

LABORATORIUM 4. PROSTE PROGRAMY STRUKTURALNE. FUNKCJE STANDARDOWE I FUNKCJE WŁASNE. ARGUMENTY FUNKCJI.

Cel laboratorium:

Zaznajomienie z realizacją algorytmów liniowych (sekwencyjnych), z ideą powtórnego wykorzystania kodu i programowaniem strukturalnym. Nabycie praktycznych umiejętności pisania prostych programów strukturalnych podzielonych na mniejsze bloki z wykorzystaniem funkcji własnych i standardowych.

Zakres tematyczny zajęć:

- programowanie strukturalne, proceduralne, modularne
- wykorzystanie funkcji standardowych,
- prototypy, definicje i wywołania funkcji własnych,
- parametry funkcji.

Kompendium wiedzy:

Programowanie strukturalne wykorzystuje ideę dzielenia programu na mniejsze części – bloki typu: sekwencja, wybór, iteracja z jednym wejściem i jednym lub kilkoma wyjściami. Jeżeli blokiem jest funkcja, umieszczona w pliku z funkcją główną `main`, może być ona wielokrotnie wykorzystana w tym samym programie – **programowanie proceduralne**. Większym blokiem do wielokrotnego wykorzystania w wielu programach jest moduł – plik źródłowy lub nagłówkowy dołączony do programu – **programowanie modułowe (modularne)**.

Funkcja – wydzielony fragment kodu spełniający określone czynności. Może zwracać wartość określonego typu (instrukcja **return**) lub nie (typ **void**).

Prototyp funkcji – zapowiedź funkcji:

```
typ_wyniku nazwa_funkcji();  
typ_wyniku nazwa_funkcji(parametry_formalne);
```

Definicja funkcji – opis algorytmu funkcji:

```
typ_wyniku nazwa_funkcji()  
{//deklaracje i instrukcje;  
    return wyrażenie; //opcjonalnie  
}  
typ_wyniku nazwa_funkcji(parametry_formalne)  
{//deklaracje i instrukcje;  
    return wyrażenie; //opcjonalnie  
}
```

Wywołanie funkcji:

```
nazwa_funkcji();  
nazwa_funkcji(parametry_aktualne);
```



W programie można wykorzystywać funkcje własne i funkcje standardowe, zgrupowane w odpowiednich modułach – dyrektywą **#include** należy dołączyć wybrany plik nagłówkowy.

Wykorzystanie funkcji standardowych:

```
#include <plik_naglowkowy_z_funkcjami>
```

```
nazwa_funkcji();
```

```
nazwa_funkcji(parametry_aktualne);
```

Grupy wybranych funkcji standardowych:

- Funkcje matematyczne <math.h>
 **asin(x), acos(x), atan(x), sin(x), cos(x), tan(x),
 exp(x), log(x), pow(x,y), sqrt(x), ceil(x), floor(x),
 fabs(x), fmod(x,y), ...**
- Funkcje łańcuchowe <string.h>
 strlen(), strcat(), strcmp(), strcpy(), ...
- Funkcje znakowe <ctype.h>
 tolower(), toupper(), isalpha(), isdigit(), isalnum(),
- Funkcje ogólnego użytku <stdlib.h>
 abs(), rand(), qsort(), ...

Prosty program proceduralny w języku C posiada następującą strukturę:

```
//dyrektywy preprocesora  
  
//prototypy funkcji własnych  
  
int main() //naglowek funkcji głównej  
{  
    //deklaracje  
    //instrukcje korzystające z ww. funkcji  
    return 0; //instrukcja powrotu  
}  
  
//definicje funkcji własnych
```

Pytania kontrolne:

1. Na czym polega programowanie strukturalne?
2. Na czym polega programowanie proceduralne?
3. Objaśnij pojęcie funkcji.
4. Jak wykorzystać w programie funkcje standardowe?
5. Co to jest prototyp funkcji?
6. Jak zdefiniować funkcję?
7. Jak wywołać funkcję?
8. Objaśnij pojęcia: parametry formalne i parametry aktualne funkcji.
9. Jak wygląda struktura programu pisana techniką (paradygmatem) programowania proceduralnego?



