

Ćwiczenie nr 3

Temat: Definicje i wykorzystanie funkcji, parametry funkcji

Zagadnienia:

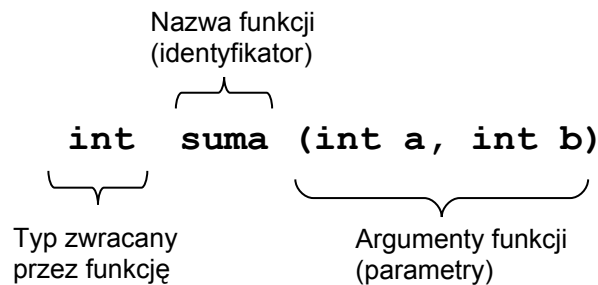
- Definicja funkcji – składnia podstawowa.
- Sposoby przekazania parametrów (argumentów) funkcji:
 - przez wartość,
 - przez referencje do zmiennej,
 - przez wskaźnik.

1. Definicja funkcji – składnia podstawowa.

Definicja funkcji składa się z nagłówka funkcji i ciała funkcji. Przykład - funkcja **suma** z dwoma parametrami typu **int** zwraca wartość typu **int**:

```
int suma(int a, int b)
{
    int wynik;
    wynik = a + b;
    return wynik;
}
```

Nagłówek funkcji jest interfejsem funkcji: definiuje nazwę funkcji, argumenty funkcji, typ danych zwracanych przez funkcję:



Ciało funkcji definiuje działanie funkcji i zwraca wartość funkcji (instrukcja **return wynik**). Zwrot wartości funkcji (instrukcja `return`) jest konieczny gdy zadeklarowany typ zwracany przez funkcję będzie inny niż **void**.

```
{
    int wynik;      /* zmienna lokalna */
    wynik = a + b;  /* obliczenia */
    return wynik;   /* zwracana wartosc */
}
```

2. Sposoby przekazania parametrów (argumentów) funkcji.

– przekazanie przez wartość

Przykład:

```
int dodaj(int a, int b)
{
    a = a + b;           /* zmiana wartości zmiennej a */
    return a;
}

int main()
{
    int i = 2, j = 3, k; /* nie powoduje zmiany */
    k = dodaj(i, j);     /* wartości zmiennej i */
}
```

– przekazanie przez wskaźnik do zmiennej

Przykład:

```
int pomniejsz(int *a)
{
    *a = *a - 2;         /* zmiana wartości zmiennej wskazywanej przez a */
    return *a;
}

int main()
{
    int e = 5, f;        /* powoduje zmianę */
    f = pomniejsz(&e);   /* wartości zmiennej e */
}
```

– przekazanie przez referencje do zmiennej (język C++)

Przykład:

```
int kwadrat(int &a)
{
    a = a * a;           /* zmiana wartości zmiennej a */
    return a;
}

int main()
{
    int c = 2, d;        /* powoduje zmianę */
    d = kwadrat(c);      /* wartości zmiennej c */
}
```

3. Przykłady definicji i wykorzystania funkcji.

Przykład 1. Program silnia.c. Kilka przydatnych funkcji.

```
#include <stdio.h>

/* funkcja etykieta bez parametrow, nie zwraca wartosci
   wykonuje ciag operacji i konczy dzialanie */
void etykieta()
{
    printf(" ----- \n\r");
    printf("   Program wykonuje obliczenia y=n!   \n\r");
    printf(" ----- \n\r");
}

/* funkcja silnia zwraca wartosc typu unsigned long (bez znaku)
   oblicza wartosc n!, parametr typu int */

unsigned long silnia(int i)
{
    unsigned long s; // wartosc silni
    int j;           // licznik petli
    for (s = 1, j = 2; j<=i; j++)
        s *= j;      // mnozenie s = s * j = wartosc n! = i!

    return s;        // wartosc zwracana przez funkcje
}

/* funkcja jestKoniec pyta uzytkownika czy zakonczyc program
   zwraca wartosc typu int: 1 gdy koniec, 0 gdy powtorka */

int jestKoniec()
{
    int key; // kod znaku z klawiatury
    printf("\n\n\r Zakonczyc program? (t/n)");
    do {
        key = getch(); // pobranie kodu znaku z klawiatury
    }
    while ((key!='t') && (key!='n')); // czy wcisnieto 'n' lub 't'
    if (key=='t')
        return 1; else // 1 gdy koniec
        return 0;      // 0 gdy powtorzyc program
}

int main()
{
    int n;
    unsigned long y;
    etykieta(); // wywołanie funkcji etykieta
    do {
        printf("\n\n\r Podaj wartosc liczby n=");
        scanf("%i",&n);
        y = silnia(n); // obliczenia silni - wywołanie funkcji silnia
        printf("\n\r Wartosc y = %2i! = %li ", n, y);
    }
    while (!jestKoniec()); // zakoncz program? - wyw. funkcji jestKoniec

    system("pause");
    return 0;
}
```

