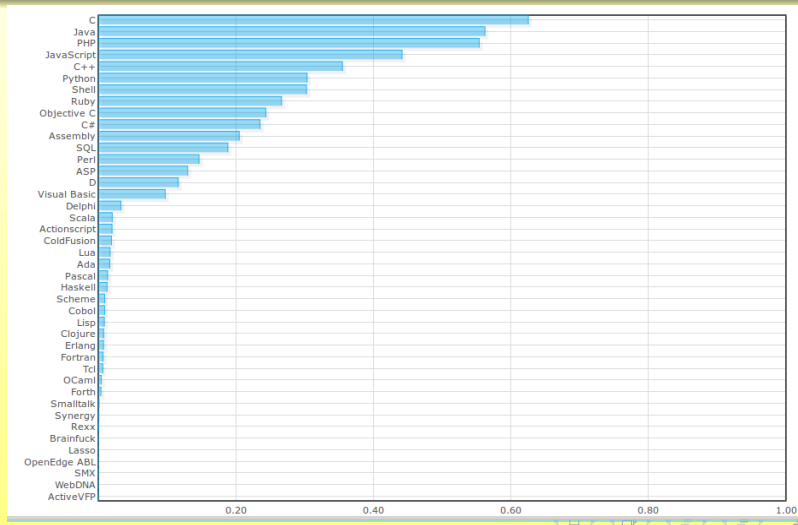


Wstęp do programowania w C

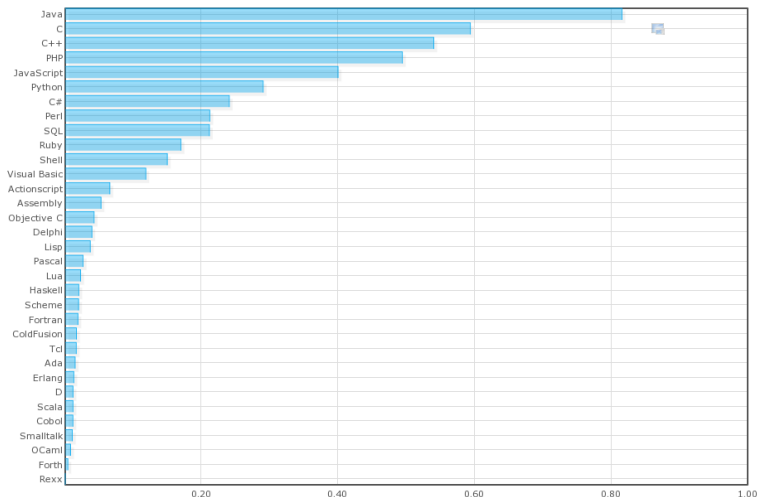
Marek Piotrów - Wykład 1

7 października 2014

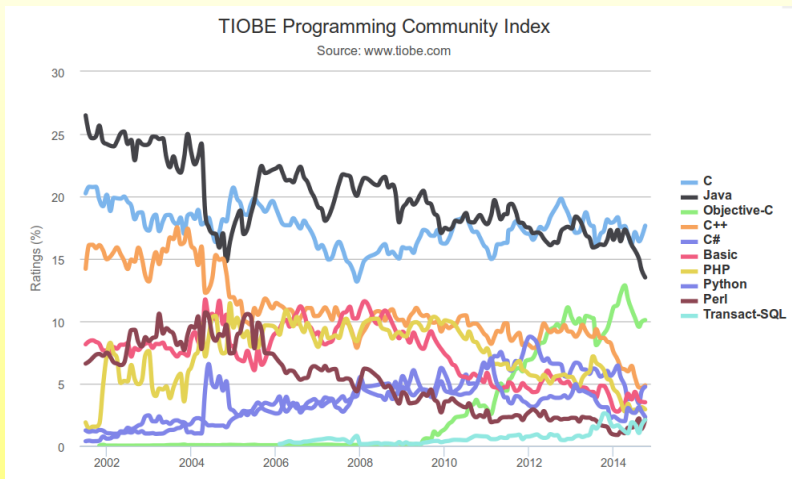
Najbardziej popularne języki programowania (według www.langpop.com) - dane z 2014 roku



