

ПРОГРАММИРОВАНИЕ. Практика

Ссылки в C++



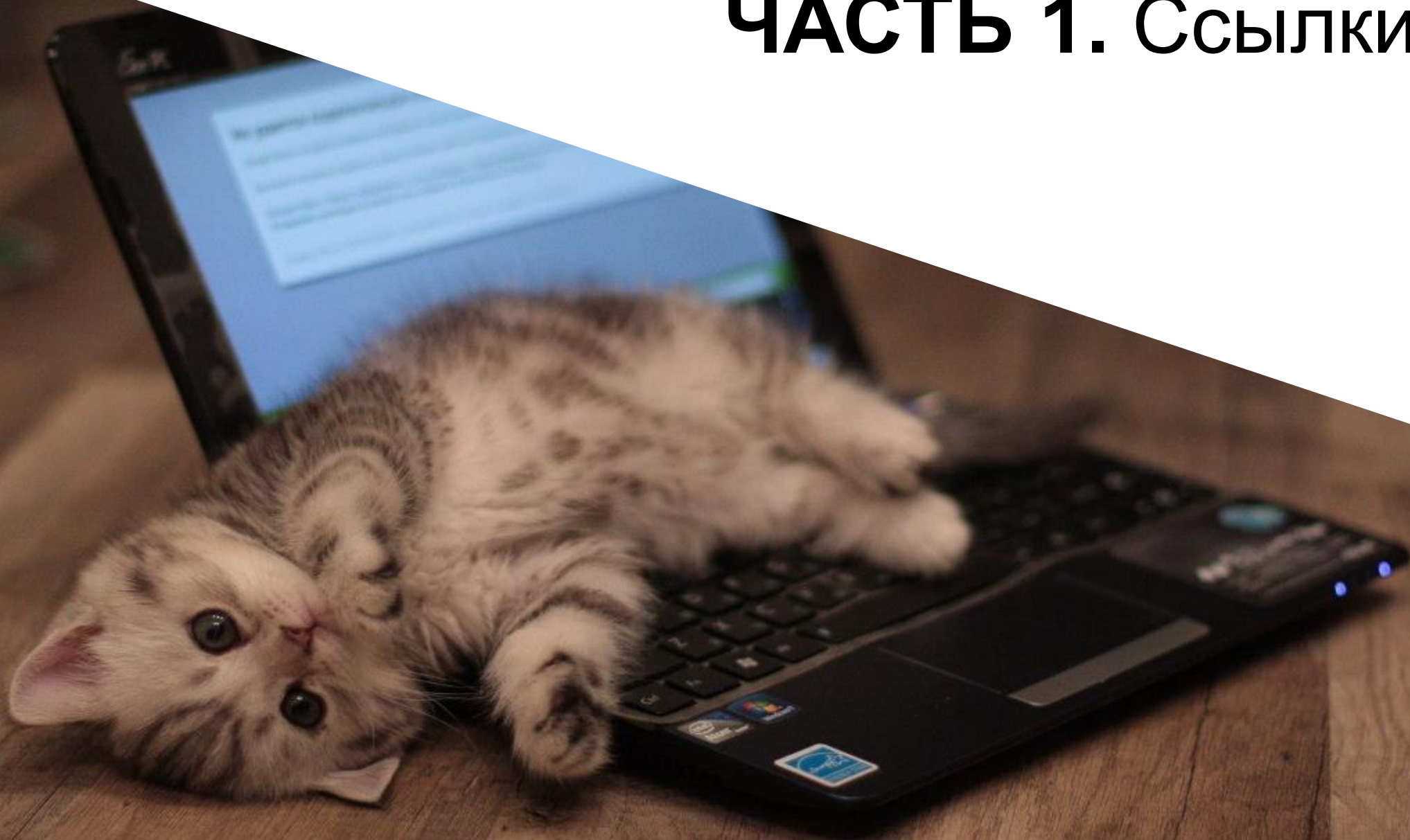
ФЕДОТЕНКО

Мария Александровна

 ma.fedotenko@mpgu.su

 [marusyafm](https://vk.com/marusyafm)

ЧАСТЬ 1. Ссылки



Ссылки в C++

Ссылка – тип данных, являющийся скрытой формой указателя, который разыменовывается автоматически в момент использования.



