

Программирование на языке C++

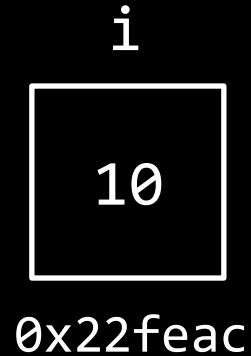
Лекция 1.2

Ссылки

```
int i = 10;
```

```
cout << i; // 10
```

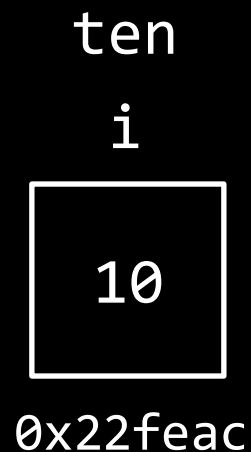
```
cout << &i; // 0x22feac
```



Независимые ссылки

```
int i = 10;
```

```
int& ten = i;
```



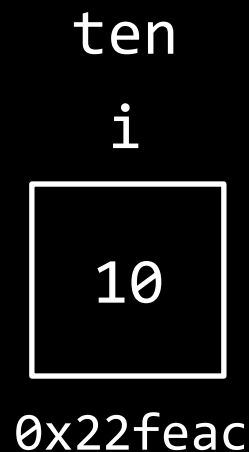
`&` - НЕ оператор взятия адреса

Независимые ссылки

```
int i = 10;  
int& ten = i;
```

```
cout << i;    // 10  
cout << &i;   // 0x22feac
```

```
cout << ten;   // 10  
cout << &ten;  // 0x22feac
```



Независимые ссылки

```
int i[4] = {1,2,3,4};  
int& x = i[2];
```

```
x = 10;
```

```
cout << i[2]; // 10
```

x			
i[0]	i[1]	i[2]	i[3]
1	2	3	4

x			
i[0]	i[1]	i[2]	i[3]
1	2	10	4

Независимые ссылки

```
double d = 1.5;  
double& x = d;
```

```
double d;  
d = 1.5;
```

```
double d = 1.5;  
double& x; // Ошибка  
x = d;
```

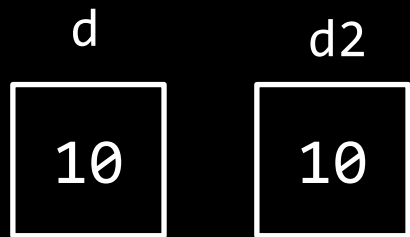
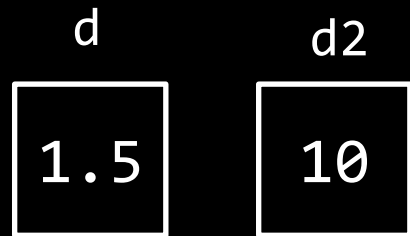
Независимые ссылки

```
double d = 1.5;
```

```
double d2 = 10;
```

```
double& x = d;
```

```
x = d2; // Копирование значения d2
```



Независимые ссылки

```
int i[4] = {1,2,3,4};
```

```
int& j[4] = {1,2,3,4}; // Ошибка
```


Независимые ссылки. rvalue

`int& i = 10;` `//` Ошибка. Литерал

`int& i = 1 + 1;` `//` Ошибка. Временный объект

`double d = 18.5;`

`int& i = d;` `//` Ошибка. Временный объект

Независимые ссылки. const

```
const int& i = 10;
```

```
const int& i = 1 + 1;
```

```
double d = 18.5;
```

