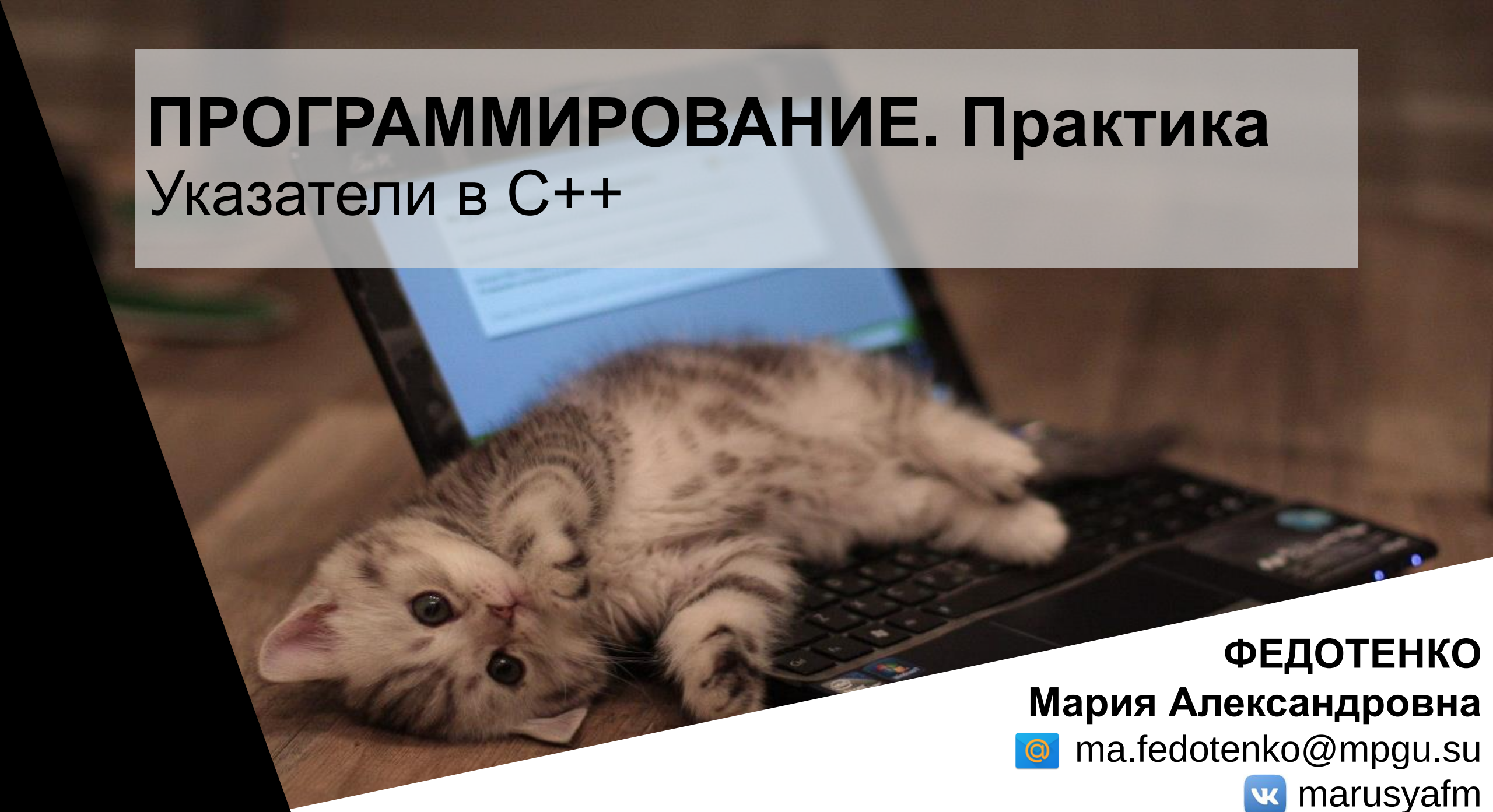
