

Technologie Informacyjne

Języki komputerowe

Adam Krasuski

Szkoła Główna Służby Pożarniczej
Zakład Informatyki i Łączności

December 19, 2016

- 1 Wprowadzenie
- 2 Elementy programowania
- 3 Typy programowania
- 4 Inne języki komputerowe
- 5 Top 10 Technologii

Definicje

Język komputerowy

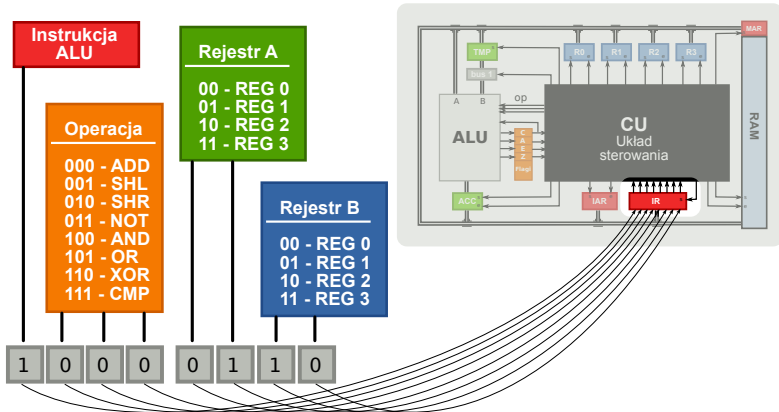
jest to język formalny zaprojektowany do komunikowania się z maszyną, w szczególności komputerem.

Programowanie






proces opracowywania i wdrażania różnych zestawów instrukcji, w celu umożliwienia komputerowi realizację określonego celu.

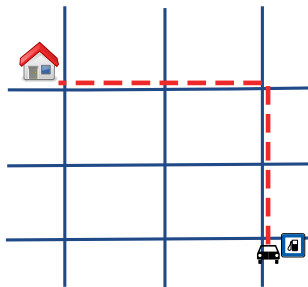
Język komputerowy

ADD R1 R2 1 000 01 10



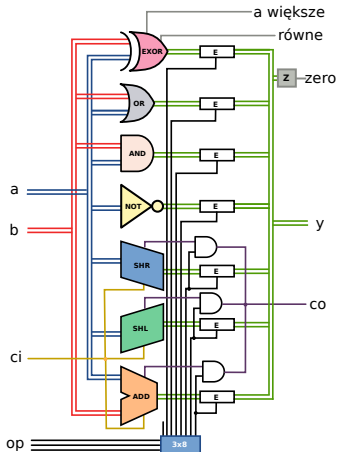
Programowanie

- start ↑ Prosto 1000 m
-  na 1 skrzyżowaniu prosto
-  na 2 skrzyżowaniu w lewo
-  na 3 skrzyżowaniu prosto
-  na 4 skrzyżowaniu w prawo
-  pierwszy dom po lewej



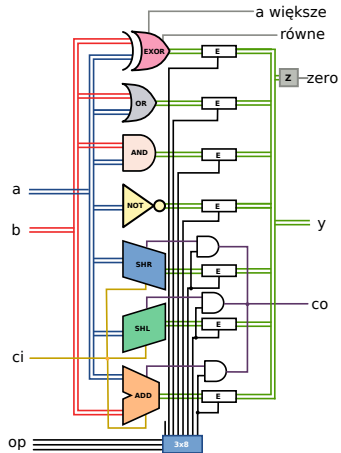
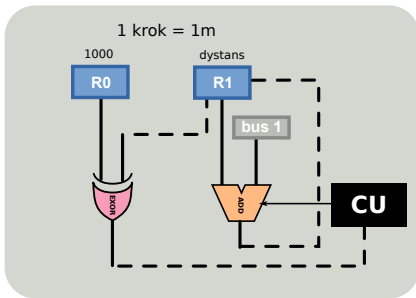
Programowanie

- start ↑ Prosto 1000 m
- ↑ na 1 skrzyżowaniu prosto
- ↶ na 2 skrzyżowaniu w lewo
- ↑ na 3 skrzyżowaniu prosto
- ↷ na 4 skrzyżowaniu w prawo
- 📍 pierwszy dom po lewej

