Программирование на языке C++
Лекция 2.7.1

Указатели

```
...

void swap(int a, int b){
    int  temp  =  a;
    a  =  b;
    b  =  temp;
}

int main(){
    int i1 = 5;
    int i2 = 0;
    swap(i1, i2);
    cout << i1 << ' ' << i2;
}
```

```
int i;
```

```
int i;
int j;
```

Имя переменной	Тип переменной	Размер, байт	Адрес
i	int	4	0x00EC4682
j	int	4	0x00EC46CE

```
&i;        // Получение адреса
&"Hello"; // Получение адреса
```

```
// Может хранить почти любой адрес
void* ptr;
```

```
int i = 12.3;
char c = ‘$’;

void* p;
p = &c;
p = &i;
```

```
int i = 1;
double d = 10;
char c = '~';

int* ip = &i;
double* dp = &d;
char* cp = &c;
```

```
int* ip = NULL;
double* dp = NULL;
char* cp = NULL;

int* ip = nullptr;
double* dp = nullptr;
char* cp = nullptr;
```

```
int i = 2;
int j = 10;

int* p = &i;
p = &j;
```

```
int i       = 2;
const int j = 10;

int* p = &i;
p = &j;       // запрещено
```

```
int i       = 2;
const int j = 10;

const int* p = &i;
p = &j;
```

```
int i = 2;
int j = 10;

int* const p = &i;
p = &j;                 // запрещено
int* const p2;          // запрещено
```

```
const int i = 10;

const int* const p = &i;
```

```
int* const p;
int const *p;
const int * const p;
const int * const *p;
```

```
*p; // Оператор разыменования
```

```
// Доступ на чтение
std::cout << *p;

// Доступ на запись
*p = 10;
```

Переменная способная хранить адрес – указатель (pointer)

```
...

void swap(int a, int b){
    int  temp  =  a;
    a  =  b;
    b  =  temp;
}

int main(){
    int i1 = 5;
    int i2 = 0;
    swap(i1, i2);
    cout << i1 << ' ' << i2;
}
```

```
...

void swap(int *a, int *b){
    int  temp  =  *a;
    *a  =  *b;
    *b  =  temp;
}

int main(){
    int i1 = 5;
    int i2 = 0;
    swap(&i1, &i2);
    cout << i1 << ‘ ‘ << i2;
}
```

```
...

void print(vector<int>* a){
    for(int i=0; i<(*a).size(); i++)
        cout << (*a)[i] << endl;
}

int main(){
    vector<int> arr = {5,2,9,1,4};
    print(&arr);
}
```

```
...

void print(vector<int>* a){
    for(auto item : *a)
        cout << item << endl;
}

int main(){
    vector<int> arr = {5,2,9,1,4};
    print(&arr);
}
```

Некоторые операции

```
int i = 12.3;
int* c = &i, j = 10;

&c; // Получение адреса
*c; // Разыменование

int* p;
p = c; // Копируем адрес

int **p2 = &p; // Указатель на указатель
```

Оператор + -

```
int* p = 0;
cout << p;  // 00000000
p = p + 1;  // 00000004

double* p = 0;
cout << p;  // 00000000
p = p + 1;  // 00000008

// Только с целыми числами
```

Оператор -

```
int   i = 12,   j = 12;
int *pi = &i, *pj = &j;
std::cout << pi; // 004FFC3C
std::cout << pj; // 004FFC30
std::cout << (pj - pi); // -3
```

Операторы ++ --

```
pi++;  // pi = pi + 1
pi--;  // pi = pi - 1
```

Операторы сравнения

```
int   i = 12,   j = 12;
int *pi = &i, *pj = &j;
std::cout << pi; // 004FFC3C
std::cout << pj; // 004FFC30

if (pi > pj) std::cout << ">";
if (pi < pj) std::cout << "<";
if (pi == pj) std::cout << "==";
```

Операторы и void*

Разрешены операторы:

- + и – с целым числом (зависит от компилятора) – не использовать;
- операторы сравнения;
- операторы приведения типа;

Размер

Зависит от платформы:
- на 32 битной – 4 байта;
- на 64 битной – 8 байт;

Можно узнать с помощью sizeof()

Дополнительно

Что такое size_t и ptrdiff_t: https://www.viva64.com/ru/a/0050/