Przykład 2. Szkielet programu z prostym menu użytkownika.

```
#include <stdio.h>
#include <stdlib.h>
#include <dos.h>

void print_menu()
{
    printf("  Program demonstruje konstrukcje prostego menu  \n\r");
    printf("  '1' - Opcja 1 = funkcja1  \n\r");
    printf("  '2' - Opcja 2 = funkcja2  \n\r");
    printf("  '3' - Opcja 3 = funkcja3  \n\r");
    printf("  '0' - Opcja 4 = Wyjście  \n\r");
}

void funkcja1()
{
    printf(" Wybrano 1 opcje z menu - funkcja1  \n\n");
}

void funkcja2()
{
    printf(" Wybrano 2 opcje z menu - funkcja2  \n\n");
}

void funkcja3()
{
    printf(" Wybrano 3 opcje z menu - funkcja3  \n\n");
}

int main()
{
    int key;

    print_menu();
    do {

        key = getch();
        switch (key) {

            case '1' : funkcja1(); break;
            case '2' : funkcja2(); break;
            case '3' : funkcja3(); break;
            case '0' : printf(" Koniec programu \n\n");break;
            default  : printf(" Nieznane polecenie \n\n");
        } // koniec switch

    } while (key!='0');

    system("pause");
    return 0;
}
```

5. Zadania do wykonania na zajęciach lub w domu:

1. Napisać program wykonujący obliczenia prądu, napięcia, rezystancji na podstawie prawa Ohma. Program powinien zawierać:

- funkcję obliczania prądu na podstawie napięcia i rezystancji,
- funkcję obliczania napięcia na podstawie rezystancji i prądu,
- funkcję obliczania rezystancji na podstawie napięcia i prądu,
- program główny testujący powyższe funkcje z możliwością wyboru rodzaju obliczeń i powtórzenia obliczeń.

2. Napisać program wykonujący obliczenia wartości funkcji wielomianowych

Program powinien zawierać:

- funkcję obliczającą wartość wielomianu $y = a \cdot x^2 + b \cdot x + c$,
- funkcję obliczającą wartość wielomianu $y = a \cdot \sin^2 x + b \cdot \sin x + c$,
- funkcję obliczającą wartość wielomianu $y = a \cdot \ln^2 x + b \cdot \ln x + c$,
- program główny testujący powyższe funkcje z możliwością wyboru rodzaju obliczeń i powtórzenia obliczeń.

3. Napisać program wykonujący obliczenia funkcji trygonometrycznych

Program powinien zawierać:

- funkcję obliczającą wartość $y = \tan x$ (funkcja \tan nie jest dostępna w standardowych bibliotekach),
- funkcję obliczającą wartość $y = \cot x$ (funkcja \cot nie jest dostępna w standardowych bibl.),
- funkcję obliczającą wartość wielomianu $y = |1 - \sin^2 x|$,
- program główny testujący powyższe funkcje z możliwością wyboru rodzaju obliczeń i powtórzenia obliczeń.

4. Napisać program wykonujący obliczenia pól figur geometrycznych

Program powinien zawierać:

- funkcję obliczającą wartość pola prostokąta,
- funkcję obliczającą wartość pola koła,
- funkcję obliczającą wartość objętość kuli,
- funkcję obliczającą wartość pola trójkąta,
- funkcję obliczającą wartość objętość stożka,
- program główny testujący powyższe funkcje z możliwością wyboru rodzaju obliczeń i powtórzenia obliczeń.