

WSTĘP DO PROGRAMOWANIA

Rozdział Pierwszy

Pytania sprawdzające

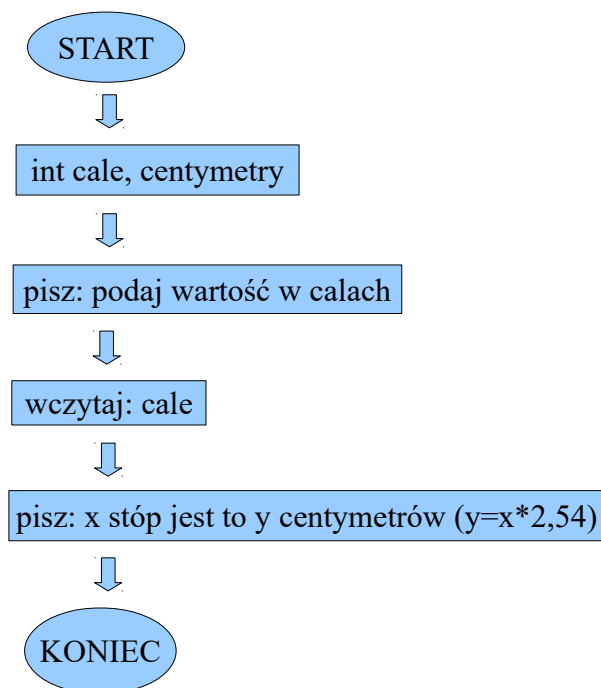
1. Przenośność w kontekście programowania oznacza, że programy napisane w jednym systemie mogą być uruchomione w innych systemach. Może wymagać to niewielkich modyfikacji programu. Język C jest przykładem języka przenośnego. Kompilatory języka C są dostępne dla wielu platform systemowych, co sprawia, że programy w tym języku są łatwe do przenoszenia.
2. Plik kodu źródłowego (=plik źródłowy) – jest zazwyczaj plikiem tekstowym o rozszerzeniu ".c". Jest to pierwszy plik na drodze do powstania programu. W tym pliku możemy wpisywać za pomocą edytora tekstu kolejne polecenia.
Plik kodu obiektowego – powstaje na skutek kompilacji kodu źródłowego. Kompilator tłumaczy kod źródłowy na kod przejściowy, kod maszynowy przechowywany w pliku kodu obiektowego. Nie jest on gotowy do uruchomienia. Ma rozszerzenie ".obj". Linker (program łączący), łączy kod przejściowy z pozostałym kodem i tak powstaje plik wykonywalny. Kod obiektowy, kod startowy, kod/-y procedur bibliotekowych są scalane przez linker do pliku wykonywalnego o rozszerzeniu ".exe".
3. Siedem głównych etapów programowania:
 - 1) określenie cel-u/-ów programu,
 - 2) Projektowanie
 - 3) Pisanie kodu
 - 4) Kompilacja
 - 5) Uruchomienie programu
 - 6) Testowanie i usuwanie błędów
 - 7) "Pielęgnowanie" i modyfikacja programu
4. Kompilator służy do tłumaczenia kodu źródłowego w języku C na język maszynowy i zapisania go w pliku obiektowym. Często kompilator wywołuje również drugi program, linker (program łączący).
5. Linker (program łączący) służy do połączenia kodu obiektowego z kodem startowym (właściwym dla danego systemu operacyjnego) oraz z kod-em/-ami bibliotek operacyjnych w jeden plik wykonywalny.

Ćwiczenie

Krok pierwszy: określenie celów programu

Celem jest przeliczenie cali na centymetry (użytkownik wpisuje wartość w calach, program przelicza na cale)

Krok drugi: projektowanie programu



Rozdział Drugi

Pytania sprawdzające

1. Funkcje są podstawowymi modułami, z jakich składa się program w języku C. Każdy program pisany w języku C musi zawierać co najmniej jedną funkcję, funkcję główną "main". Funkcja składa się z dwóch części: nałówka i treści.

2. Błąd składniowy – wynika z niezastosowania się do zasad języka C. Polega na zastosowaniu prawidłowych symboli języka w niewłaściwych miejscach.

przykład w języku polskim: "Składniowy błędy bardziej częsta jesteśmy."

przykład w języku C:

`include <stdio.h>` (brak znaku # na początku wiersza)

`int main(void)`

`[` (nawias kwadratowy zamiast klamry "{")

`int p, r, s;`

`p=2*r;` (dwukropek zamiast średnika)

`r=3*s;`

`printf{"%d jest równe %d\n", p, r}` (klamra "{" zamiast nawiasu "()", pojedynczy cudzysłów zamiast podwójnego)

`return 0;`

`]` (nawias kwadratowy zamiast klamry "}")

3. Błąd semantyczny – błąd w znaczeniu

przykład w języku polskim: "Przebiegłe błędy semantyczne chytrze łypią na czarny monitor."

przykład w języku C:

```
#include <stdio.h>
int main(void)
{
int p, r;
p=2*r; (według funkcji p ma równać się połowie r, więc r=2*p)

printf("%d jest równe połowie %d\n", p, r);
return 0;
}
```

4. Program po korekcie błędów

```
#include <stdio.h>
int main(void) /* program wyświetla liczbę tygodni w roku */
{
int s;
s=56;
printf("W roku jest %d tygodni.\n", s);
return 0;
}
```

*na czerwono poprawione elementy

5.

- a. Program wyświetli zdanie: "Wlaził kotek na plotek i mruga.", kursor pozostaje na końcu wiersza
- b. Program wyświetli zdanie: "Ladna to piosenka, niedługa.", kursor zostaje przeniesiony na początek następnego wiersza
- c. Program wyświetli dwa wiersze: pierwszy wiersz: "Kobieto! puchu marny!"
drugi wiersz: "Ty wietrzna istoto!", kursor pozostaje na końcu wiersza
- d. Program wyświetli dwa wiersze: pierwszy wiersz: "Co?"
drugi wiersz: "Nie ma/nobiadu?", kursor zostaje przeniesiony na początek następnego wiersza
- e. Program wyświetli: "2 + 2 = 4", kursor pozostaje na końcu wiersza

6. **Słowa kluczowe** języka C (słownictwo języka C) – mają specjalne znaczenie
main, **int**, function, **char**, =

7.

```
#include <stdio.h>
int main(void)
{
int slowa, wiersze;
slowa=3020;
wiersze=350;
printf("Było %d słów i %d wierszy.", slowa, wiersze);
return 0;
}
```

}

8. Program wyświetla: "5 5". Po wykonaniu wiersza 7 powinien wyświetlić: "2 5".
Po wykonaniu wiersza 8: "5 5". Po wykonaniu wiersza 8 powinien wyświetlić: "5 5"