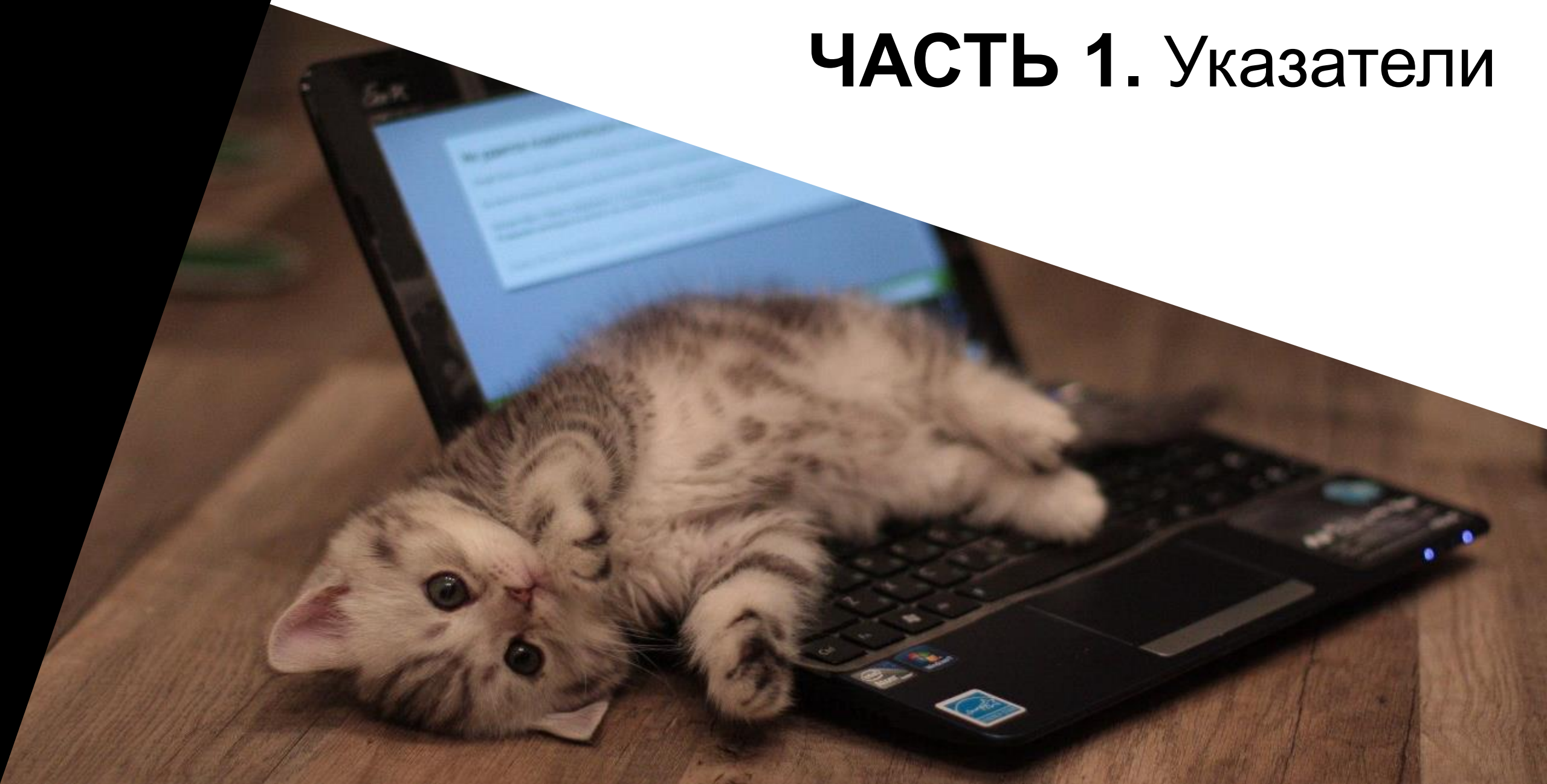
ПРОГРАММИРОВАНИЕ. Практика