Именованние ссылок

Ссылки именуются по тем же правилам, что и другие переменные. Но традиционно имя ссылки содержит префикс ref («reference» - ссылка) и имя переменной, на которую она ссылается.

```
int someNumber = 10;  
int &refSomeNumber = someNumber;
```

При этом тип данных ссылки должен совпадать с типом данных переменной, на которую она ссылается

```
int someNumber = 10;  
int &refSomeNumber = someNumber; // Ок  
double &refSomeNumber = someNumber; // Не ок, разные типы данных
```


Инициализация ссылок

Ссылка обязательно должна быть инициализирована при объявлении.

Объявление ссылки без указания значения – не работает!

```
int someNumber = 10;  
int &refSomeNumber; // Не ок, ссылка не инициализирована  
int &refSomeNumber = 50; // Тоже не ок, литеральные значения недопустимы  
int &refSomeNumber = someNumber; // Ок, ссылка инициализирована значением переменной
```

Как работают ссылки



```
int someNumber = 10;
cout << "Значение переменной: " << someNumber << endl;
int &refSomeNumber = someNumber;
cout << "Значение ссылки (адрес): " << &refSomeNumber << endl;
cout << "Значение по ссылке: " << refSomeNumber << endl;
```

✗ Output

```
Значение переменной: 10
Значение ссылки (адрес): 0x7ffe54c392a4
Значение по ссылке: 10
```

Изменение значений



Если поменять значение ссылки – изменится и значение переменной

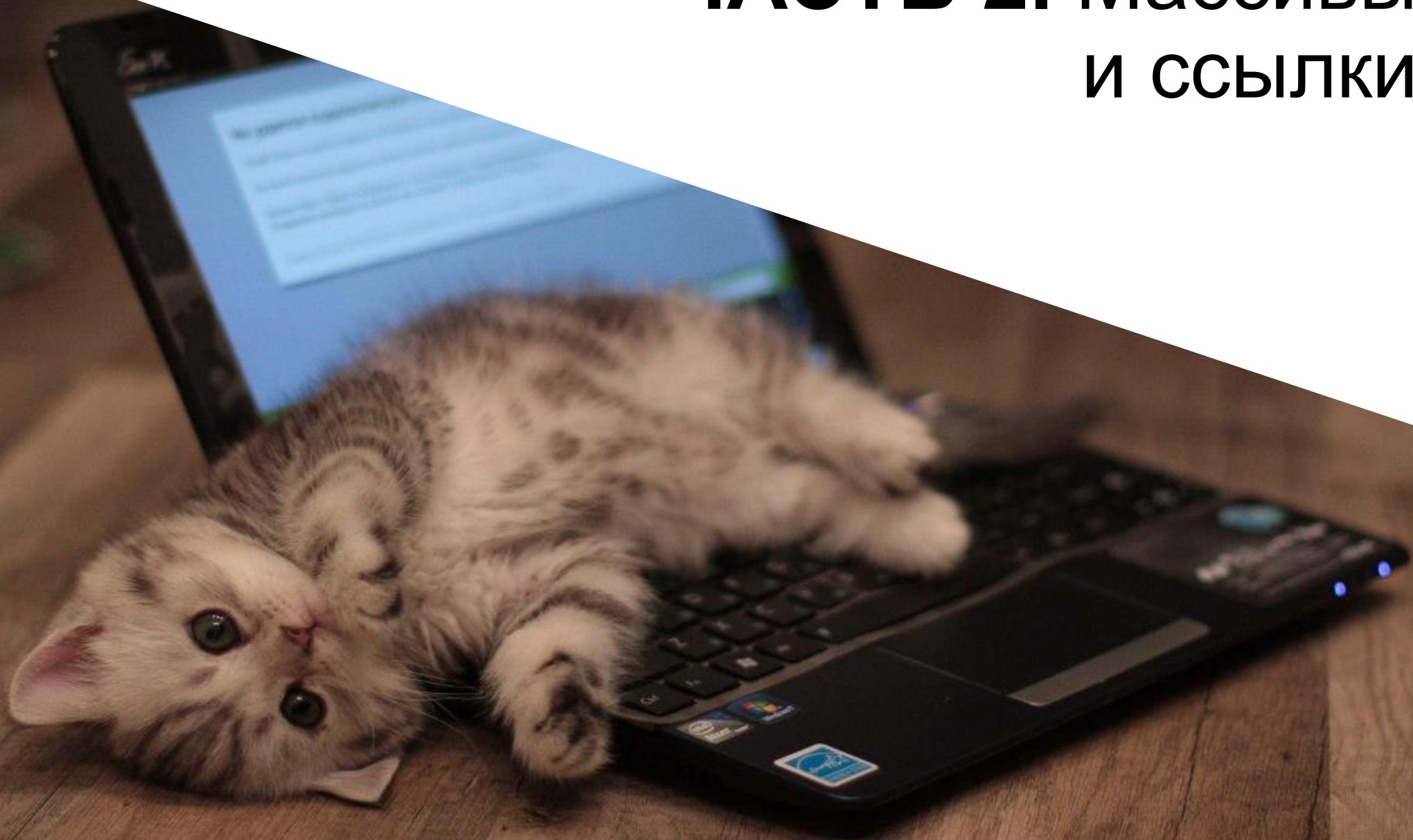
```
int someNumber = 10;  
cout << "Значение переменной: " << someNumber << endl;  
int &refSomeNumber = someNumber;  
refSomeNumber = 20;  
cout << "Значение, измененное через ссылку: " << someNumber << endl;
```

