

## Języki programowania 1

---

### Pierwsze programy w języku C

Zakres:

- dyrektywa #include,
  - pliki nagłówkowe,
  - komentarze,
  - funkcja main(),
  - instrukcja return,
  - funkcja exit(),
  - znak przejścia do nowego wiersza,
  - typ danych void,
  - translacja programu w języku C na plik wykonywalny,
  - diagnostyka i korekta błędów.
- 

### Prosty program w języku C

```
1: /*to jest pierwszy programik*/
2: #include <stdio.h>
3: main()
4: {
5: printf("Hallo....To jest moj pierwszy program.\n");
6: return 0;
7: }
```

---

Na ww. listingu w linijce nr 1 zawarty jest komentarz. Rozpoczyna się on kombinacją dwóch znaków i tak samo też kończy kombinacją dwóch znaków:

`/* rozpoczęcie komentarza (//),`

`*/ zakończenie komentarza.`

Wszystko co znajdzie się między parą tych dwóch znaków zostanie zignorowane przez nasz kompilator. Dodając komentarze zwiększa się tylko objętość kodu tekstowego, a nie ma to żadnego wpływu na program, czy też długość kodu wykonywalnego.

---

## **Dyrektywa #include**

W linii nr 2 występuje:

`#include <stdio.h>`

Wiersz ten rozpoczyna się od # (hasz), a po nim słowo include (dołącz). W języku C #include jest to tzw. dyrektywa preprocesora. Innymi słowy jest to polecenie dla preprocesora, aby w danym miejscu dołączyć kod zawarty w pliku (np. stdio.h standard input-output- standard wejścia-wyjścia).

---

## **Pliki nagłówkowe**

Pliki dodawane do programów przy pomocy dyrektywy #include, nazywane są plikami nagłówkowymi (header files), ponieważ takie dołączeni odbywa się zawsze na początku każdego przez nas pisanego programu. Litera 'h' jest jako rozszerzenie do nazw tych plików (od słowa header – nagłówek).

Przykładowe:

stdlib.h – standardowa biblioteka funkcji,  
string.h – operacje na łańcuchach tekstowych,  
math.h – funkcje matematyczne.

---

### **Nawiasy trójkątne (<>) lub cudzysłów (" ")**

Jeżeli dana nazwa pliku ujęta jest w nawiasy trójkątne np. <stdio.h>, to oznacza to, że preprocesor pliku tego będzie poszukiwał w innych zasobach dyskowych niż znajduje się sam program. Jeżeli mamy przykładowo własne pliki, które wchodzi np. w skład naszego rozbudowanego projektu i pliki, które chcemy dołączać do programu głównego wpisać możemy bezpośrednio "dane.h". Wówczas preprocesor najpierw 'zajrzy' do danego katalogu roboczego.

---

### **Funkcja main()**

Jest to specjalna funkcja w języku C. Każdy program musi ją posiadać (dokładnie jedną). Można ją wstawiać w dowolne miejsce, ale komputer zawsze rozpocznie wykonywanie programu od funkcji main(). Ww. listing zawiera dokładnie funkcję main(). Pozostałe linie kodu tegoż programu są tzw. ciałem funkcji. Wykonanie każdego programu nie tylko zaczyna się, ale również po wykonaniu wszystkich operacji kończy się.

---

### **Znak przejścia do nowego wiersza \n.**

W wywołaniu funkcji printf() użyto jednego szczegółu, a mianowicie \n. Jest to znak przejście do nowej linii w programie. Znak ten powoduje, że komputer wykonujący dany program przeniesie kursor do nowego wiersza.

W systemie UNIX będzie to „/r/n” powrót na początek wiersza i przejście do nowego.

---

## **Instrukcja return**

W języku C funkcje zazwyczaj zwracają jakiś wymik.

Funkcja main() także zwraca wartość. Domyślnie jest to liczba typu integer. Jeżeli funkcja main() zwraca 0 oznacza to, że program wykonał się prawidłowo i do końca. Wartość różna od zera zwrócona przez funkcję main() oznacza błąd (błędy) wykonywania programu.

---

## **Funkcja exit()**

Jedną z funkcji bibliotecznych w C jest funkcja exit() (wyjście), która może być używana do zakończenia programu lub też zakończenia wszelkich procesów wykonywanych w programie. Funkcja ta jest zdefiniowana w pliku nagłówkowym stdlib.h. Zatem chcąc ją wykorzystać, należy w nagłówku programu dołączyć tenże plik.

W przeciwieństwie do funkcji main() funkcja exit() nie zwraca żadnego wyniku, ale argument przekazywany tej funkcji wskazuje, czy program zakończył się normalnie.

Wykorzystać funkcję exit() zamiast return w przykładzie (pierwszy program).

---

## **Kompilacja i konsolidacja programów**

Do utworzenia pliku wykonywalnego potrzebne są co najmniej trzy etapy:

- napisanie programu w C (kod źródłowy),
  - dołączenie przez preprocesor plików nagłówkowych,
  - kompilator C kompiluje program tworząc na dysku plik binarny, nazywany plikiem skompilowanym – object file (w Unix’ie takie pliki kończą się na .o, w DOS’ie .obj),
  - następnie plik zero-jedynkowy wymaga konsolidacji, czyli dołączania; pracę tę wykonuje tzw. linker, który wchodzi w skład fabrycznego kompilatora; konsolidator dołącza do skompilowanego kodu programu .obj standardowe biblioteki oraz ewentualne biblioteki użytkownika, tworząc plik wykonywalny .exe.
- 

## **Informacja o błędach i ostrzeżeniach**

1. error message
2. warning message