Указатели в C++



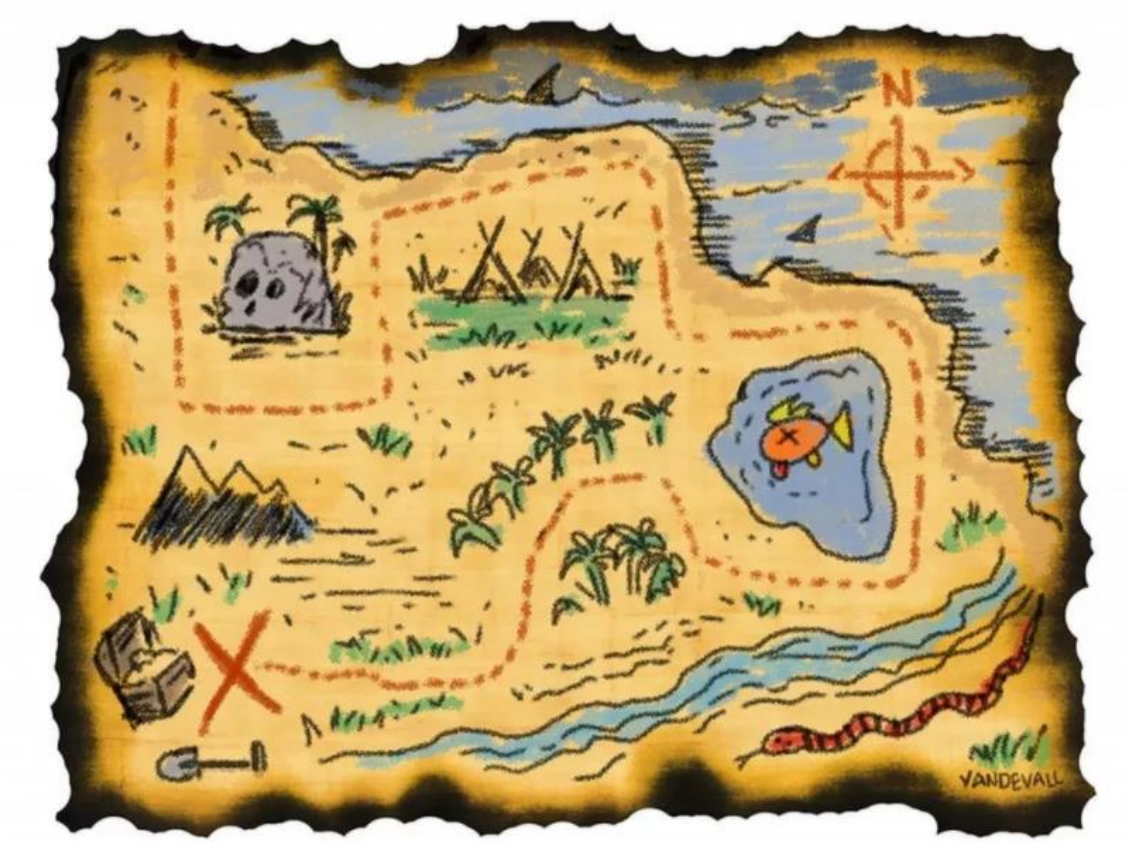
ФЕДОТЕНКО
Мария Александровна
ma.fedotenko@mpgu.su
marusyafm

ЧАСТЬ 1. Указатели



Указатель

Указатель – переменная, содержащая адрес ячейки памяти другой переменной.



Именование указателей

Указатели именуются по тем же правилам, что и другие переменные. Но традиционно имя указателя содержит префикс p («pointer» - указатель) и имя переменной, на которую он указывает.

```
int someNumber = 15; // Любая переменная
int *pSomeNumber = &someNumber; // Указатель на эту переменную (& - адрес)
```

При этом тип данных указателя должен совпадать с типом данных переменной, на которую он указывает

```
int someNumber = 15;
int *pSomeNumber = &someNumber; // Ок, один тип данных
double *pSomeNumber = &someNumber; // НЕ ок, разные типы данных
```

Как работают указатели

```cpp
int someNumber = 15;
int *pSomeNumber = &someNumber;
// Посмотрим как это работает
cout << "Значение переменной: " << someNumber << endl;
cout << "Значение указателя: " << pSomeNumber << endl;
cout << "Значение переменной под указателем: " << *pSomeNumber << endl;
```

Output

```
Значение переменной: 15
Значение указателя: 0x7ffff2b51264
Значение переменной под указателем: 15
```

Операция разыменования

Пишем звездочку – получаем значение

pSomeNumber – получаем адрес ячейки памяти
*pSomeNumber – получаем значение переменной – это и есть **операция разыменования**

