

Zajęcia 4 (2016)

Cele zajęć:

- zapisywanie powtarzalnych procedur w postaci funkcji
- ćwiczenie w pisaniu programów obliczeniowych
- ćwiczenie w układaniu algorytmów działania

Programy do napisania:

- 1) Zmodyfikować program liczący pierwiastki równania kwadratowego (który był podawany na wykładzie), tak aby zapisać podnoszenie do kwadratu oraz obliczanie delty w postaci funkcji.
- 2) Zmodyfikować program, który rysował na ekranie trójkąt z gwiazdek, tak aby rysowanie trójkąta było zapisane w postaci funkcji, która jako parametr dostaje wielkość podstawy trójkąta. Program powinien zapytać użytkownika jak duży trójkąt sobie życzy, no i oczywiście powinien używać w swoim ciele zdefiniowanej wcześniej funkcji.
- 3) Do programu z punktu 2 dołączyć jeszcze jedną funkcję, tym razem rysującą romb, czyli coś takiego:

```
  *
 ***
*****
 ***
  *
```

Oczywiście program powinien teraz dawać użytkownikowi możliwość wybrania co ma rysować. W tworzeniu funkcji rysującej romb można posłużyć się funkcją rysującą trójkąt.

- 4) Napisać program wypisujący wszystkie dzielniki podanej liczby. W programie napisać i użyć funkcji sprawdzającej czy dana liczba jest dzielnikiem podanej liczby czy nie. Proszę spróbować samemu wymyślić algorytm działania. Generalnie można zacząć od sprawdzania wszystkich potencjalnie możliwych dzielników. W tym celu może się przydać operator ' % ' w wyniku dający resztę z dzielenia, czyli np. $6\%3$ daje 0 a $6\%4$ daje 2. Oczywiście sprawdzana liczba będzie dzielnikiem danej liczby jeśli reszta z dzielenia będzie równa 0.

Przypomnienie.

Po co pisać własne funkcje w programie? Fragmenty kodu, które mogą stanowić pewną całość, a które mogą być używane wiele razy w programie można i należy zapisywać w postaci osobnych funkcji. Często nawet fragment programu, który nie będzie wiele razy powtarzany, dobrze jest zapisać w postaci odrębnej funkcji dla czytelności kodu programu głównego. Używanie funkcji wymusza pewną formalizację zapisu, funkcja bowiem musi mieć zdefiniowaną liczbę i typ parametrów, oraz zdefiniowaną wartość zwracaną. Funkcja może korzystać z własnych lokalnych zmiennych, które są niezależne od zmiennych zdefiniowanych w programie głównym.

Deklaracja i zarazem definicja funkcji może odbywać się następująco:

Przed funkcją „main” należy zapisać:

```
typ_zwracanej_wielkości nazwa_funkcji(typ_parametru
nazwa_parametru)
{
//tutaj ma być zawarte „ciało” funkcji
}
```

//parametrów może być w zasadzie dowolna ilość, a oddzielone muszą być przecinkami, na przykład:

```
float kwadrat(float podstawa)
{
//tutaj można deklarować zmienne wewnętrzne
//i wyliczyć to co trzeba
float wynik; // to jest zmienna wewnętrzna
wynik=podstawa*podstawa; //obliczenia
return(wynik) // instrukcja zwrócenia wartości
}
```

Następnie, już wewnątrz funkcji 'main' można użyć własnej funkcji np. tak:

```
delta=kwadrat(b)-4*a*c;
```

W przypadku szczególnym funkcja nie musi pobierać żadnego parametru i żadnej wartości zwracać, a za to może np. coś rysować na ekranie na przykład funkcja rysująca poziomą kreskę na ekranie (w konsoli):

```
void rysuj_linie(void)
{
int i;
for(i=0;i<12;i++)
{
    printf("-");
}
printf("\n");

return;
}
```