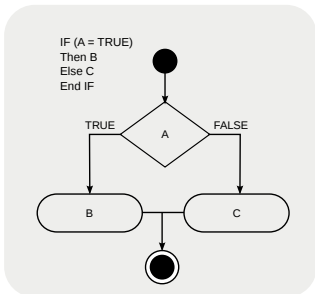


Programowanie

Prosto 1000 m

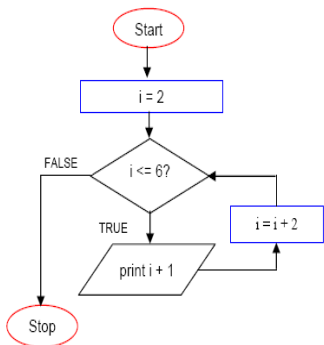


Warunki IF-THEN



```
if (a<=2) {  
    print (a);  
}else{  
    print (b);  
}
```


Pętle FOR,

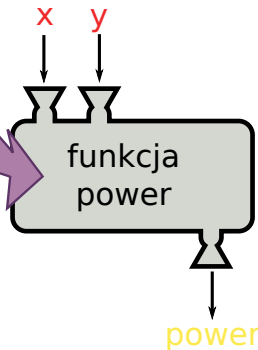


```
for (i=1,i<=6,i++) {  
    print (i);  
}
```

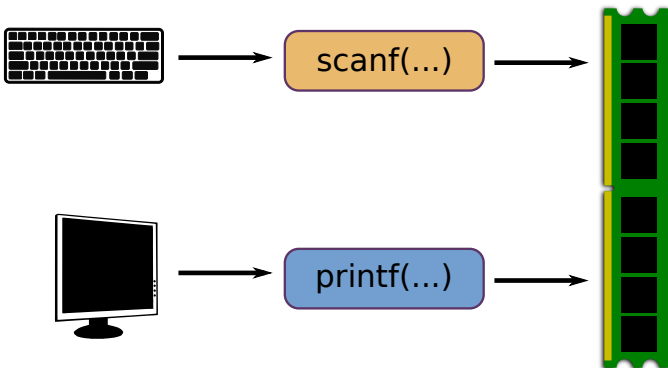
Funkcje

 x^y

```
power = 1;
for (i=1, i<=y, i++) {
    power=power*x;
}
print (power);
```

`power(x, y)`

wejście, wyjście



zmienne

General purpose RAM								7F
								30
7F	7E	7D	7C	7B	79	78		2F
77	76	75	74	73	72	71		2E
6F	6E	6D	6C	6B	69	68		2D
67	66	65	64	63	62	61		2C
6F	5E	7D	5C	5B	59	58		2B
57	56	55	54	53	52	51		2A
4F	4E	4D	4C	4B	49	48		29
47	46	45	44	43	42	41		28
3F	3E	3D	3C	3B	39	38		27
37	36	35	34	33	32	31		26
2F	2E	2D	2C	2B	29	28		25
27	26	25	24	23	22	21		24
1F	1E	1D	1C	1B	19	18		23
17	16	15	14	13	12	11		22
0F	0E	0D	0C	0B	09	08		21
07	06	05	04	03	02	01		20
Register Banks								1F
								00

67 →

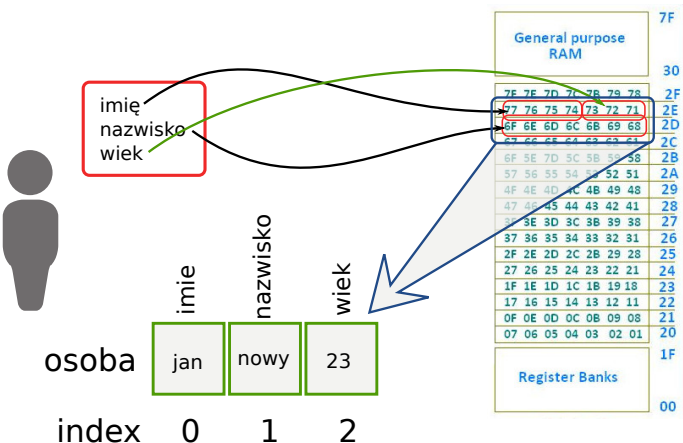
→ [11 - 14]

zmienne

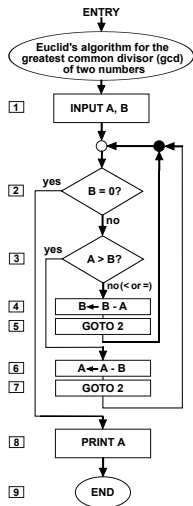
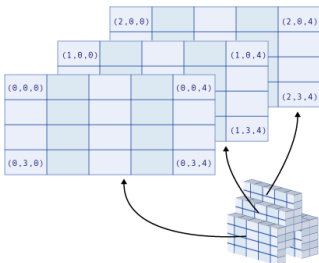
```
power = 1;
for (i=1,i<=y,i++){
    power=power*x;
}
print (power);
```

