

Anna Jasielec

Podstawy programowania, grupa nr 4

Sprawozdanie z laboratorium nr 8

Cel laboratorium: Opanowanie podstaw tworzenia i wykorzystania funkcji w C

Przebieg zajęć:

Utworzyłam katalog roboczy lab_8 i podkatalog simple. Skopiowałam ze strony przedmiotu plik prosta_funkcja.c. Zanalizowałam kod i go skompilowałam. Zmodyfikowałam kod zgodnie z poleceniami w kodzie.

```
// 1. definicja zmiennej test np. typu int z inicjowaniem
int test = 5;

// 2. drukowanie wartości zmiennej test
printf("przed wywołaniem prostej funkcji: %d \n", test);
```

```
// 3. definicja zmiennej test np. typu int z inicjowaniem
test = 10;

// 4. drukowanie wartości zmiennej test
printf("wewnątrz prostej funkcji: zmienna lokalna %d\n", test);
```

```
// 5. zmiana definicji - przyjmowanie argumentu
void prosta_funkcja( int test )
{
```

```
// 6. (po wykomentowaniu wywołania powyżej) wywołanie funkcji z argumentem
prosta_funkcja(test);
```

Po wykonaniu kroków 5 i 6, należało usunąć argument *void* z deklaracji żeby program zadziałał.

```
//11. drukowanie wartości argumentu
printf("wewnątrz prostej funkcji: argument przed modyfikacją: %d \n", test);

// 10. zmiana wartości przesłanego argumentu
test += 4;

// 11. drukowanie wartości argumentu
printf("wewnątrz prostej funkcji: argument po modyfikacji: %d \n", test);
```

Następnie utworzyłam podkatalog *pierwiastek* i skopiowałam do niego plik *test_pierwiastka.c*. Analiza, uruchomienie i modyfikacja zgodnie z krokami w kodzie.

```
// 1 i 2 zaprojektowanie prototypu funkcji obliczania pierwiastka i umieszczenie go
//      jako deklaracji funkcji
double pierwiastek();
void ObliczPierwiastek();
```

```
// 3. wywołanie zaprojektowanej funkcji obliczania pierwiastka
pierwiastek(liczba);
```

```
4. wydruk obliczonego pierwiastka
ewentualne dodatkowe wywołania funkcji obliczania pierwiastka, zgodnie z instrukcją
printf("pierwiastek liczby: obliczony %18.15lf, dokładny %18.15lf\n",
    pierwiastek(liczba), sqrt(liczba));
```

Stworzenie własnej funkcji obliczającej pierwiastek:

```
//MOJA FUNKCJA
// 6. nagłówek - początek definicji funkcji
double pierwiastek (double arg) {

    double pierwiastek = 1.0;
    double temp; int iteracje=0;
    // obliczenia pierwiastka za pomocą wzoru Herona
    do{
        iteracje++;
        temp = pierwiastek;
        pierwiastek = 0.5 * (temp + arg/temp);

    } while(fabs(pierwiastek*pierwiastek - arg)/arg > ACCURACY);

    return(pierwiastek);
}
```

Wnioski:

- Używanie funkcji w C sprawia, że program staje się czytelniejszy.
- Funkcje należy zadeklarować na początku pliku, zdefiniować możemy ją na końcu. Chcąc aby funkcja się wykonała przywołamy ją w funkcji main prostą komendą Funkcja().
- Funkcja może przyjmować argument, który musi mieć typ (void funkcja (int argument)).
- Gdy w wywołaniu funkcji wpisujemy argument to dla niego wykona się ta funkcja.
- Funkcja może zwracać liczbę, będzie wtedy typu int, double. Gdy jest typu void nie zwróci nic.
- Poznałam matematyczny wzór Herona, dzięki któremu oblicza się pierwiastek.