× Output

Значение переменной: 10

Значение, измененное через ссылку: 20

ЧАСТЬ 2. Массивы и ССЫЛКИ





Массивы и ссылки

По стандарту языка C++ ссылки не могут ссылаться на ссылки, а также запрещено создание массива ссылок.

Таким образом, при необходимости ссылки могут ссылаться только на конкретный элемент массива.

```
int n = 5;  
int arr[n] = {1,2,3,4,5};  
for (int i=0; i<n; i++){  
    cout << arr[i] << "\\t";  
}
```

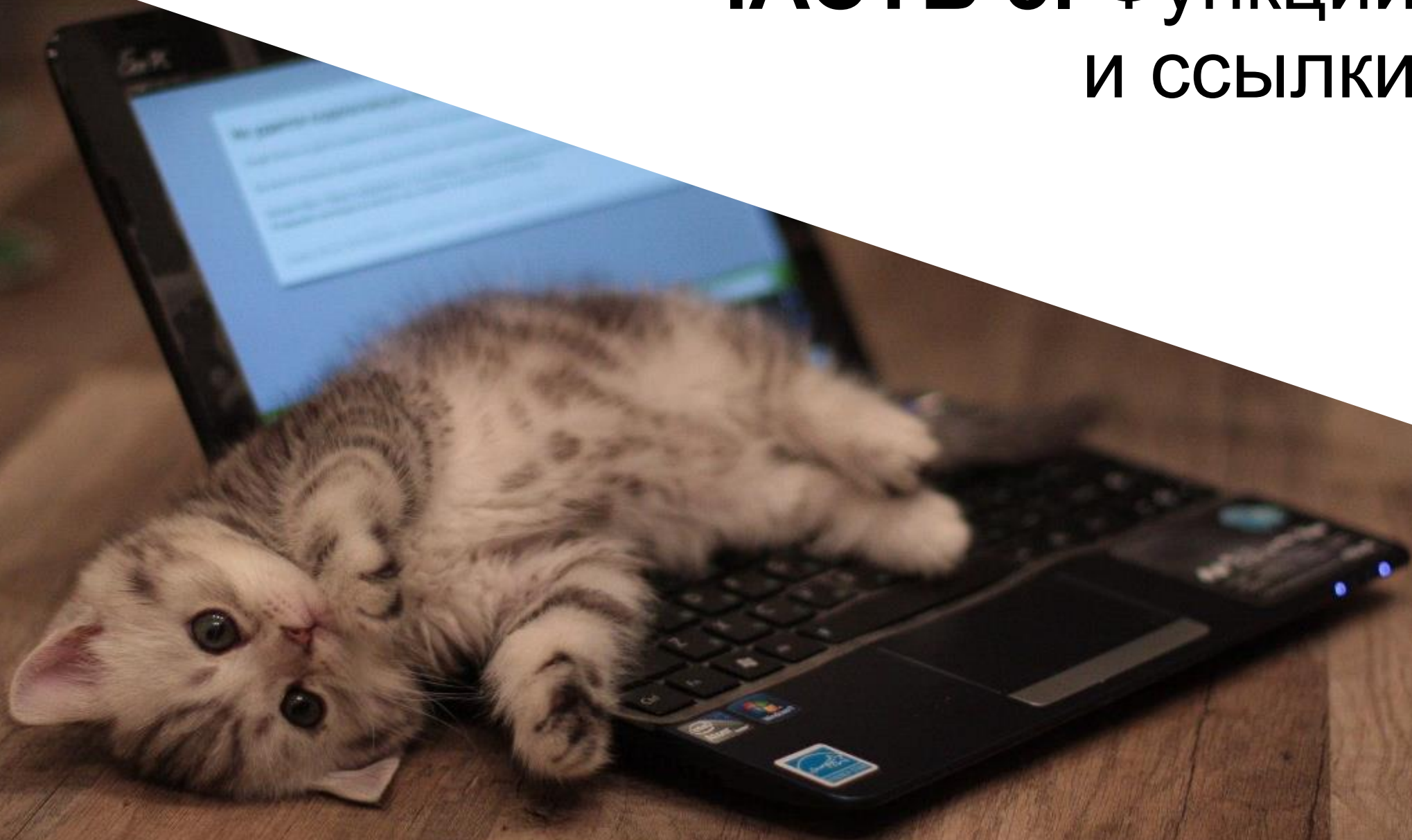
// Ссылка

```
int &refArr = arr[3];  
cout << endl << "Элемент по ссылке: " << refArr;
```