General purpose RAM								7F
								30
7F	7E	7D	7C	7B	79	78	2F	
77	76	75	74	73	72	71	2E	
6F	6E	6D	6C	6B	69	68	2D	
67	66	65	64	63	62	61	2C	
6F	6E	7D	5C	5B	59	58	2B	
57	56	55	54	53	52	51	2A	
4F	4E	4D	4C	4B	49	48	29	
47	46	45	44	43	42	41	28	
3F	3E	3D	3C	3B	39	38	27	
37	36	35	34	33	32	31	26	
2F	2E	2D	2C	2B	29	28	25	
27	26	25	24	23	22	21	24	
1F	1E	1D	1C	1B	19	18	23	
17	16	15	14	13	12	11	22	
0F	0E	0D	0C	0B	09	08	21	
07	06	05	04	03	02	01	20	
Register Banks								1F
								00

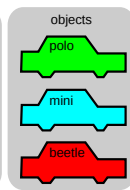
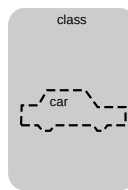
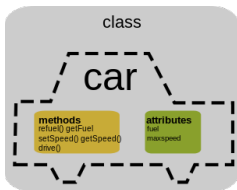
zmienne



programowanie proceduralne

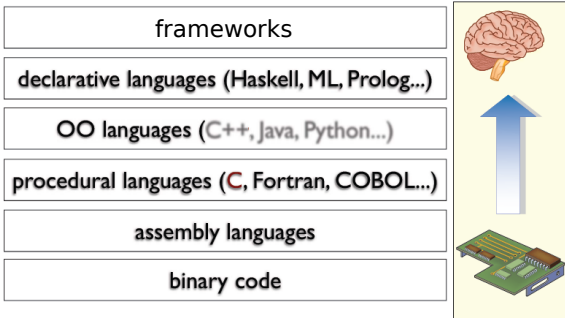


programowanie obiektowe

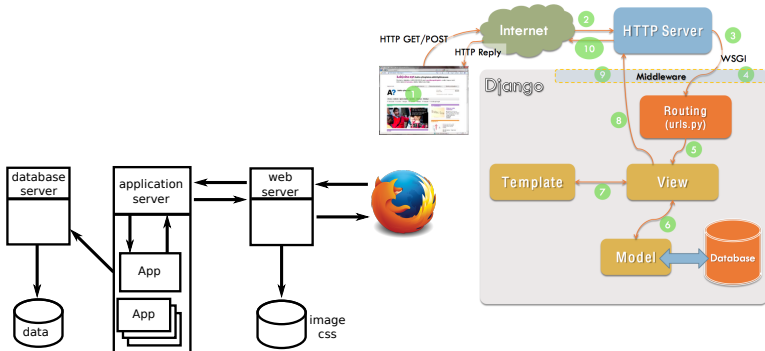


```
polo = new car();  
polo.drive();  
polo.setSpeed(10);  
fuel = polo.getFuel();  
print(fuel);
```

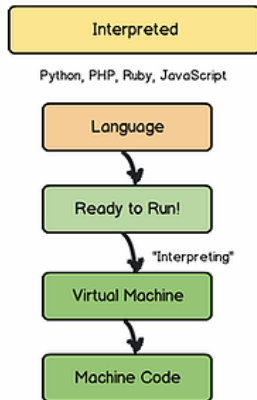
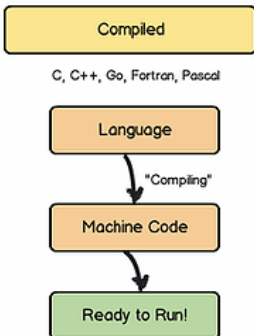

poziomy języków programowania



software framework



kompilacja vs interpretacja



IDE

The screenshot displays an IDE interface with the following components:

- Top Menu:** File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help.
- Project Explorer:** Shows a project structure for 'krasus' with files like 'animation', 'importGeoms', 'exampleLibGeom.json', 'CFASD.DTO.py', and 'Pedestrian.py'.
- Code Editor:** Displays the code for the `CFASD.DTO` class.


```

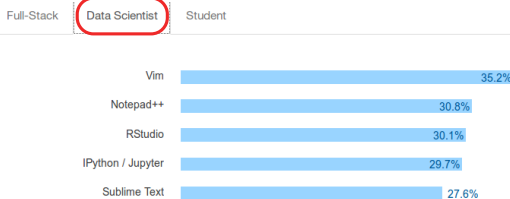
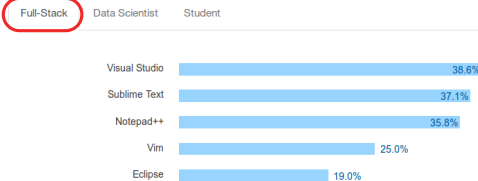
class CFASD.DTO:
    def __init__(self):
        self.path_to_cfast_dir = "/home/krasus/owncloud/projects/radon"
        self.project_name = "radon"
        self.path_to_json = "importGeoms/exampleLibGeom.json"
        self.sql_connection = sqlite3.connect(":memory:")
        self.c_const = 8
        self.json_object = None

    def __getitem__(self, item):
        return self.__dict__[item]

    def __setitem__(self, key, value):
        self.__dict__[key] = value

    @staticmethod
    def __process_csv(file):
        with open(file, 'r') as csvfile:
            reader = csv.reader(csvfile, delimiter=',')
            headers = next(reader)
            next(reader, None)
            headers1 = next(reader)
            next(reader, None)
      
```
- Debugger:** Shows the execution state with the following variables:
 - `env`: `<EvacEnv>` `<EvacEnv.EvacEnv object at 0x7f16003d2e80>`
 - `evacuees`: `<Evacuees>` `<Evacuees.Evacuees object at 0x7f16003d29b0>`
 - `goal`: `(list <class 'list'> [(80, 20), (100, 10), (140, 40)])`
 - `l`: `(int) 0`
 - `obst1`: `(list <class 'list'> [(0, 1), (1036, 1), (1036, 39), (1037, 39), (1037, 1), (1548, 1), (1548, 400), (1510, 400), (1510, 401), (1548, 401), (1549, 0), (1548, ...])`
 - `obst2`: `(list <class 'list'> [(80, 1, 0, 1), (103, 6, 0, 1), (103, 6, 3, 9), (103, 7, 3, 9), (103, 7, 0, 1), (154, 8, 0, 1), (154, 8, 40, 0), (151, 0, 40, 0), (151, 0, 40, 1), (154, 8, 40, 1), (154, 9, 40, 1), ...])`
 - `p1`, `p2`, `p3`: `<Pedestrian>` `<Pedestrian.Pedestrian object at 0x7f16180e1f98>`, `<Pedestrian.Pedestrian object at 0x7f16180e66a0>`, `<Pedestrian.Pedestrian object at 0x7f16180e66d8>`

Zestawienie środowisk



Zestawienie środowisk

Full-Stack

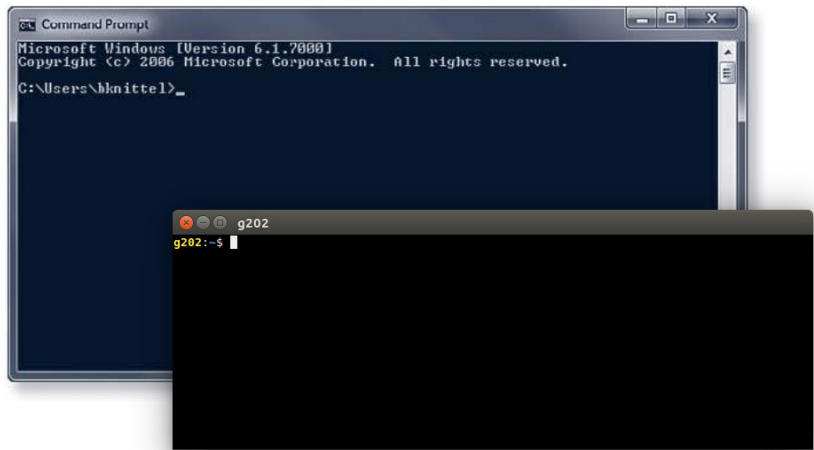
Data Scientist

Student



Podział języków komputerowych

Język komend



Kod maszynowy

ADD R1 R2 1 000 01 10

```

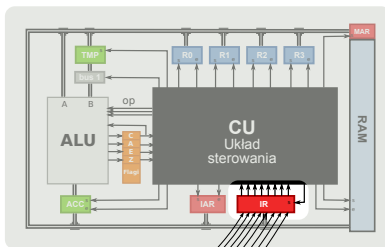
A 002000 C2 30 REP #930
A 002002 18 CLC
A 002003 F8 SED
A 002004 A9 34 12 LDA #91234
A 002007 69 21 43 ADC #94321
A 00200A 8F 03 7F 01 STA $017F03
A 00200E D8 CLD
A 00200F E2 30 SEP #930
A 002011 00 BRK
A 2012

r
PB PC NVMxDIZC .A .X .Y SP DP DB
; 00 E012 00110000 0000 0000 0002 CFFF 0000 00
g 2000

BREAK

PB PC NVMxDIZC .A .X .Y SP DP DB
; 00 2013 00110000 5555 0000 0002 CFFF 0000 00
m 7F03 7F03
>007F03 55 55 00 00 00 00 00 00 00 00 00 00 00 00 00 00:

```



1 0 0 0 0 1 1 0

Język znaczników

```

<h2>Wykład</h2>

Prezentacje:
<ol>
<li><a href=wyklad_inf/w1.pdf>Wprowadzenie </a>
<li><a href=wyklad_inf/w2.pdf>System binarny</a>
<li><a href=wyklad_inf/w3.pdf>Układy cyfrowe</a>
<li><a href=wyklad_inf/w4.pdf>Mikroprocesor</a>
<li><a href=wyklad_inf/w5.pdf>Komputer PC</a>
<li><a href=https://www.youtube.com/playlist?list=PLcR1x6ft
<li><a href=https://biedronek.inf.sgsp.edu.pl/biedronek/omr
<li><a href=wyklad_inf/w7.pdf>Bazy danych</a>
<li><a href=wyklad_inf/w8.pdf>Pre-processing</a>
<li><a href=wyklad_inf/w9.pdf>Programowanie</a>
<li><a href=wyklad_inf/w10.pdf>Grafika</a>
</ol>

<a href=https://godzinek.inf.sgsp.edu.pl/biedronek/omr/inde
<br><br>

<h2>Laboratorium</H2>
<a name=zasady></a>
<h3>Regulamin laboratorium</h3>
<ol>
<li>Wszystkie zajęcia na godz. 08:00 rozpoczynają się o godz. <b>08:15</b>.
<li>Student może przyjść z inną grupą o ile są wolne miejsca.
<li>Wzrost całego semestru może mieć nieuregulowane jedno spotkanie

```

```

<?xml version="1.0"
encoding="iso-8859-1" ?>
<languages>
  <language id="fr">
    <name lang="fr">Français</name>
    <name lang="en">French</name>
    <name lang="es">Frances</name>
    <name lang="de">Französisch</name>
    <name lang="eo">Franca</name>
  </language>

```

Język arkuszy stylów

```

1  body {
2      font-family: verdana;
3  }
4  }
5  hr {
6      margin: 30px 0px 30px 0px;
7  }
8  }
9  h2 {
10     padding: 20px 0px 20px 0px;
11     font-family: verdana;
12     font-size:20px;
13 }
14 }
15 h3 {
16     padding: 20px 0px 20px 0px;
17     font-family: verdana;
18     font-size:16px;
19 }
20 }
21 h1 {
22     padding: 20px 0px 20px 0px;
23     font-family: verdana;
24     font-size:20px;
25 }

```

Most Downloaded Movies on BitTorrent, 2011

rank	movie	downloads	grosses
1	Fast Five	9,260,000	\$626,137,675
2	The Hangover II	8,840,000	\$581,464,305
3	Thor	8,330,000	\$449,326,618
4	Source Code	7,910,000	\$123,278,618
5	I Am Number Four	7,670,000	\$144,500,437
6	Sucker Punch	7,200,000	\$89,792,502
7	127 Hours	6,910,000	\$60,738,797
8	Rango	6,480,000	\$245,155,348
9	The King's Speech	6,250,000	\$414,211,549
10	Harry Potter and the Deathly Hallows Part 2	6,030,000	\$1,328,111,219

source:torrent.com

Most Downloaded Movies on BitTorrent, 2011

Rank	Movie	Downloads	Grosses
1	Fast Five	9,260,000	\$626,137,675
2	The Hangover II	8,840,000	\$581,464,305
3	Thor	8,330,000	\$449,326,618
4	Source Code	7,910,000	\$123,278,618
5	I Am Number Four	7,670,000	\$144,500,437
6	Sucker Punch	7,200,000	\$89,792,502
7	127 Hours	6,910,000	\$60,738,797
8	Rango	6,480,000	\$245,155,348
9	The King's Speech	6,250,000	\$414,211,549
10	Harry Potter and the Deathly Hallows Part 2	6,030,000	\$1,328,111,219

