `>>=` `~=`

inne `()` `[]` `&` `*` `->` `.` `,` `?:` `sizeof`

Strona 29

1.	()	[]	->	.
2.	!	~	++ --	+ - * & sizeof
3.	*	/	%	
4.	+	-	przedrostkowe	jednoargumentowe
5.	<<	>>		
6.	<	<=	>	>=
7.	= =	!=		
8.	&			
9.	^			
10.				
11.	&&			
12.				
13.	? :			
14.	=	+=	-=	*= /= %= ^= = <<= >>=
15.	/			

Kolejność wykonywania działań

Strona 30

$$\frac{a + b}{c - d}$$

$$a + \boxed{b / c} - d$$

ŹŁE

$$c - d$$

$$(a + b) / (c - d)$$

DOBRE

1. $\boxed{()}$
2. $++$ $--$ $+$ $-$ (jednoargumentowe)
3. $*$ $\boxed{/}$ $\%$
4. $\boxed{+}$ $\boxed{-}$
5. $=$ $+=$ $-=$ $*=$ $/=$ $\%=$

Kolejność wykonywania działań

Strona 31

$$\frac{-b}{2a}$$

$$- \boxed{b / 2} * a$$

ŹŁE

$$- b / (2 * a)$$

DOBRE

$$- b / 2 / a$$

1. $\boxed{()}$
2. $++$ $--$ $+$ $\boxed{-}$ (jednoargumentowe)
3. $\boxed{*}$ $\boxed{/}$ $\%$
4. $+$ $-$
5. $=$ $+=$ $-=$ $*=$ $/=$ $\%=$

Wyrażenia arytmetyczne

$$x = \frac{-b + \sqrt{\text{delta}}}{2a}$$

```
x = (-b+sqrt(delta)) / (2*a);
```

**Dzielenie całkowite „obcina”
część ułamkową.**

UWAGA: 9 / 2 = 4

9/2.0 = 4.5

9.0/2 = 4.5

9.0/2.0 = 4.5

**Rzutowanie typu
(type cast)**

```
int a = 9, b = 2;
```

```
float c;
```

```
c = a/b; 4.0
```

```
c = (float)a/b; 4.5
```

```
c = a/(float)b; 4.5
```

$$c = \frac{a+b}{2}$$

```
c = (a + b) / 2.0;
```

Wyrażenia logiczne

czy **a** jest równe **b** ?

a == b

czy **x** jest większe od **y**
i jednocześnie **a** jest różne od **b** ?

x > y && a != b

łączenie warunków:

a1 == b1 && a2 == b2 && a3 == b3

a1 == b1 || a2 == b2 || a3 == b3