Zadania do analizy

Zadanie 4.1. Programowanie proceduralne - struktura programu w języku C

- Przeanalizuj przykład programu pisanego techniką proceduralną i porównaj go z kodem źródłowym zadania 2.2:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h> //M_PI
4
5  #define AUTOR "Jan Kowalski"
6  #define KIERUNEK "Informatyka"
7
8  //prototypy funkcji=====
9  void szlaczek(); //funkcja bez parametrów
10 void info(); //funkcja bez parametrów
11 float pole(float promien); //funkcja z 1 parametrem
12 float wyplata(int godziny, float stawka); //funkcja z 2 par.
13
14 int main(int argc, char *argv[]) //funkcja glowna=====
15 { //deklaracje
16     float r,st;
17     int lg;
18     //instrukcje
19     szlaczek(); //wywołanie funkcji
20     info();
21     szlaczek();
22     printf("Podaj promien kola "); scanf("%f",&r);
23     printf("Pole kola o promieniu %0.2f= %0.2f\n",r, pole(r));
24                                     //wywołanie funkcji
25     szlaczek();
26     printf("Pracownik1\n");
27     printf("Podaj liczbe godzin "); scanf("%d",&lg);
28     printf("Podaj stawke "); scanf("%f",&st);
29     printf("Wyplata = %0.2f\n", wyplata(lg,st));
30                                     //wywołanie funkcji
31     printf("Pracownik2\n");
32     printf("Podaj liczbe godzin "); scanf("%d",&lg);
33     printf("Podaj stawke "); scanf("%f",&st);
34     printf("Wyplata = %0.2f\n",wyplata(lg,st));
35                                     //wywołanie funkcji
36     szlaczek();
37     system("PAUSE");
38     return 0;
39 }
40 //definicje funkcji=====
41 void szlaczek()
42 {
```



```
42 printf("=====\n");
43 printf("*****\n");
44 printf("=====\n");
45 }//=====
46 void info()
47 { const int ROK=1;
48   printf("Programowanie liniowe\n");
49   printf("Autor programu: %s kierunek: %s rok:
50         %d\n", AUTOR, KIERUNEK, ROK);
51 } //=====
52 float pole(float promien)
53 {
54   return M_PI*pow(promien,2);
55 }//=====
56 float wypłata(int godziny, float stawka)
57 { const float PREMIA=0.20;
58   return godziny*stawka+ godziny*stawka*PREMIA;
59 }//=====
```

Zadanie 4.2. Programowanie proceduralne - struktura programu w języku C

- Przeanalizuj przykład programu pisanego techniką proceduralną.
- Podaj tekst w komentarzach.

```
1  #include <stdio.h> //???
2  #include <stdlib.h> //???
3  //ponizej sa ???
4  int suma(int a, int b);
5  int roznica(int a, int b);
6  int iloczyn(int a, int b);
7  float iloraz(int a, int b);
8  void iloraz2(int a, int b);
9
10 int main(int argc, char *argv[])
11 {int x,y,reszta;
12 printf("Podaj 2 liczby calkowite\n");//???
13 scanf("%d %d", &x, &y);//???
14 printf("suma: %d \n", suma(x,y));
15 printf("roznica: %d \n", roznica(x,y)); //???
16 printf("iloczyn: %d \n", iloczyn(x,y));
17 printf("iloraz - wynik dzielenia rzeczywistego:
18        %0.2f\n", iloraz(x,y));
19 iloraz2(x,y);
20 system("PAUSE");
21 return 0;
22 }
23 // ponizej sa???
24 int suma(int a, int b)
```



```
25 { return a+b; }

26 int roznica(int a, int b)
27 { return a-b;}
28 int iloczyn(int a, int b)
29 { return a*b;}