source:torrent.com

Język konfiguracji

```

1  VERSN      , 7      , zup
2  TIMES     , 7200   , -120 , 10 , 30
3  EAMB      , 287.8  , 101300 , 0
4  TAMB      , 293.15 , 101300 , 0 , 50
5  DTCHECK   , 1.E-9  , 100
6
7
8  !!MATL, name, param1, param2, param3, param4, param5
9  MATL, CON, 1.7, 840, 2500, 0.4, 0.9, concrete
10 MATL, GIPS, 0.3, 1000, 1000, 0.02, 0.85, gipsum
11 MATL, GLASS, 0.8, 840, 2500, 0.013, 0.9, glass
12 MATL, BLOCK, 0.3, 840, 800, 0.03, 0.85, floor
13
14 !!COMPA , name , width , depth , height , x , y , z , matlCeiling , matlFloor , matlWall
15 COMPA , COR_1_1 , 23.84 , 1.6 , 3.0 , 71.15 , 6.52 , 0.0 , BLOCK , BLOCK , BLOCK
16 COMPA , COR_1_2 , 29.91 , 1.6 , 3.0 , 41.24 , 6.52 , 0.0 , BLOCK , BLOCK , BLOCK
17 COMPA , COR_1_3 , 26.56 , 2.0 , 3.0 , 7.79 , 6.12 , 0.0 , BLOCK , BLOCK , BLOCK
18 COMPA , HAL_1_4 , 7.83 , 2.12 , 3.0 , 94.99 , 6.0 , 0.0 , BLOCK , BLOCK , BLOCK
19 COMPA , HAL_1_5 , 6.89 , 8.12 , 3.0 , 34.35 , 0.0 , 0.0 , BLOCK , BLOCK , BLOCK
20 COMPA , ROM_1_6 , 7.79 , 7.7 , 3.0 , 0.0 , 0.36 , 0.0 , BLOCK , BLOCK , BLOCK
21 COMPA , ROM_1_7 , 12.41 , 6.16 , 3.0 , 72.13 , 0.36 , 0.0 , BLOCK , BLOCK , BLOCK
22 COMPA , ROM_1_8 , 6.1 , 6.16 , 3.0 , 66.03 , 0.36 , 0.0 , BLOCK , BLOCK , BLOCK
23 COMPA , ROM_1_9 , 6.1 , 6.16 , 3.0 , 59.93 , 0.36 , 0.0 , BLOCK , BLOCK , BLOCK
24 COMPA , ROM_1_10 , 6.17 , 6.16 , 3.0 , 53.76 , 0.36 , 0.0 , BLOCK , BLOCK , BLOCK

```

```

CONFIG - Notatnik
Plik Edycja Format Wzrostk Pomoc
REM
REM BaseSegment
REM określa adres pierwszego segmentu w pamięci konwencjonalnej
REM który zostanie przydzielony przez system dla bloków stron EMM.
REM Musi to być wartość szesnastkowa (hex).
REM 0x1000 <= BaseSegment <= 0x4000. Wartość ta jest zaokrąglana
REM dół do granicy 16KB. wartością domyślną jest 0x4000
REM
REM RAM
REM określa, że system powinien przydzielać 64 KB przestrzeni a
REM z pamięci górnej (UMB) dla bloków stron EMM, a pozostawiać
REM resztę (jeśli jest dostępna) do użycia przez DOS do obsługi
REM poleceń LOADHIGH i DEVICEHIGH. System standardowo używa całej
REM dostępnej pamięci UMB dla bloków stron.
REM
REM Rozmiar EMM jest określony w pliku PIF (przypisanym do programu
REM lub _default.pif). Jeśli rozmiar określony w pliku PIF wynosi 0
REM EMM jest wyłączana i polecenie EMM jest ignorowane.
REM
REM dos-high, umb
REM device=%SystemRoot%\system32\hmem.sys
REM files=200

```

język zapytań

produkt

id	nazwa
1	płatki śniadaniowe
2	jogurt

sprzedawca

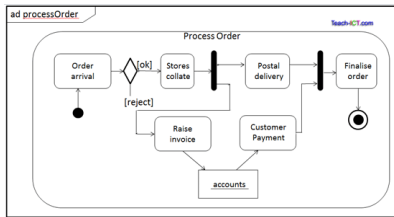
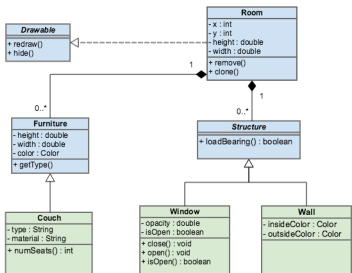
id	imię	nazwisko
1	Jan	Nowak
2	Janina	Kowalska