× Output

1	2	3	4	5
Элемент по ссылке: 4				

ЧАСТЬ 3. Функции и ССЫЛКИ



Вспомним про область видимости

```
▼ void someFunction(int someNumber) {  
    someNumber++;  
    cout << "Значение переменной В МОМЕНТ исполнения функции: " << someNumber << endl;  
}  
▼ int main() {  
    int someNumber = 120;  
    cout << "Значение переменной ДО вызова функции: " << someNumber << endl;  
    someFunction(someNumber);  
    cout << "Значение переменной ПОСЛЕ вызова функции: " << someNumber << endl;  
}
```



✖ Output

```
Значение переменной ДО вызова функции: 120  
Значение переменной В МОМЕНТ исполнения функции: 121  
Значение переменной ПОСЛЕ вызова функции: 120
```

Передача переменных в функцию



```
1 ▾ #include <iostream>
2  using namespace std;
3  // Эта программа показывает 3 способа передачи переменных в функцию:
4  int sumByValue(int); // По значению
5  int sumByReference(int &); // По ссылке
6  int sumByPointer(int *); // По указателю
7
```

Передача переменных в функцию



```
20 ▾ int sumByValue(int someNumber){
21     someNumber += someNumber;
22     cout << "Передача в функцию по значению: " << someNumber << endl;
23     return someNumber;
24 }
25
26 ▾ int sumByReference(int &refSomeNumber){
27     refSomeNumber += refSomeNumber;
28     cout << "Передача в функцию по ссылке: " << refSomeNumber << endl;
29     return refSomeNumber;
30 }
31
32 ▾ int sumByPointer(int *pSomeNumber){
33     *pSomeNumber += *pSomeNumber;
34     cout << "Передача в функцию по указателю: " << *pSomeNumber << endl;
35     return *pSomeNumber;
36 }
```