Najbardziej popularne języki programowania (według www.langpop.com) - dane z 2011 roku



Zmiany w popularności języków programowania (według *www.tiobe.com*) - dane z października 2014



Źródła sukcesu języka C

- Zaprojektowany przez programistę-praktyka Dennisa Ritchie'go dla praktyków. Użyty do napisania systemu UNIX.
- Ściśle związany z systemem UNIX oraz Linux - stale dostępne kompilatory.
- Książka Kernighan'a i Ritchie'go definiująca język przed wydaniem standardu.
- Język wysokiego poziomu umożliwiający programowanie na poziomie sprzętowym.
- Rozszerzalny przez biblioteki, bogata biblioteka standardowa.

Literatura i inne źródła

- B.W. Kernighan i D.M. Ritchie, *Język ANSI C*, WNT Warszawa, 2003.
- King K.N.: *Język C, Nowoczesne programowanie*, Wydanie II, Helion 2011 (uwzględnia standard C99).
- Podstawowe środowisko pracy: Code::Blocks
<http://www.codeblocks.org>
- Podstawowy kompilator: GCC
<http://www.mingw.org> oraz <http://gcc.gnu.org>
- Materiały z wykładu:
<http://kno.ii.uni.wroc.pl/ii> - trzeba się zalogować i zapisać na kurs (jednorazowe hasło do kursu podano na wykładzie).
- Podstawowy standard języka C: ISO/IEC 9899:1999.
- Wymagane opcje kompilatora gcc: `-xc -std=c99 -Wall -Wextra -Werror`

Zasady zaliczenia kursu

Składniki oceny

- 50% - zadania z 10 list rozwiązywane na pracowni;
- 25% - zadanie końcowe (4-tygodniowe);
- 25% - kolokwium na wykładzie w styczniu.

Przeliczanie procentów na ocenę

- 40% - dostateczny (3,0);
- 55% - dostateczny plus (3,5);
- 65% - dobry (4.0);
- 75% - dobry plus (4,5);
- 85% - bardzo dobry (5,0).

Język programowania C

C jest językiem:

- imperatywnym;
- proceduralnym;
- o strukturze blokowej;
- ogólnego przeznaczenia.

Język programowania jest opisywany poprzez

- leksykę (wyodrębnienie słów, operatorów i separatorów);
- składnię (określenie poprawnych ciągów symboli leksykalnych);
- semantykę (określenie znaczenia poprawnych konstrukcji językowych).

Język programowania C

C jest językiem:

- imperatywnym;
- proceduralnym;
- o strukturze blokowej;
- ogólnego przeznaczenia.

Język programowania jest opisywany poprzez

- leksykę (wyodrębnienie słów, operatorów i separatorów);
- składnię (określenie poprawnych ciągów symboli leksykalnych);
- semantykę (określenie znaczenia poprawnych konstrukcji językowych).

Język programowania C

C jest językiem:

- imperatywnym;
- proceduralnym;
- o strukturze blokowej;
- ogólnego przeznaczenia.

Język programowania jest opisywany poprzez

- leksykę (wyodrębnienie słów, operatorów i separatorów);
- składnię (określenie poprawnych ciągów symboli leksykalnych);
- semantykę (określenie znaczenia poprawnych konstrukcji językowych).

Język programowania C

C jest językiem:

- imperatywnym;
- proceduralnym;
- o strukturze blokowej;
- ogólnego przeznaczenia.

Język programowania jest opisywany poprzez

- leksykę (wyodrębnienie słów, operatorów i separatorów);
- składnię (określenie poprawnych ciągów symboli leksykalnych);
- semantykę (określenie znaczenia poprawnych konstrukcji językowych).

Język programowania C

C jest językiem:

- imperatywnym;
- proceduralnym;
- o strukturze blokowej;
- ogólnego przeznaczenia.

