

PROGRAMOWANIE

- jednoznaczne formułowanie zadań i problemów
- określenie metod rozwiązywania
- dyskusja warunków rozwiązywania
- tworzenie sieci działań
- kodowanie algorytmu w wybranym języku
- testowanie
- opracowanie dokumentacji

PROGRAMOWANIE STRUKTURALNE:

- prog. hierarchiczne \rightarrow góra-dół
 - podział problemu na podproblemy tak długo jak to możliwe

PODSTAWOWE STRUKTURY:

- ① Instrukcja jako ciąg instrukcji prostych ($I_1, I_2 \dots$)
- ② Instrukcja warunkowa w postaci „jeżeli W , wykonaj I_2 ”
- ②b Instrukcja warunkowa o postaci „jeżeli W wykonaj I_1 ,
jeżeli nie - wykonaj I_2 ”
- ③ Instrukcja iteracyjna o postaci „powtarzaj I dopóki jest spełnione W ”
- ④ Instrukcja iteracyjna o postaci „powtarzaj I aż do spełnienia W ”

SCHEMAT POSTĘPOWANIA - PODSUMOWANIE:

- ① Wstępne sformułowanie problemu, budowa początkowego algorytmu jako liniowej sekwencji instrukcji
- ② Rozłożenie ww. instrukcji na struktury instrukcji warunkowych i iteracyjnych

- ③ Kontynuacja ② aż do uzyskania instrukcji elementarnych w danym języku programowania

ETAPY OPRACOWYWANIA PROGRAMU

- ① Analiza problemu, sformułowanie algorytmu
- ② Zakodowanie programu w języku programowania
- ③ Wprowadzenie tekstu programu źródłowego do komputera
- ④ Tłumaczenie programu
- ⑤ Poprawianie błędów formalnych (składniowych) wykrytych w kroku ④ i ponowne tłumaczenie
- ⑥ Uruchomienie, poprawienie błędów logicznych
- ⑦ Opracowanie dokumentacji

CECHY DOBREGO PROGRAMU:

- ① poprawność
- ② przystosowalność
- ③ odporność
- ④ stabilność
- ⑤ optymalność

ŹRÓDŁA BŁĘDÓW W REALIZACJI NA KOMPUTERZE ALGORYTMÓW DOKŁADNYCH:

- ① Stosowanie metod numerycznych, zakładających:
 - a) wykorzystanie metod przybliżonych, np. zastąpienie pochodnej jej ilorazem różnicowym
 - b) obcinanie procesów nieskończonych do skończonych, np. zastąpienie całkowania obliczaniem skończonych sum
- ② Operowanie przez komputer na liczbach mających skończone rozwinięcie dwójkowe

Język C:

HISTORIA

- ① C → interpretacja B (1969-73)
- ② implementacja kernela Unix (1973)
- ③ 1978 - dokumentacja C (Kernighan/Ritchie)

ZNAKI W JĘZYKU C:

- A-Z, a-z, 0-9
- klawisze „pod klaw F#”
- spacja
- znaki matematyczne i nawiasy

PRZYKŁADOWY PROGRAM:

```
/* moj pierwszy program w C */      komentarz
#include <stdio.h>                    dyrektywa preprocesora
int main <void>                       standardowy początek programu w C
{
    puts "Halo Ziemianie!");          początek listy instrukcji
    return 0;                         } instrukcje
}                                     koniec listy instrukcji
```

średnik = koniec instrukcji

małe litery ≠ duże litery

OGÓLNA POSTAĆ PROGRAMU:

dyrektywy preprocesora
deklaracje globalne

```
main()
{
```

zmienne lokalne funkcji main()

po nazwie funkcji zawsze nawiasy okrągłe

```

{
    zmienne lokalne funkcji main()
    instrukcje funkcji main
}
f1()
{
    zmienne lokalne f1()
    instrukcje f1()
}
...

```

po nazwie funkcji zawsze nawiasy okrągłe
 {} - nawiasy ograniczające treść funkcji

FUNKCJA - NAJWAŻNIEJSZA CZĘŚĆ PROGRAMU:

- w programie musi być min. 1 funkcja
- musi być funkcja main
- main → pierwsza funkcja po odpaleniu programu

SŁOWO KLUCZOWE:

- słowo o specjalnym znaczeniu.
- niedopuszczalne użycie w sposób inny niż zgodnie z regułami języka
- słowa kluczowe w C → małe litery

SI IK ;

DYREKTYWY PREPROCESORA:

- ① Dyrektywa - polecenie dla preprocesora → # nazwa . parametry
- ② Preprocesor - program interpretujący, którego zadaniem jest przetwarzanie tekstu wejściowego na wyjściowy w sposób określony przez programistę

KOMENTARZE

- ① Bez udziału w programie
- ② Zastosowanie: uwagi, opisy, etc.
- ③ Nie można zagnieżdżać jednego w drugim
- ④ Nie może być w słowie kluczowym

ZMIENNE, TYPY DANYCH, STAŁE

① IDENTYFIKATOR (NAZWA)

- ciąg nast. po sobie znaków, pierwszy znak musi być literą, lub podkreślnikiem, każdy następny - litera, cyfra, podkreślnik
- nazwa \neq słowo klucz; nazwa \neq st. zarez. komputera
- unikalność nazw jest konieczna
- w C zawsze trzeba zadeklarować nazwę

TYPY DANYCH:

① Całkowite

- int (2B/4B)
- short int (2B)
- long int (4B) / long long int (8B)

② Zmiennoprzecinkowe

- float (4 bajty) (-37; 38) \leftarrow wykładniki; 6 cyfr znaczących
- double (8 bajtów); wykładnik -307; 308; 15 c
- long double

③ Znakowe

- char (1B) - znak lub liczba całkowita

④ Logiczne (prawda/fałsz)

- _Bool \rightarrow true / false (1 bit)

DEKLARACJE ZMIENNYCH:

- int a; float mynumber;
- unsigned short NumberOfSons;
- signed int \Leftrightarrow signed
- unsigned int \Leftrightarrow unsigned
- int a, b, c \Leftrightarrow int a; int b; int c;