transakcje

id	produkt_id	sprzedawca_id	data	ilość	obrót	region
1	1	1	12.03.04	12	12 231 zł	centrum
2	2	1	13.04.09	123	1 zł	pomorze
3	1	2	12.03.05	27	12 231 zł	centrum
4	1	2	17.05.08	1	231 zł	małopolska
5	2	1	12.06.06	132	1 231 zł	pomorze
6	1	2	21.03.06	19	2 231 zł	pomorze
7	2	1	12.09.11	12	12 zł	małopolska
8	1	2	02.03.09	56	31 zł	wielkopolska
9	2	1	01.11.01	12	281 zł	pomorze
10	2	1	12.12.07	89	12 231 zł	centrum
11	2	2	27.09.09	198	52 231 zł	centrum

```
SELECT ilość FROM transakcje WHERE region='centrum';
```

język modelowania



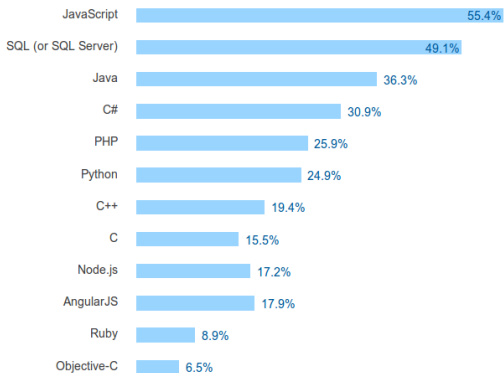
najpopularniejsze technologie

2016

2015

2014

2013



JavaScript

A yellow square containing the letters 'JS' in a bold, black, sans-serif font.

JavaScript jest to wysokopoziomowy, interpretowany język programowania opracowany przez Netscape, celem zapewnienia większej dynamiki przeglądania sieci.

JavaScript



W połączeniu z HTML oraz CSS, JavaScript jest jedną z trzech kluczowych technologii tworzenia stron WWW na świecie.

Większość stron www używa JS.

Wszystkie nowoczesne przeglądarki mają domyślnie wbudowaną obsługę JS.

JS jest językiem wieloparadygmatowym, wspiera programowanie obiektowe, imperatywne oraz funkcyjne.

JavaScript

JS

```
for (i = 0; i < 5; i++){  
    window.print(i);  
}
```

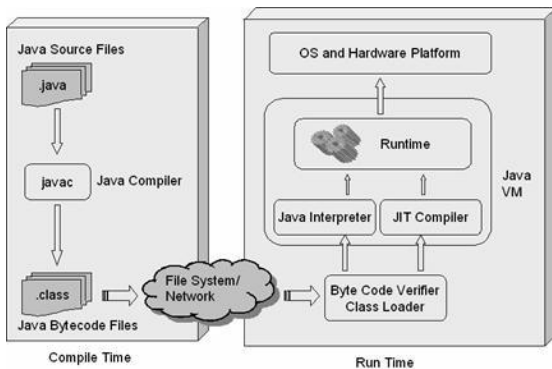
Java



Java jest językiem ogólnego przeznaczenia, bazującym na paradygmacie obiektowym i wspierającym wielowątkowość.

Opracowany przez Jamesa Goslinga z firmy Sun Microsystems.

Java



Java



```
class Demo {  
    public static void main(String[] args){  
        for( int i=1; i<11; i++){  
            System.out.println("Count is: " + i);  
        }  
    }  
}
```

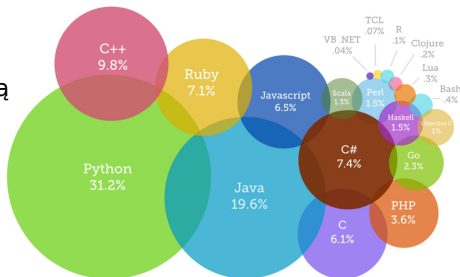
C++

C++ to jeden z najbardziej znanych języków programowania.



Język C++ jest potomkiem języka C. Twórcą języka C++ jest Bjarne Stroustrup, informatyk duńskiego pochodzenia.

Most Popular Coding Languages of 2015



Język C++ stał się inspiracją do powstania wielu innych języków programowania, np. Javy. Wciąż trwają prace nad jego udoskonalaniem.