Język programowania jest opisywany poprzez

- leksykę (wyodrębnienie słów, operatorów i separatorów);
- składnię (określenie poprawnych ciągów symboli leksykalnych);
- semantykę (określenie znaczenia poprawnych konstrukcji językowych).

Język programowania C

C jest językiem:

- imperatywnym;
- proceduralnym;
- o strukturze blokowej;
- ogólnego przeznaczenia.

Język programowania jest opisywany poprzez

- leksykę (wyodrębnienie słów, operatorów i separatorów);
- składnię (określenie poprawnych ciągów symboli leksykalnych);
- semantykę (określenie znaczenia poprawnych konstrukcji językowych).

Język programowania C

C jest językiem:

- imperatywnym;
- proceduralnym;
- o strukturze blokowej;
- ogólnego przeznaczenia.

Język programowania jest opisywany poprzez

- leksykę (wyodrębnienie słów, operatorów i separatorów);
- składnię (określenie poprawnych ciągów symboli leksykalnych);
- semantykę (określenie znaczenia poprawnych konstrukcji językowych).

Krótką historia języka C i C++

- 1977 - pierwszy opis języka C Kernighan'a i Ritchie'go;
- 1983 - ANSI tworzy komitet do opisu standardu;
- 1985 - opis języka C++ przez B. Stroustrup'a;
- 1989 - opublikowanie standardu ANSI C;
- 1990 - przekazanie prac nad standardem do ISO/IEC;
opublikowanie standardu ISO/IEC 9899:1990 (tzw. c90);
- 1994-96 - poprawki do standardu c90;
- 1995 - rozpoczęcie prac w ISO/IEC nad nowym standardem;
- 1998 - opublikowanie standardu języka C++;
- 1999 - nowy standard języka C: ISO/IEC 9899:1999 (tzw. c99);
- 2001, 2004, 2007 - poprawki do standardu c99;
- 2011 - nowe standardy ISO/IEC dla C i C++ (tzw. c11 i c++11)

Krótką historia języka C i C++

- 1977 - pierwszy opis języka C Kernighan'a i Ritchie'go;
- 1983 - ANSI tworzy komitet do opisu standardu;
- 1985 - opis języka C++ przez B. Stroustrup'a;
- 1989 - opublikowanie standardu ANSI C;
- 1990 - przekazanie prac nad standardem do ISO/IEC;
opublikowanie standardu ISO/IEC 9899:1990 (tzw. c90);
- 1994-96 - poprawki do standardu c90;
- 1995 - rozpoczęcie prac w ISO/IEC nad nowym standardem;
- 1998 - opublikowanie standardu języka C++;
- 1999 - nowy standard języka C: ISO/IEC 9899:1999 (tzw. c99);
- 2001, 2004, 2007 - poprawki do standardu c99;
- 2011 - nowe standardy ISO/IEC dla C i C++ (tzw. c11 i c++11)

Krótką historia języka C i C++

- 1977 - pierwszy opis języka C Kernighan'a i Ritchie'go;
- 1983 - ANSI tworzy komitet do opisu standardu;
- 1985 - opis języka C++ przez B. Stroustrup'a;
- 1989 - opublikowanie standardu ANSI C;
- 1990 - przekazanie prac nad standardem do ISO/IEC;
opublikowanie standardu ISO/IEC 9899:1990 (tzw. c90);
- 1994-96 - poprawki do standardu c90;
- 1995 - rozpoczęcie prac w ISO/IEC nad nowym standardem;
- 1998 - opublikowanie standardu języka C++;
- 1999 - nowy standard języka C: ISO/IEC 9899:1999 (tzw. c99);
- 2001, 2004, 2007 - poprawki do standardu c99;
- 2011 - nowe standardy ISO/IEC dla C i C++ (tzw. c11 i c++11)

