

Procedural programming

Jan Urbanek



Programovací paradigmatata

- imperativní vs. deklarativní
 - https://cs.wikipedia.org/wiki/Programovac%C3%AD_paradigma
- Metodiky vývoje softwaru
 - https://cs.wikipedia.org/wiki/Metodologie_v%C3%BDvoje_softwaru
- von Neumannova architektura
 - https://cs.wikipedia.org/wiki/Von_Neumannova_architektura

Procedurální paradigma

- "Imperativní programování (též procedurální programování) je jedno z programovacích paradigmat, neboli způsobů, jak jsou v programovacím jazyku formulována řešení problémů. Imperativní programování popisuje výpočet pomocí posloupnosti příkazů a určuje přesný postup (algoritmus), jak danou úlohu řešit. Program je sadou proměnných, jež v závislosti na vyhodnocení podmínek mění pomocí příkazů svůj stav. Základní metodou imperativního programování je procedurální programování, tyto termíny bývají proto často zaměňovány."
- Dělení
- naivní: např. BASIC
- nestrukturované: např. raný FORTRAN, COBOL
- strukturované: např. C, Pascal
 - https://cs.wikipedia.org/wiki/Imperativn%C3%AD_programov%C3%A1n%C3%AD

Ladění (debugging)

- Zásobník (stack)
 - [https://cs.wikipedia.org/wiki/Z%C3%A1sobn%C3%ADk_\(datov%C3%A1_struktura\)](https://cs.wikipedia.org/wiki/Z%C3%A1sobn%C3%ADk_(datov%C3%A1_struktura))
- Core dump
 - https://cs.wikipedia.org/wiki/Core_dump

Edsger Wybe Dijkstra (1930-2002)

- nizozemský matematik, fyzik a informatik
- tvůrce Dijkstrova algoritmu, otec strukturovaného programování, vynálezce semaforů
- semafor = nástroj pro synchronizaci více procesorů či procesů
- spolutvůrce prvního překladače Algolu
- formální verifikace algoritmů pomocí důkazu jejich matematické korektnosti
- autor příspěvku Go To Considered Harmful do Communications of the ACM (CACM), březen 1968, str. 147-148, název příspěvku je však dílem editora Niklause Wirtha
 - https://cs.wikipedia.org/wiki/Edsger_Dijkstra

Niklaus E. Wirth (*1934)

- švýcarský informatik
- vedoucí návrhář jazyků Algol, Pascal, Modula, Oberon
- Pascal jako didaktický programovací jazyk
- kniha Algoritmy + datové struktury = programy
 - https://cs.wikipedia.org/wiki/Niklaus_Wirth

GoTo umožňuje

- skočit kdykoli kamkoli
- tedy předčasně opustit cyklus
- vynořit se okamžitě z několikanásobně vnořené podmínky
- zpracování synchronních výjimek
- (selhání při alokaci paměti, otevření souboru, atp.)

Řídící struktury nahrazující GoTo

- if then else
- if then elseif elseif else
- též switch case + default
- while podmínka do/do while
 - + break
- for cyklus
- procedury, funkce

FORTRAN

- FORmula TRANslator, první jazyk
- *1954 v IBM
- 20x úspornější oproti assembleru
- mainframy
 - <https://cs.wikipedia.org/wiki/Mainframe>
- superpočítače, vědecké výpočty, počasí, simulace, fyzika...
- ANSI standardy Fortran 66, Fortran 77...

Perl

- *1987, autor Larry Wall
- interpretovaný skriptovací jazyk, dnes multiparadigmatický
- náhrada awk, sedu a bashe
- "There's more than one way to do it"
- vysoký jazyk (x C)
- výhody: regexy, práce s typy, hashe, komunita

Primárně strukturované jazyky

kompilované

- Fortran
- Algol
- Cobol
- PL/I
- Pascal
- C

interpretované

- BASIC
- Perl

Python procedural version

```
def procedure(input):  
    output = input + 1  
    return output  
  
procedure(2)
```

Fortran (compiler)

```
PROGRAM MAIN

INTEGER N, X

EXTERNAL SUB1

COMMON /GLOBALS/ N

X = 0

PRINT *, 'Enter number of repeats'

READ (*,*) N

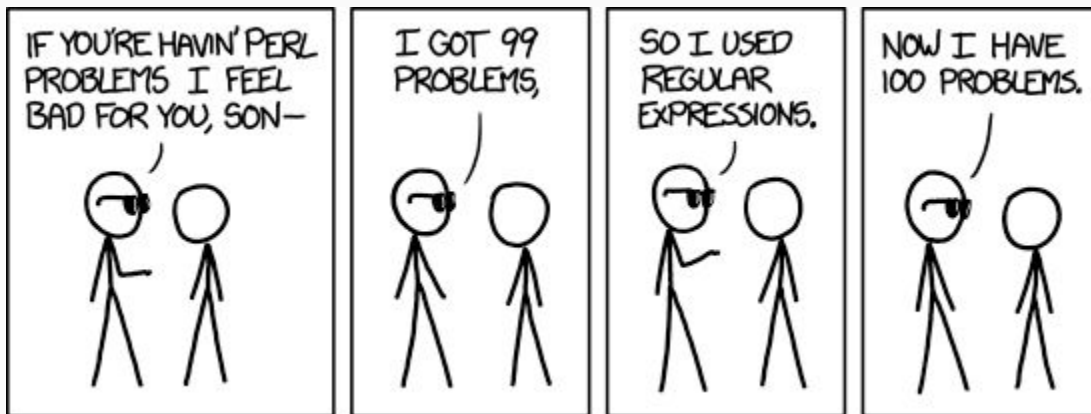
CALL SUB1(X,SUB1)

END

SUBROUTINE SUB1(X,DUMSUB)
```

Perl (compiler)

```
my $x = "foo";  
  
my $some_condition = 1;  
  
if ($some_condition) {  
    my $y = "bar";  
  
    print $x;           # prints "foo"  
  
    print $y;           # prints "bar"  
  
}  
  
print $x;               # prints "foo"  
  
print $y;               # prints nothing; $y has fallen out of scope
```



C (compiler)

```
#include <stdio.h>

int main() {

    int num;

    printf("Enter an integer: ");

    scanf("%d", &num);

    // True if num is perfectly divisible by 2
    if(num % 2 == 0)

        printf("%d is even.", num);

    else

        printf("%d is odd.", num);
```


PHP (compiler)

```
<?php
// PHP code to check whether the number
// is Even or Odd in Normal way

function check($number){

    if($number % 2 == 0){

        echo "Even";

    }

    else{

        echo "Odd";

    }

}
```

AWK (compiler)

```
awk 'BEGIN { print "Hello, world" }'
```

Brainfuck (compiler)

```
+++++++ [>+++++>+++++++>++++><<<<-]>+.> .+++++ . .+++.>+. <<+++++++> .> .+++ .----- .----- .>+.> .
```

Netlogo

```
to go

  if all? turtles [xcor >= food-x]

    [ stop ]

  ask leaders                                ;; the leader ant wiggles and moves

    [ wiggle leader-wiggle-angle

      correct-path

      if (xcor > (food-x - 5 ))              ;; leader heads straight for food, if it is close

        [ facexy food-x food-y ]

      if xcor < food-x                       ;; do nothing if you're at or past the food

        [ fd 0.5 ] ]

  ask followers
```