

Лабораторная работа № 2

Работа с указателями и ссылками в C++

Оглавление

Цель:	1
Инструкция:.....	2
Задания для самостоятельного выполнения:	3
Домашнее задание:.....	4
Требования к оформлению программ:	5
Контрольные вопросы:	6

Цель:

Научиться работать с динамическими переменными и ссылками

Инструкция:

```
#include <iostream>

using namespace std;

int main()
{
    int a=10;    //объявление автоматической переменной
    int *b;      //объявление динамической переменной

    //----инициализация динамической переменной автоматической и работа с ней (начало)----

    b=&a;        //инициализация динамической переменной автоматической переменной

    cout<<"----1----\n";
    cout<<"adres a: "<<&a<<" a="<<a<<endl;    //вывод адреса и значения динамической переменной
    cout<<"adres b: "<<&b<<" b="<<*b<<endl;    //вывод адреса и значения автоматической переменной

    cout<<"----2----\n";
    a=4;        //изменение значения автоматической переменной
    cout<<"adres a: "<<&a<<" a="<<a<<endl;
    cout<<"adres b: "<<&b<<" b="<<*b<<endl;

    cout<<"----3----\n";
    b++;        //изменение адреса динамической переменной
    cout<<"adres a: "<<&a<<" a="<<a<<endl;
    cout<<"adres b: "<<&b<<" b="<<*b<<endl;

    cout<<"----4----\n";
    b=&a;        //возврат к начальным данным
    (*b)++;      //изменение значения динамической переменной на 1
    cout<<"adres a: "<<&a<<" a="<<a<<endl;
    cout<<"adres b: "<<&b<<" b="<<*b<<endl;

    cout<<"----5----\n";
    b=&a;        //возврат к начальным данным
    *b++;        //изменение адреса и затем разыменование
    cout<<"adres a: "<<&a<<" a="<<a<<endl;
    cout<<"adres b: "<<&b<<" b="<<*b<<endl;

    cout<<"----6----\n";
    b=&a;        //возврат к начальным данным
    ++*b;        //разыменование и увеличение значения на 1
    cout<<"adres a: "<<&a<<" a="<<a<<endl;
    cout<<"adres b: "<<&b<<" b="<<*b<<endl;

    //----инициализация динамической переменной автоматической и работа с ней (конец)----
```

```

//----выделение и освобождение памяти под динамическую переменную(начало)----
cout<<"----7----\n";
b=new int;           //выделение памяти под динамическую переменную
cout<<"adres a: "<<&a<<" a="<<a<<endl;
cout<<"adres b: "<<b<<" b="<<*b<<endl;

cout<<"----8----\n";
*b=7;               //изменение значения динамической переменной
cout<<"adres b: "<<b<<" b="<<*b<<endl;

cout<<"----9----\n";
//вывод размеров адреса и значения динамической переменной
cout<<"adres size b: "<<sizeof(b)<<" size of b="<<sizeof(*b)<<endl;

cout<<"----10----\n";
delete b;           //освобождение памяти
cout<<"adres b: "<<b<<" b="<<*b<<endl;
/*
cout<<"----11----\n";
b=NULL;             //присваивание адресу неопределенного указателя
cout<<"adres b: "<<b<<" b="<<*b<<endl;
*/
//----выделение и освобождение памяти под динамическую переменную(конец)----

//----работа со ссылками(начало)----
cout<<"----12----\n";
int &r=a;            //объявление ссылки
cout<<"adres a: "<<&a<<" a="<<a<<endl;
cout<<"adres r: "<<&r<<" r="<<r<<endl;

cout<<"----13----\n";
r=100;              //изменение значения переменной-ссылки
cout<<"adres a: "<<&a<<" a="<<a<<endl;
cout<<"adres r: "<<&r<<" r="<<r<<endl;

cout<<"----14----\n";
a=70;               //изменение значения инициализатора ссылки
cout<<"adres a: "<<&a<<" a="<<a<<endl;
cout<<"adres r: "<<&r<<" r="<<r<<endl;
//----работа со ссылками(конец)----

return 0;
}

```

Задания для самостоятельного выполнения:

1. Объявите динамическую переменную – указатель на тип данных **float* b** и выполните с ней следующие действия:
 - а) Инициализировать ее значением нединамической переменной. Вывести адреса и значения обеих переменных. Определить количество байтов, необходимое для хранения адресов и значений обеих объявленных переменных.