Передача переменных в функцию

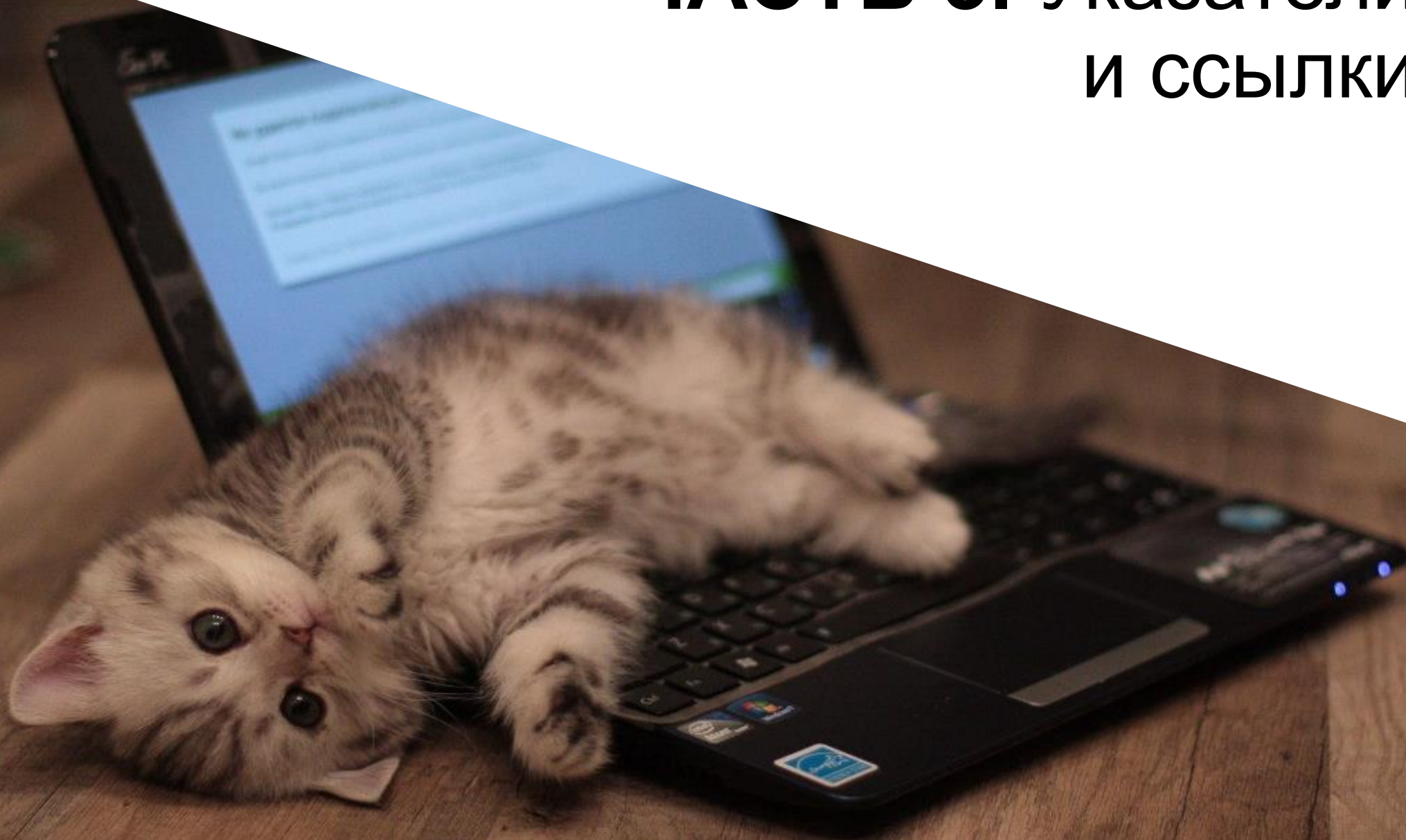


```
8  int main()
9  {
10     int someNumber = 10;
11     cout << "Само число: " << someNumber << endl;
12     sumByValue(someNumber);
13     cout << "Вернулось значение: " << someNumber << endl;
14     sumByReference(someNumber);
15     cout << "Вернулось значение: " << someNumber << endl;
16     sumByPointer(&someNumber);
17     cout << "Вернулось значение: " << someNumber << endl;
18 }
```

✕ Output

Само число: 10
Передача в функцию по значению: 20
Вернулось значение: 10
Передача в функцию по ссылке: 20
Вернулось значение: 20
Передача в функцию по указателю: 40
Вернулось значение: 40