30 float iloraz(int a, int b)
31 { return (float)a/b; }

32 void iloraz2(int a, int b)
33 { printf("iloraz - wynik dzielenia całkowitego:
34         %d reszta: %d\n",a/b, a%b);
35 }
```

Zadania do wykonania

Zrealizuj zadania do wykonania z laboratorium 2 techniką programowania proceduralnego:

Zadanie 4.3. Dane studenta

Napisz funkcję, która na podstawie wprowadzonych z klawiatury danych (imię, nazwisko, wiek, płeć) i zdefiniowanych stałych (STATUS przyjmujący wartość student i SREDNIA przyjmująca wartość twojej pożądanej średniej ocen - liczba rzeczywista) wyświetli w jednym wierszu imię, nazwisko, i płeć, a w drugim twój status i średnią. Wywołaj tą funkcję.

Zadanie 4.4. Obliczanie objętości i pola powierzchni brył.

Napisz funkcje obliczające objętość i pole powierzchni sześcianu, prostopadłościanu o podstawie kwadratowej i prostokątnej oraz walca. Wywołaj te funkcje.

Zadanie 4.5. Obliczanie średniej arytmetycznej i średniej geometrycznej dwóch liczb całkowitych

Napisz funkcje obliczające średnią arytmetyczną i średnią geometryczną dwóch liczb całkowitych. Wywołaj te funkcje.

Zadanie 4.6. Przeliczanie wielkości fizycznych

Napisz funkcje przeliczające wielkości fizyczne:

- z mili na kilometry (1 mila=1.6 km),
- z kilometrów na mile (1 km=0.62 mili),
- z koni mechanicznych [KM] na waty [W] (1KM = 735W)),



Fundusze Europejskie
Wiedza Edukacja Rozwój



**Rzeczpospolita
Polska**

Unia Europejska
Europejski Fundusz Społeczny



- z kilometrów na godzinę [km/h] na metry na sekundę [m/s],
- ze stopni Fahrenheita [$^{\circ}\text{F}$] na stopnie Celsjusza [$^{\circ}\text{C}$] ($t^{\circ}\text{C} = 5/9(t^{\circ}\text{F} - 32)$).

Zadanie 4.7. Wiek w przyszłości

Napisz funkcję, która na podstawie twojego obecnego wieku (w latach i miesiącach) i podanego okresu czasu (w miesiącach), obliczy twój wiek w przyszłości (w latach i miesiącach). Wywołaj tę funkcję.

Zadania dodatkowe

Zadanie 4.8. Wyrażenia

Napisz i wywołaj funkcje obliczające wartość wyrażeń (wykorzystaj funkcje standardowe):

a) $10\cos x - 0,1x^2 + \sin x + \sqrt{4x^2 + 7}$

b) $\lg(x + 5) + e^{x+1} - |tgx + 1|$

c) $\frac{\sin^2 \alpha + 0,5}{\cos \alpha^4 + tg^4 \alpha^2}$

d) $\sqrt{\frac{|5\sin \beta^5 + 1|}{3,5(\sin \beta + \cos \beta)^2}}$

Zadanie 4.9. Wyrażenia - funkcje trygonometryczne

Napisz i wywołaj funkcję obliczającą wartość funkcji trygonometrycznych $\sin \alpha$, $\cos \alpha$ i $tg \alpha$ dla α podanego w stopniach np. 90° , 120° , 180° .

Zadanie 4.10. Zamiana miejscami zmiennych i liczba odwrotna

Zmienne a i b to dwie liczby całkowite trzycyfrowe. Napisz i wywołaj funkcję zamieniającą miejscami wartości tych zmiennych. Po zamianie wyświetl te liczby, a następnie zmodyfikuj je zamieniając miejscami cyfrę jedności i cyfrę setek - wyświetl liczby w odwrotnej kolejności cyfr.

Zadanie 4.11. Obwód okręgu

Napisz i wywołaj funkcję, która obliczy obwód okręgu, który przechodzi przez punkt A(x1, y1) i którego środek znajduje się w punkcie B(x2, y2).



Zadanie 4.12. Znaki

Napisz i wywołaj funkcje, które:

- a) po podaniu dowolnego znaku wyświetli go wraz z kodem ASCII, a następnie wyświetli znak o kodzie następnym,
- b) po podaniu małej litery zamieni ją na dużą.



Fundusze Europejskie
Wiedza Edukacja Rozwój



**Rzeczpospolita
Polska**

Unia Europejska
Europejski Fundusz Społeczny