- b) Выделить память для объявленной переменной. Вывести адрес и значение переменной. Определить количество байтов, необходимое для хранения адреса и значения переменной. Освободить память.
 - c) Примените операцию инкремента ++ к переменной **b**. Как изменился адрес и значение переменной?
 - d) Примените операцию декремента -- к переменной **b**. Как изменился адрес и значение переменной?
2. Объявите переменную – ссылку на тип данных **float & b**. Вывести адреса и значения переменной-ссылки и переменной- инициализатора. Определить количество байтов, необходимое для хранения адресов и значений объявленных переменных.
3. Выполните задание 1 и 2 для переменной **double* b (double & b)**.
4. Выполните задание 1 и 2 для переменной **char* b (char& b)**.

Домашнее задание:

Выполнять следующие операции с динамической переменной:

```

x++;      ++x;      *x++;      ++*x;
x--;      --x;      *x--;      --*x;
(*x)++;   (*x)--;   ++(*x);   --(*x);
*x+=1;    *x+=2;    x+=1;     x+=2;    x=1;

```

Заполнить сравнительную таблицу (в тетради или текстовом файле):

операции	x		y		комментарии
	адрес	значение	адрес	значение	
<i>int x=10;</i> <i>int *y;</i> <i>y=&x;</i>					Начальные данные
y++;					
++y;					
y--;					
--y;					
*y++;					
++*y;					
*y--;					
--*y;					
(*y)++;					
++(*y);					
(*y)--;					
--(*y);					
*y+=1;					
*y+=2;					
y+=1;					

y+=2;					
y=2;					

Пояснение: Для заполнения таблицы, в код программы добавлять после вывода адресов и значений переменных `y` и `y` строчку с проверяемой операцией и снова выводить адреса и значения переменных. Перед выполнением каждой следующей операции выполнить возврат к начальным данным. В комментариях описать, как изменились адрес и значение переменных. Если какая-то из операций привела к ошибке компиляции – так и писать в комментариях, пояснив, что за ошибку вывел компилятор, что она означает и почему возникла.

Пример программы:

```
#include <iostream>

using namespace std;

int main()
{
    int x=10;
    int *y;

    cout<<"---->1 \n";
    y=&x;          //начальные данные
    cout<<"adres x: "<<&x<<" x="<<x<<endl;
    cout<<"adres y: "<<y<<" y="<<*y<<endl;

    cout<<"---->2  x-- \n";
    y=&x; x=10;
    x--;
    cout<<"adres x: "<<&x<<" x="<<x<<endl;
    cout<<"adres y: "<<y<<" y="<<*y<<endl;

    cout<<"---->3  x++ \n";
    y=&x; x=10;
    x++;
    cout<<"adres x: "<<&x<<" x="<<x<<endl;
    cout<<"adres y: "<<y<<" y="<<*y<<endl;

    return 0;
}
```

Требования к оформлению программ:

1. **Содержание.** Программа должна делать то, что предусмотрено заданием, и больше ничего делать не должна
2. **Спецификация.** В преамбуле программы в комментариях указывать сведения:
 - Кто выполнил.
 - Что делает программа (кратко).
 - Что на входе, имена входных файлов указываются.
 - Что на выходе (что является результатов работы программы).
3. **Ввод и вывод**

- Приглашение пользователю. (Например, сколько чисел, какого типа и через какой разделитель нужно вводить).
 - Контрольный вывод (все введенные данные выводить на экран, и только после этого выполнять необходимые вычисления.)
 - «Защита от дурака». Проверять вводимые данные на корректность. Например, если необходимо считать количество чего – то, то эта величина не может быть отрицательной и т.д.
4. **Структура кода.** Набираемый код должен быть хорошо структурированным. Использовать:
- Отступы.
 - Комментарии – поясняют решение программы.
 - Осмысленные названия переменных.
 - Пояснения о назначении переменных в комментариях (кроме счетчиков).

Контрольные вопросы:

1. Что такое адрес переменной?
2. Что такое значение переменной?
3. Как объявить динамическую переменную?
4. Как инициализировать динамическую переменную?
5. Как найти адрес нединамической переменной?
6. Как найти значение нединамической переменной?
7. Как найти адрес динамической переменной?
8. Как найти значение динамической переменной?
9. К чему приведет применение оператора инкремента к адресу?
10. К чему приведет применение оператора декремента к адресу?
11. Что такое **new** и для чего используется?
12. Что такое **delete** и для чего используется?
13. Что такое **NULL**?
14. Что такое утечка памяти?
15. Чем опасна утечка памяти?
16. Что такое ссылка?
17. Как вывести адрес и значение переменной-ссылки?
18. Можно ли адресу присвоить число?