

**Міністерство освіти і науки України Національний
університет “Львівська Політехніка”**



**Лабораторна робота №13-14
з дисципліни «Програмування
частина № 2»**

Виконав:
Студент групи АП-11
Іщенко Василь

Прийняв:
Чайковський І.Б.

Львів – 2024

Тема: Структура функції. Локальні та глобальні змінні. Класи пам'яті

Мета роботи: навчитися використовувати функції у процесі програмування, розуміти особливості використання локальних та глобальних змінних та специфікаторів різних класів пам'яті.

Теоретичні відомості

При програмуванні будь-яких задач, крім найпростіших, постійно виникає потреба виконувати в кількох різних місцях алгоритму одні й ті самі дії над різними значеннями. Наприклад, нехай потрібно обчислити площу трьох різних трикутників, перший з яких має сторони a_1, b_1, c_1 , другий – a_2, b_2, c_2 , третій – a_3, b_3, c_3 .

Класи пам'яті. Однією з головних переваг мови C є те, що вона дає змогу керувати ресурсами програми. До таких ресурсів відноситься і пам'ять програм. Кожна змінна в програмі належить до певного типу, який визначає скільки пам'яті необхідно виділити для її збереження. Крім того, кожна змінна належить до певного класу пам'яті, що визначає час її існування та область видимості. Час існування змінної - це період, протягом якого змінна існує в пам'яті, а область видимості (область дії) – це частина програми, в якій змінна може використовуватися. В C визначено чотири специфікатори класу пам'яті: `auto`, `register`, `extern` і `static`.

Таблиця

Область дії та час існування змінних різних класів пам'яті

Клас пам'яті	Ключове слово	Тривалість існування	Область дії
Автоматичний	Auto	Тимчасово	Блок (локальна)
Регістровий	Register	Тимчасово	Блок (локальна)
Статичний	Static	Постійно	Блок (локальна)
Зовнішній статичний	Static	Постійно	Файл (один файл, глобальна)
Зовнішній	Extern	Постійно	Програма (всі файли, глобальна)

Приклад 1

```
#include<stdio.h>
#include<math.h>
double Geron (double a,double b,double c); /*оголошення функції- прототипу*/
double Geron (double a, double b, double c) { /*реалізація функції*/
double p; /* локальна змінна */
p= (a + b + c)/2;
return sqrt(p*(p-a)*(p-b)*(p-c));/*обчислити вираз і повернути результат*/
}
int main ( )
{ double u, v, w;
```

```

double s;
printf ("Vvedit storonu trikutnika"); /* у консолі введіть значення 2, 3, 4 */
scanf ("%lf %lf %lf",&u, &v,&w);
s=Geron(u,v,w); /*викликається функція Geron. До аргументів a,b,c
передаються значення змінних u,v,w. Результат виклику присвоюється
змінній s*/
printf("Ploscha 1 trikutnika %lf\n",s);
s=Geron (10.3,8.1,9.7); /* викликається функція Geron. До аргументів a,b,c
передаються константи */
printf("Ploscha 2 trikutnika %lf\n",s);
s=Geron(u+10.3,v+w,w*1.7); /* викликається функція Geron. До аргументів
передаються значення виразів */
printf("Ploscha 3 trikutnika %lf\n",s);
return 0;
}

```

```

Vvedit storonu trikutnika 15 15 15
Ploscha 1 trikutnika 97.427858
Ploscha 2 trikutnika 36.928095
Ploscha 3 trikutnika 307.460239

```

Приклад 2

```

#include<stdio.h>
#include<math.h>
double fract(int,int);
double factorial(int);
int main( ) {
    int m,n,s;
    scanf ("%d %d",&m,&n);
    printf ("%lf\n",fract(m+1,n+1));
    return 0;}
double fract(int x, int y) {
    double t;
    t=factorial(x+y)/(x*y);
    return t;}
double factorial (int n) {
    int i;
    double p;
    p=1;
    for(i=1;i<=n;++i)
        p*=i;
    return p;}

```

```

1 2
20.000000

```

Приклад 3

```
#include <stdio.h>
#include <windows.h>
int func(int x);
char str[50];
void main(){
    SetConsoleCP(65001);
    SetConsoleOutputCP(65001);
    int a = 0;
    printf("Введіть строку: ");
    scanf("%s",&str);
    printf("Введіть скільки символів повинно вивести: ");
    scanf("%d",&a);
    func(a);
}
int func(int x){
    for(int i=0;i<x;i++){
        printf("%c",str[i]);
    }
}
```

Введіть строку: Hello
Введіть скільки символів повинно вивести: 2
He

Відповіді на контрольні запитання

1) Дайте визначення поняття функція в мові C:

У мові програмування C функція - це блок коду, який має своє ім'я та може виконуватися певною послідовністю команд. Функції дозволяють організовувати програму на більш логічні блоки, полегшуючи її розуміння та управління.

2) Поясніть призначення типу void:

Тип void використовується для вказівки на те, що функція не повертає значення. Функції з типом void використовуються для виконання певних операцій без повернення будь-яких результатів.

3) Чим локальні змінні відрізняються від глобальних?

Локальні змінні оголошуються всередині блоку коду функції та існують лише в межах цього блоку.

Глобальні змінні оголошуються поза будь-якими функціями і можуть бути доступні для всіх функцій в програмі.

Локальні змінні існують лише під час виконання функції, в той час як глобальні змінні існують протягом всього часу роботи програми.

4) Яку функцію називають рекурсивною?

Функцію, яка викликає саму себе, називають рекурсивною.

5) Назвіть переваги і недоліки використання рекурсивної функції:

Переваги:

Рекурсивні функції можуть бути коротшими та більш зрозумілими, особливо для вирішення завдань, що вимагають рекурсивного підходу.

Вони дозволяють легше розв'язувати завдання, що мають структуру, схожу на дерево або складну індуктивну структуру даних.

Недоліки:

Рекурсивні функції можуть призвести до перевищення обсягу стеку, особливо якщо рекурсивні виклики глибоко вкладені або кількість викликів дуже велика.

Вони можуть бути менш ефективними в порівнянні з ітеративними рішеннями через додаткові витрати пам'яті та часу на кожен рекурсивний виклик.

6) Назвіть специфікатори класів пам'яті, які використовуються у мові C:

У мові C використовуються такі специфікатори класів пам'яті:

auto: Змінні, що оголошуються з цим специфікатором, зберігаються в автоматичній області пам'яті (стеку).

register: Змінні, що оголошуються з цим специфікатором, зберігаються в регістрах процесора, якщо це можливо.

static: Змінні, що оголошуються з цим специфікатором, зберігаються в статичній області пам'яті

Висновок: Я навчився використовувати функції у процесі програмування, розуміти особливості використання локальних та глобальних змінних та специфікаторів різних класів пам'яті.