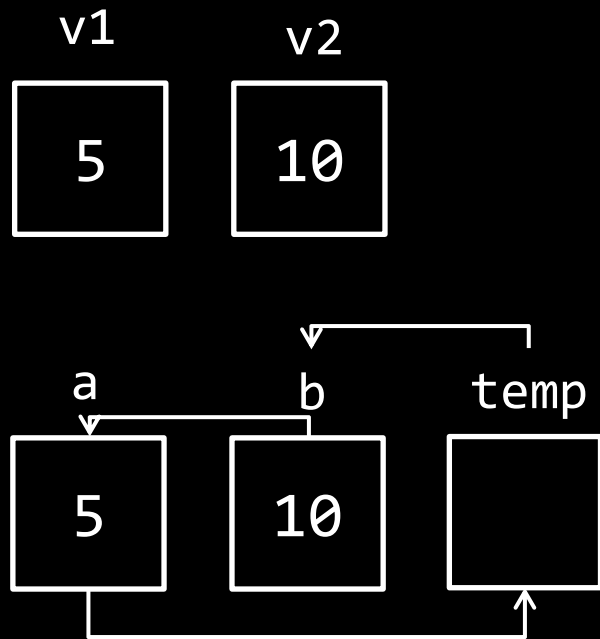
```
const int& i = d;
```

Ссылки как параметры функции

swap 1 Передача параметров по значению

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

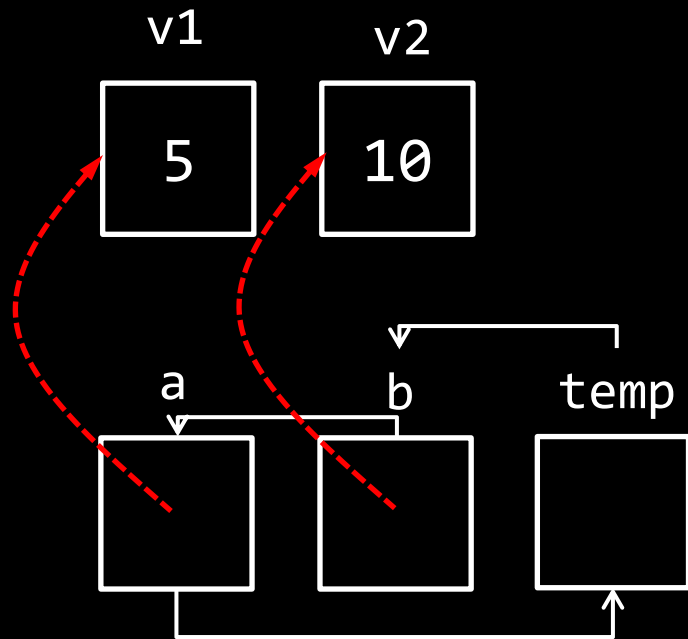
```
int v1 = 5, v2 = 10;  
swap(v1, v2);  
cout << v1 << ' ' << v2;
```



swap 2 Передача параметров по значению

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

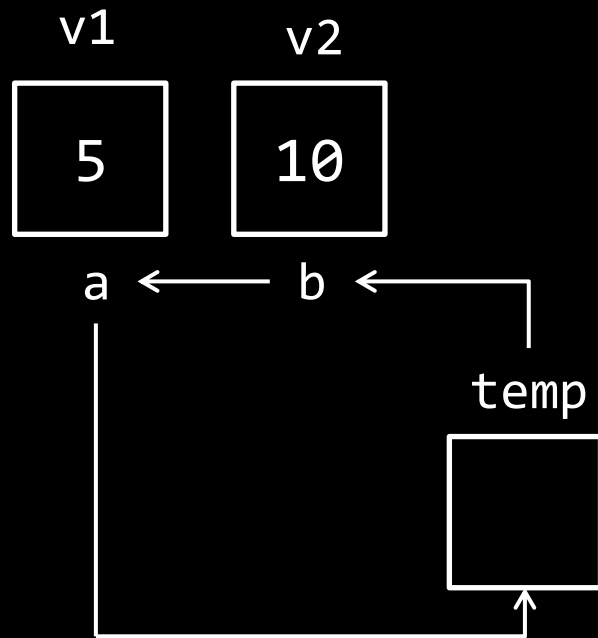
```
int v1 = 5, v2 = 10;  
swap(&v1, &v2);  
cout << v1 << ' ' << v2;
```



swap 3 Передача параметров по ссылке

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

```
int v1 = 5, v2 = 10;  
swap(v1, v2);  
cout << v1 << ' ' << v2;
```



```
void func(int i){  
    i = 10;  
}
```

```
void func_ref(int& i){  
    i = 10;  
}
```

```
int data = 0;  
func(data);  
func_ref(data);
```

```
void func(int i){  
    i = 10;  
}
```

```
void func_ref(int& i){  
    i = 10;  
}
```

```
int data = 0;  
func(data + 1);  
func_ref(data + 1); // Ошибка
```


Возврат ссылок

```
int max(int a, int b){  
    if (a>b) return a;  
    else b;  
}
```

```
int& max_ref(int& a, int& b){  
    if (a>b) return a;  
    else b;  
}
```

```
int v1 = 0, v2 = 10;  
cout << max(v1, v2);  
cout << max_ref(v1, v2);
```

```
int max(int a, int b){  
    if (a>b) return a;  
    else b;  
}
```

```
int& max_ref(int& a, int& b){  
    if (a>b) return a;  
    else b;  
}
```

```
int v1 = 0, v2 = 10;  
max(v1, v2) = 3; // Ошибка  
max_ref(v1, v2) = 3; // Изменит значение v2 на 3
```

```
int max(int a, int b){  
    if (a>b) return a;  
    else b;  
}
```

```
int& max_ref(int a, int b){  
    if (a>b) return a;  
    else b;  
}
```

```
int v1 = 0, v2 = 10;  
max(v1, v2) = 3; // Ошибка  
max_ref(v1, v2) = 3; // Ошибка, но может запуститься
```

Применение ссылок

- Внутри функции нужно изменить значение внешней переменной;
- Чтобы не копировать некоторый большой объект. Если нужно запретить его изменять используют константную ссылку;