

C++ course

*name: Виктория Александровна
e-mail: kewtree1408@gmail.com*

*source: <https://bitbucket.org/Kewtree/coursecpp/src>
login: coursecpp
password: 12345*

General Plans

1. Inception
- 2-3. Statements
4. Functions
5. Pointers
6. Massives
7. Structures
8. Files, OOP: encapsulation
9. OOP: inheritance
10. OOP: polimorph
11. Test
12. Game over

Pointers

1. Introduction
2. Get address. &
3. Heap & stack
4. Get pointer. *
5. Arithmetic operations with pointers
6. Arguments in functions:
 - by value
 - by pointers
 - by link
7. Function as function argument
8. Errors with pointers
9. Massives (introduction)

Pointers

Указатели. Можно ли без них обойтись?
Зачем они нужны?

- 1! Эффективное распределение памяти
- 2! Увеличение производительности (времени работы программы)
3. Более глубокое понимание устройства переменных и объектов в C++
4. Возврат из функции более одного значения

Pointers

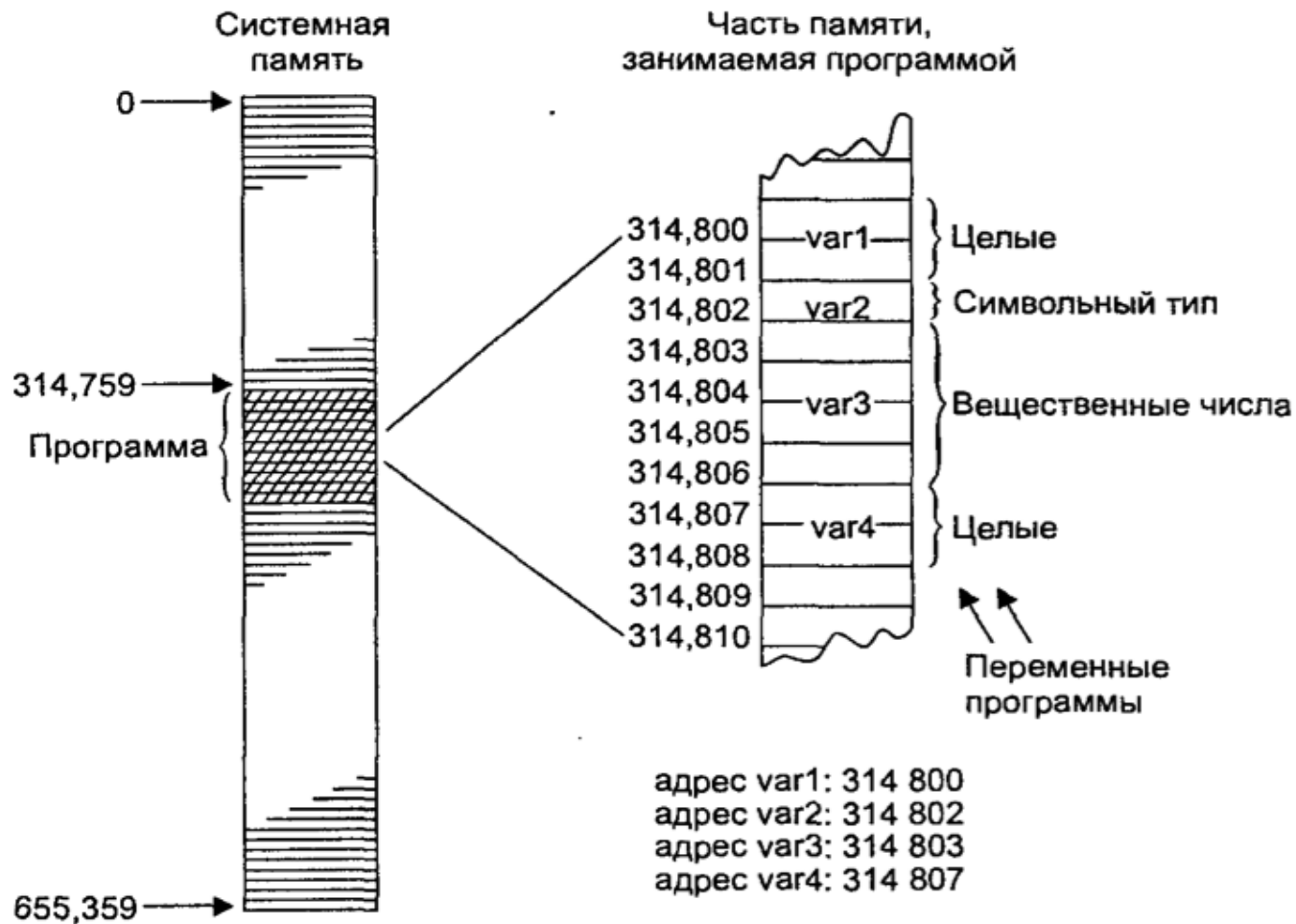


Рис. 10.1. Адреса в памяти

Pointers

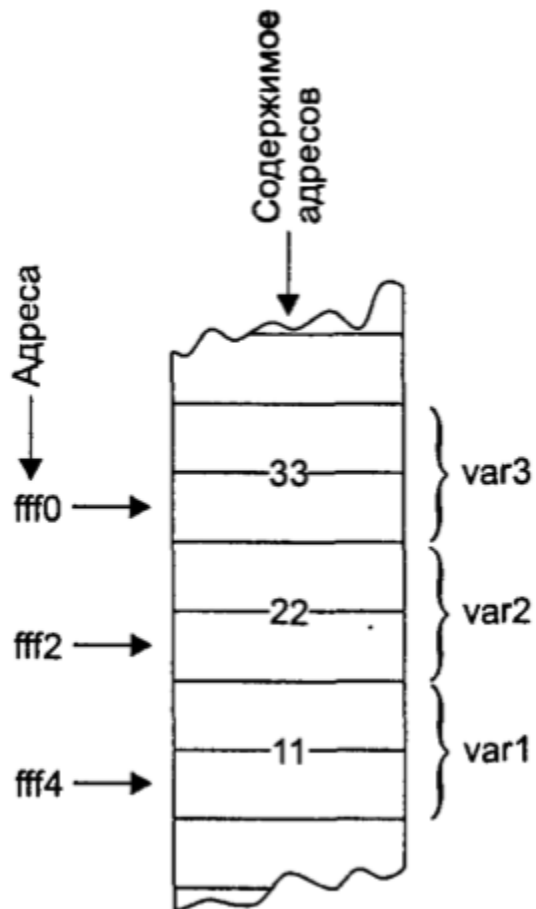


Рис. 10.2. Адреса и содержимое переменных

Pointers: *, &

* -- получение значения по известному адресу

& -- получение адреса по известному значению

Get address

`ampersand.cpp`

Напишем программу, которая показывает адреса, в которых хранится наша переменная.

Какая разница между адресами?
А в режиме Debug/Realise?

Heap and Stack

```
int a;  
float b = 5;  
double c = 3.5;
```

```
int *a = new int;  
float *b = (float *) malloc (sizeof  
(float));  
double *d = new double;  
delete (a);  
free(b);  
delete(c);
```

Get pointer

```
int* ptr; // объявление: указатель на int  
*ptr = 37; // разыменование: значение переменной, адресованной через ptr
```

