Семинар 4. Указатели на функции, передача параметров по указателю/ссылке, функции и массивы

1. Указатели на функции

При использовании имени функции без скобок и параметров имя функции выступает в качестве указателя на функцию, его значением служит адрес размещения кода функции в памяти. Это значение может быть присвоено другому указателю.

1. Указатели на функции

Указатель на функцию определяется следующим образом:
тип (*имя_указателя)(специф_парам);

1. Указатели на функции

Пример №1.

```
#include <iostream>
using namespace std;

void f1(){cout<<"Call f1"<<endl;}
void f2(){cout<<"Call f2"<<endl;}

int _tmain(int argc, _TCHAR* argv[])
{
        void (*ptr)(void); //ptr – указатель на функцию void funcName(void);
        ptr = f2;
        (*ptr)(); //Вызов f2 по её адресу
        ptr = f1;
        ptr(); //Вызов f1
        //Эквивалентно (*ptr)();
        return 0;
}
```

1. Указатели на функции

Пример №2.

```
int add(int a, int b){return a+b;}
int sub(int a, int b){return a-b;}
int _tmain(int argc, _TCHAR* argv[])
{
        int (*func)(int, int);
        char ch;
        int a, b;
        cout << "Input a, b: ";
        cin >> a >> b;
        cout << "Input operation: ";
        cin >> ch;
        if (ch == '+' || ch =='-'){
                switch(ch){
                case '+': func = add; break;
                case '-': func = sub; break;
                }
                cout << "Result = " << func(a, b);
        }
        return 0;
}
```

2. Передача параметров по указателю / ссылке

При передаче параметров в функции без использования ссылок и указателей фактически создаётся локальная копия переданного значения. Если изменить переданное значение внутри функции, то изменится только локальная переменная, область видимости которой – данная функция.

2. Передача параметров по указателю / ссылке

Пример.

```
int add(int a, int b){a++; return a+b;}

int _tmain(int argc, _TCHAR* argv[])
{
    int a = 5, b = 6;
    cout << add(a, b);
    cout << endl << "a = " <<a; // a = 5
}
```

2. Передача параметров по указателю / ссылке

Для доступа из тела функции к соответствующему аргументу необходимо использование ссылок или указателей.

Параметр-ссылка обеспечивает те же возможности, что и параметр-указатель.

При этом:

- *ссылки не надо разыменовывать*
- *при применении параметра-ссылки аргументом должен быть не указатель, а обычная переменная*

3. Функции и массивы

Правило при создании функций для работы с массивами: внутри функции необходимо знать количество элементов массива.

3. Функции и массивы

Пример №1. Функции для работы с массивами (инициализация и вывод).

```
void maxVect(int n, int *x, int *y, int *z){
	for(int i = 0; i < n; i++)
		z[i] = x[i] > y[i] ? x[i] : y[i];
}
void printVect(int n, int *x){
	for(int i = 0; i < n; i++)
		cout << x[i] << " ";
	cout << endl;
}

void main()
{
	int a[] = {1, 2, 3, 4, 5, 6, 7};
	int b[] = {8, 7, 6, 5, 4, 3, 2};
	int c[7];
	maxVect(7, a, b, c);
	printVect(7, c);
}
```

3. Функции и массивы

Пример №2. Функция для подсчёта длины строки.

```
int len(char str[]){
        int l = 0;
        while(str[l++]) ;
        return l - 1;
}

void main()
{
        char stroka[] = "Bauman Moscow State Technical University";
        cout << "Length = " << len(stroka);
}
```

4. Задания

Задания.

1. Дополнить пример №2 раздела «Указатели на функции» другими операциями (умножения, получения остатка и т.п.), выводить сообщение при вводе неверной операции.
2. Создать функцию для записи в элементы одномерного массива их индексов (x[0] == 0, x[1] == 1 и т.д.).
3. Создать функцию для записи в элементы одномерного массива чисел от N до 1, где N – размерность массива. (x[0] == N, x[1] == N-1 и т.д.)
4. Создать функцию для сдвига элементов массива таким образом, чтобы в x[0] оказалось значение x[1], в x[1] – значение x[2], … , в x[N-1] значение x[0].