Krótką historia języka C i C++

- 1977 - pierwszy opis języka C Kernighan'a i Ritchie'go;
- 1983 - ANSI tworzy komitet do opisu standardu;
- 1985 - opis języka C++ przez B. Stroustrup'a;
- 1989 - opublikowanie standardu ANSI C;
- 1990 - przekazanie prac nad standardem do ISO/IEC;
opublikowanie standardu ISO/IEC 9899:1990 (tzw. c90);
- 1994-96 - poprawki do standardu c90;
- 1995 - rozpoczęcie prac w ISO/IEC nad nowym standardem;
- 1998 - opublikowanie standardu języka C++;
- 1999 - nowy standard języka C: ISO/IEC 9899:1999 (tzw. c99);
- 2001, 2004, 2007 - poprawki do standardu c99;
- 2011 - nowe standardy ISO/IEC dla C i C++ (tzw. c11 i c++11)

Krótką historia języka C i C++

- 1977 - pierwszy opis języka C Kernighan'a i Ritchie'go;
- 1983 - ANSI tworzy komitet do opisu standardu;
- 1985 - opis języka C++ przez B. Stroustrup'a;
- 1989 - opublikowanie standardu ANSI C;
- 1990 - przekazanie prac nad standardem do ISO/IEC;
opublikowanie standardu ISO/IEC 9899:1990 (tzw. c90);
- 1994-96 - poprawki do standardu c90;
- 1995 - rozpoczęcie prac w ISO/IEC nad nowym standardem;
- 1998 - opublikowanie standardu języka C++;
- 1999 - nowy standard języka C: ISO/IEC 9899:1999 (tzw. c99);
- 2001, 2004, 2007 - poprawki do standardu c99;
- 2011 - nowe standardy ISO/IEC dla C i C++ (tzw. c11 i c++11)

Przykład 1

```
#include <stdio.h>
```

```
/* wypisz zestawienie stanu konta, jesli kwote 10 tys. zl.  
   wlozylismy na lokate 10 letnia oprocentowana 5% rocznie  
   i procenty sa dopisywane co kwartal. */
```

```
int main(void) // wersja zmiennoprzecinkowa
```

```
{  
    int lata, okresow_w_roku;  
    float procent, kwota_pocz, kwota;  
    int rok, okres;  
  
    kwota_pocz=10000; /* kwota lokaty */  
    lata=10;          /* liczba lat lokaty */  
    okresow_w_roku=4; /* liczba okresow w roku */  
    procent=0.05;      /* oprocentowanie roczne */  
  
    kwota=kwota_pocz; rok=1;  
    while (rok <= lata) {  
        okres=1;  
        while (okres <= okresow_w_roku) {  
            kwota=kwota*(1.0+procent/okresow_w_roku);  
            okres=okres+1;  
        }  
        printf("%d\t\t%9.2f\n",rok,kwota);  
        rok=rok+1;  
    }  
    return 0;  
}
```

Niektóre cechy dobrego programowania

- **Czytelność:** formatowanie tekstu, właściwe identyfikatory, komentarze.
- **Poprawność:** zgodność ze specyfikacją, brak błędów.
- **Efektywność:** dobór właściwych algorytmów i struktur danych.
- **Przenośność:** zgodność ze standardem języka.

Niektóre cechy dobrego programowania

- Czytelność: formatowanie tekstu, właściwe identyfikatory, komentarze.
- Poprawność: zgodność ze specyfikacją, brak błędów.
- Efektywność: dobór właściwych algorytmów i struktur danych.
- Przenośność: zgodność ze standardem języka.

Niektóre cechy dobrego programowania

- Czytelność: formatowanie tekstu, właściwe identyfikatory, komentarze.
- Poprawność: zgodność ze specyfikacją, brak błędów.
- Efektywność: dobór właściwych algorytmów i struktur danych.
- Przenośność: zgodność ze standardem języka.

Niektóre cechy dobrego programowania

- Czytelność: formatowanie tekstu, właściwe identyfikatory, komentarze.
- Poprawność: zgodność ze specyfikacją, brak błędów.
- Efektywność: dobór właściwych algorytmów i struktur danych.
- Przenośność: zgodność ze standardem języka.