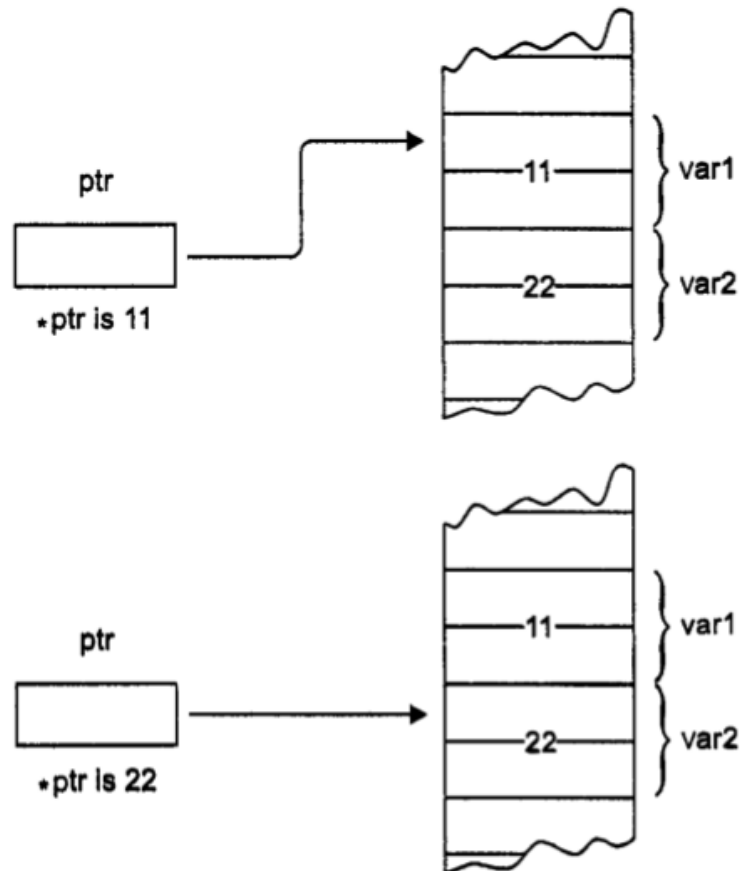


Рис. 10.4. Доступ к переменным через указатели

Get pointer

pointers.cpp

Назначение указателей.

Operations with pointers

+

-

*

/ нельзя.

arif_ptr.cpp

Арифметические операции с указателями.

Arguments in function

Передача параметров:

1. по значению
2. по указателю
3. по ссылке
4. по константной ссылке или константному указателю

`func_arguments.cpp`

Пример со сложением чисел -- со всеми видами передачи параметров

Function as argument

```
char func_name (int , float, double  
(*nameFunc)(double, int), double );
```

func_as_argument.cpp

Пример передачи в качестве параметра другой функции.

Errors in pointer

```
#include <iostream>
using namespace std;
int *global;

void Error1() {
    int a=100000;
    int *pa;

    cout << "a = " << a << endl;
    cout << "*pa = " << *pa << endl;
    cout << "pa = " << pa << endl;
    // pa = &a; // так можно
    *pa = a;
    cout << "pa = " << pa << endl;
    cout << "*pa = " << *pa << endl;
}

void Error2() {
    cout << "global = " << global << endl;
    cout << "*global = " << *global << endl;
}

void Error3() {
    int *a=reinterpret_cast<int*>(100);
    cout << *a << endl;
    // *a = 100;
}

int main() {
    // Error1(); // неинициализируемый указатель (в дебаге не работает, а в версии релиз работает)
```

Massives

<type of elements> <name of massive> [size of massive];

```
int a[5];
```

```
for (int i=0; i<5; i++)
```

```
    cout << a[i] << endl;
```


Homework

- 1) Написать функцию, которая по введенному вещественному числу возвращает целую и дробную часть. (Показать практическое использование возврата нескольких значений из функции)
- 2) Написать функцию, которая принимает 4 параметра по ссылке и такую же функцию (с теми же операциями), которая принимает на вход 4 параметра по указателю. В теле функций нужно увеличить каждое значение на 10. Результат вывести в главной функции.
- 3) Написать функцию, которая принимает на вход один символ и возвращает этот же символ но в верхнем и нижнем регистре, а также номер этого символа в таблице ASCII (т.е. функция принимает на вход 1 параметр по умолчанию и 3 по ссылке или указателю). (функции `tolower`, `toupper` из библиотеки `ctype.h` -- в помощь)
- 4) Разобрать пример из файла `dichotomy.cpp` и задать вопросы.