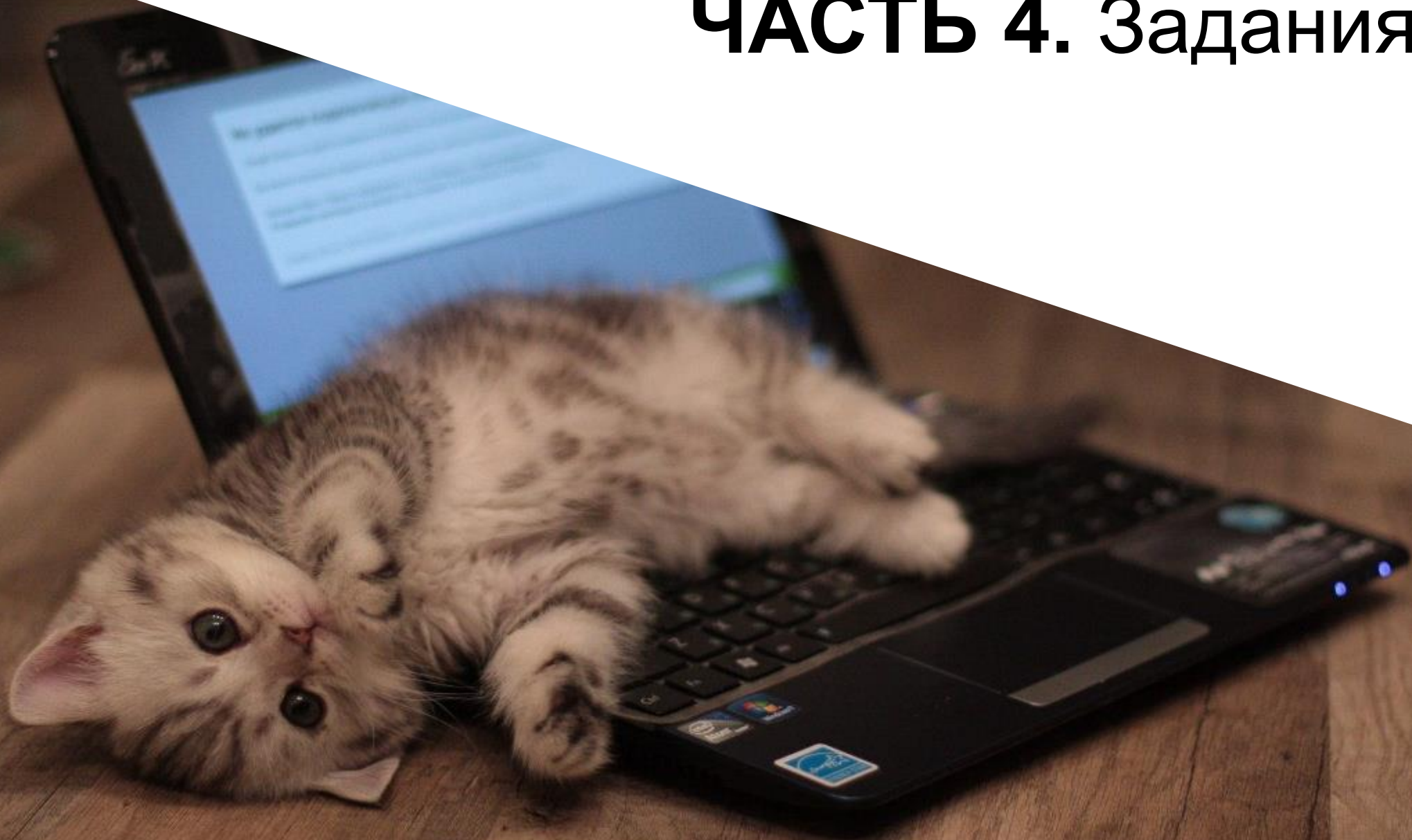
ЧАСТЬ 3. Указатели и ссылки



Указатели и ссылки (разница)

1. Поскольку ссылки должны быть инициализированы корректными объектами, они безопаснее указателей (так как риск разыменования нулевого указателя отпадает).
2. Ссылки более ограничены в функциональности.
3. Если определенное задание может быть решено с помощью и ссылок, и указателей, лучше использовать ссылки. Указатели лучше использовать при динамическом выделении памяти.
4. Возврат из функции ссылки на автоматически созданный объект (локальную переменную) приводит к появлению "битых ссылок", значение которых непредсказуемо.
5. При обращении к ссылке разыменование происходит автоматически.

ЧАСТЬ 4. Задания



Задание 1



(Вариант выбираем соответственно номеру в списке группы)

Вариант 1: Запросить у пользователя консольный ввод 2х целых чисел. Создать для них ссылки. С помощью ссылок сравнить их, затем возвести меньшее в квадрат, а большему присвоить значение среднего арифметического его цифр.

Вариант 2: Запросить у пользователя консольный ввод 2х целых чисел. Создать для них ссылки. С помощью ссылок сравнить их и большее возвести в куб, а меньшее уменьшить на первую цифру большего.

Задание 2



Выполнить Задание 1 **другого варианта.**

1. Реализовать вычисления в **отдельной функции, передавая переменные по значению.**
2. Реализовать вычисления в **отдельной функции, передавая переменные по ссылке.**
3. Реализовать вычисления в **отдельной функции, передавая переменные по указателю.**

Таким образом, в программе должно быть реализовано 3 функции (не считая main).

Благодарю за внимание!