Przykład 2

```
#include <stdio.h>
```

```
/* wypisz zestawienie stanu konta, jesli kwote 10 tys. zl.  
   wlozylismy na lokate 10 letnia oprocentowana 5% rocznie  
   i procenty sa dopisywane co kwartal. */
```

```
int main(void) // wersja staloprzecinkowa
```

```
{  
    int kwota_pocz, lata, okresow_w_roku;  
    float procent;  
    int rok, okres, kwota;  
  
    kwota_pocz=10000*100; /* kwota lokaty w groszach */  
    lata=10; /* liczba lat lokaty */  
    okresow_w_roku=4; /* liczba okresow w roku */  
    procent=0.05; /* oprocentowanie roczne */  
  
    kwota=kwota_pocz; rok=1;  
    while (rok <= lata) {  
        okres=1;  
        while (okres <= okresow_w_roku) {  
            kwota=kwota*(1.0+procent/okresow_w_roku);  
            okres=okres+1;  
        }  
        printf("%d\t\t%.2f\n",rok,kwota/100.0);  
        rok=rok+1;  
    }  
    return 0;  
}
```

Przykład 3

```
#include <stdio.h>
```

/ wypisz zestawienie stanu konta, jeśli kwotę 10 tys. zł.
włożyliśmy na lokatę 10 letnią oprocentowaną 5% rocznie
i procenty są dopisywane co miesiąc. */*

```
int main(void) // wersja stałoprzecinkowa
{
    int kwota_pocz, lata, okresow_w_roku;
    float procent;
    int kwota;

    kwota_pocz=10000*100; /* kwota lokaty w groszach */
    lata=10;             /* liczba lat lokaty */
    okresow_w_roku=12;   /* liczba okresow w roku */
    procent=0.05;        /* oprocentowanie roczne */

    kwota=kwota_pocz;
    for (int rok=1; rok <= lata; rok++) {
        for (int okres=1; okres <= okresow_w_roku; okres++)
            kwota=kwota*(1.0+procent/okresow_w_roku);
        printf("%d\t%9.2f\n",rok,kwota/100.0);
    }
    return 0;
}
```

Przykład 4

```
#include <stdio.h>
```

```
/* wypisz zestawienie stanu konta, jesli kwote 10 tys. zl.  
   wlozyliśmy na lokate 10 letnia oprocentowana 5% rocznie  
   i procenty sa dopisywane co miesiac. */
```

```
#define KWOTA_POCZ (10000*100) /* kwota lokaty w groszach */  
#define LATA 10 /* liczba lat lokaty */  
#define OKRESOW_W_ROKU 12 /* liczba okresow w roku */  
#define PROCENT 0.05 /* oprocentowanie roczne */
```

```
int main(void) // wersja staloprzecinkowa
```

```
{  
    int kwota=KWOTA_POCZ;  
  
    for (int rok=1; rok <= LATA; rok++) {  
        for (int okres=1; okres <= OKRESOW_W_ROKU; okres++)  
            kwota=kwota*(1.0+PROCENT/OKRESOW_W_ROKU);  
        printf("%d\t%9.2f\n",rok,kwota/100.0);  
    }  
    return 0;  
}
```

Przykład 5 - rozkład liczby na czynniki

```
#include <stdio.h>

/* wypisz rozkład na czynniki pierwsze podanych liczb naturalnych */

int main(void)
{
    unsigned int liczba, czynnik;

    while (1) {
        printf("Podaj liczbę do rozkładu na czynniki pierwsze (0 - koniec): ");
        scanf("%u",&liczba);
        if (liczba == 0) return 0;

        printf("Czynniki pierwsze liczby %u\n to: ",liczba);
        while (liczba % 2 == 0) {
            liczba=liczba/2;
            printf(" 2");
        }

        for (czynnik=3; czynnik*czynnik <= liczba; czynnik=czynnik+2)
            while (liczba % czynnik == 0) {
                liczba=liczba/czynnik;
                printf(" %u",czynnik);
            }
        if (liczba > 1) printf(" %u\n",liczba); else putchar(' \n');
    }
    return 0;
}
```