@codeeval

<code>eval>

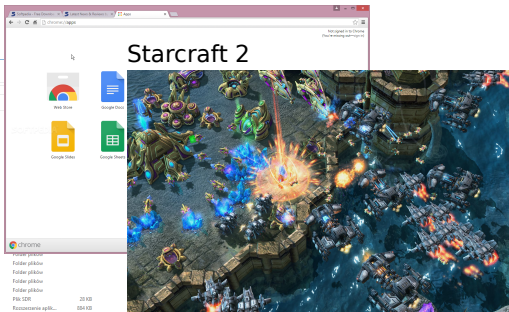
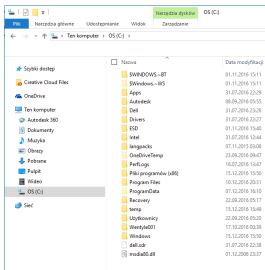
www.codeeval.com

C++

C++ ma wiele zastosowań, jest używany między innymi do programowania gier komputerowych, aplikacji desktopowych, baz danych, w miejscach gdzie ważne są wydajność i duże możliwości obliczeniowe

Google Chrome

Eksplorator Windows



C++

```
#include <iostream>
using namespace std;

void main () {
    for (int a = 5; a >= 0; a = a - 1 ) {
        count << "Wartość a: " << a << endl;
    }
}
```


Python



Python został stworzony w 1991 przez duńczyka Guido van Rossum.



Guido van Rossum

Python

Python jest językiem interpretowanym. Jest bardzo popularny w środowisku opensource.

Zastosowania

- tworzenie wtyczek (np. gimp, inkscape, libreoffice)
- obliczenia naukowe
- tworzenie interfejsów graficznych
- administracja systemem operacyjnym
- przetwarzanie tekstu

Python

Przykładowy kod

```
def odliczanie(x):  
    for i in range(x, 0, -1):  
        print(i)  
  
odliczanie(3)
```

PHP



Rasmus Lerdorf

PHP (PHP: Hypertext Preprocessor) został stworzony w 1994 przez Rasmus Lerdorfa.

PHP

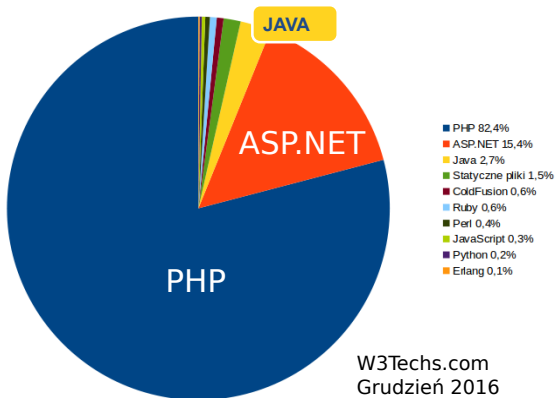
PHP jest skryptowym językiem interpretowanym działającym po stronie serwera do generowania stron WWW.

Jest językiem Open Source.

Wchodzi w skład LAMP (Linux, Apache, MySQL, and PHP), platformy serwerowej to tworzenia stron WWW.

PHP

Udział rynkowy serwerowych języków do WWW



PHP

Przykładowy kod

```
<?PHP
    function odliczanie(){
        for($i=3;$i>0;$i--){
            echo "$i\n";
        }
    }
    odliczanie(3);
?>
```

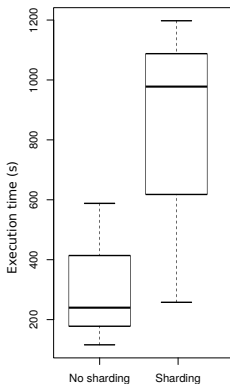
R

R jest środowiskiem oraz językiem programowania przeznaczonym do obliczeń statystycznych oraz wizualizacji danych i wyników.

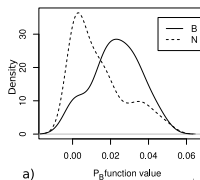
R jest implementacją języka S i został stworzony przez Roberta Gentlemana i Rossa Ihakę na uniwersytecie w Auckland.

R

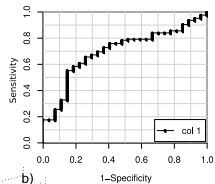
Wykorzystanie:
tworzenie oprogramowania statystycznego oraz analizy danych.



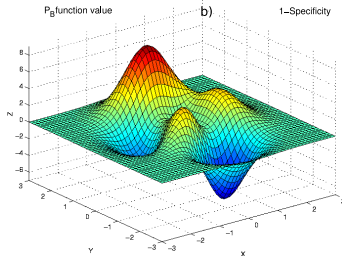
b) Stem frequency



a)



b)



R



```
for (i in 1:5){  
  print(i);  
}
```

Nauka programowania

