

# Использование функции swar

Александр Смаль

**Академический университет**  
(дополнительные материалы)  
Санкт-Петербург

## Идея swap

Функция `swap` — это примитив для обмена значений двух переменных. Для большинства типов можно определить функцию `swap` следующим образом:

```
template<typename T>
void swap(T & a, T & b) {
    T tmp(a);
    a = b;
    b = tmp;
}
```

Можно также использовать реализацию функции `swap` из стандартной библиотеки.

```
#include <utility>
using std::swap;

int main() {
    int a = 1;
    int b = 2;
    swap(a, b);
    // a = 2, b = 1
}
```

## Определение swap для сложных объектов

Для сложных объектов функцию swap лучше реализовывать отдельно.

```
#include <utility>
using std::swap;

template<typename T>
struct Array {
    ...

    void swap(Array & a) {
        swap(size_, a.size_);
        swap(data_, a.data_);
    }
private:
    size_t size_;
    T *    data_;
};

template<typename T>
void swap(Array<T> & a, Array<T> & b) {
    a.swap(b);
}
```

## Использование swap для operator=

Функция swap очень удобна для реализации operator= через конструктор копирования.

```
template<typename T>
struct Array {
    Array(Array const& a)
        : size_(a.size_)
        , data_(a.data_)
    { /* implementaion */ }

    Array & operator=(Array const& a) {
        if (this != &a)
            Array(a).swap(*this);
        return *this;
    }
    ...
    void swap(Array & a) { /* implementaion */ }
private:
    size_t size_;
    T *    data_;
};
```