Изменение значений

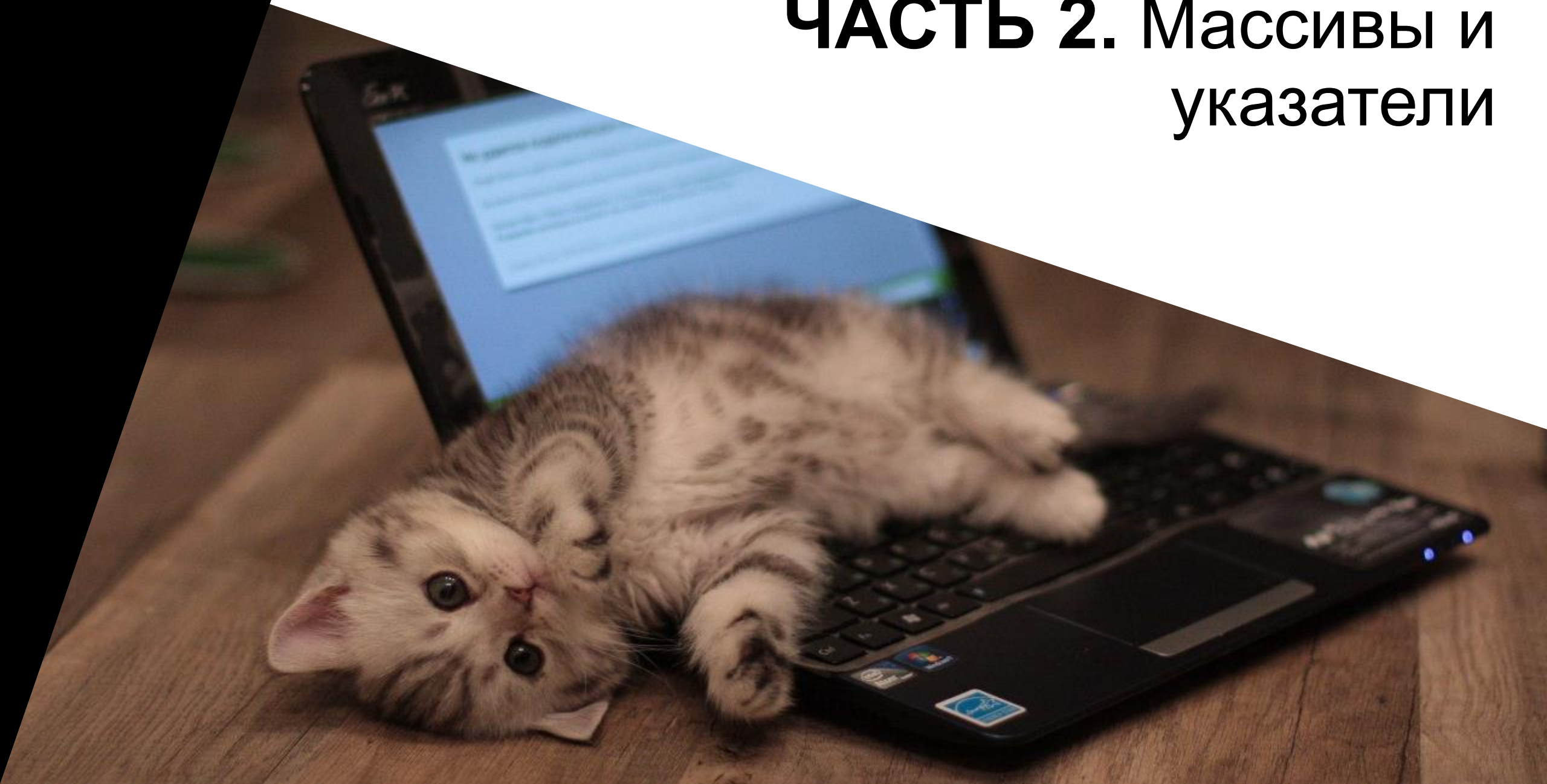
Если поменять значение указателя – изменится и значение переменной

```
int someNumber = 15;
cout << "Значение переменной: " << someNumber << endl;
int *pSomeNumber = &someNumber;
*pSomeNumber = 2;
cout << "Значение, измененное через указатель: " << someNumber << endl;
```

Output

```
Значение переменной: 15
Значение, измененное через указатель: 2
```

ЧАСТЬ 2. Массивы и указатели



Массивы и указатели

```cpp
int arrSize = 5;
int someArr[arrSize] = {1, 2, 3, 4, 5};
cout << "Простой вывод массива:" << endl;
for (int i=0; i<arrSize; i++){
    cout << someArr[i] << "\t";
}
cout << endl << "А теперь через указатели:" << endl;
int *pSomeArr = someArr; // Указатель на массив
for (int i=0; i<arrSize; i++){
    cout << pSomeArr[i] << "\t"; // Разыменование в такой форме записи не нужно
}
```

Output

```
Простой вывод массива:
1       2       3       4       5
А теперь через указатели:
1       2       3       4       5
```

Массивы и указатели

```
// Имя массива - указатель на его первый элемент
int arrSize = 5;
int someArr[arrSize] = {1, 2, 3, 4, 5};
int *pSomeArr = someArr;
cout << someArr << endl;
cout << pSomeArr << endl;
```

Output

```
0x7fff21034da0
0x7fff21034da0
```

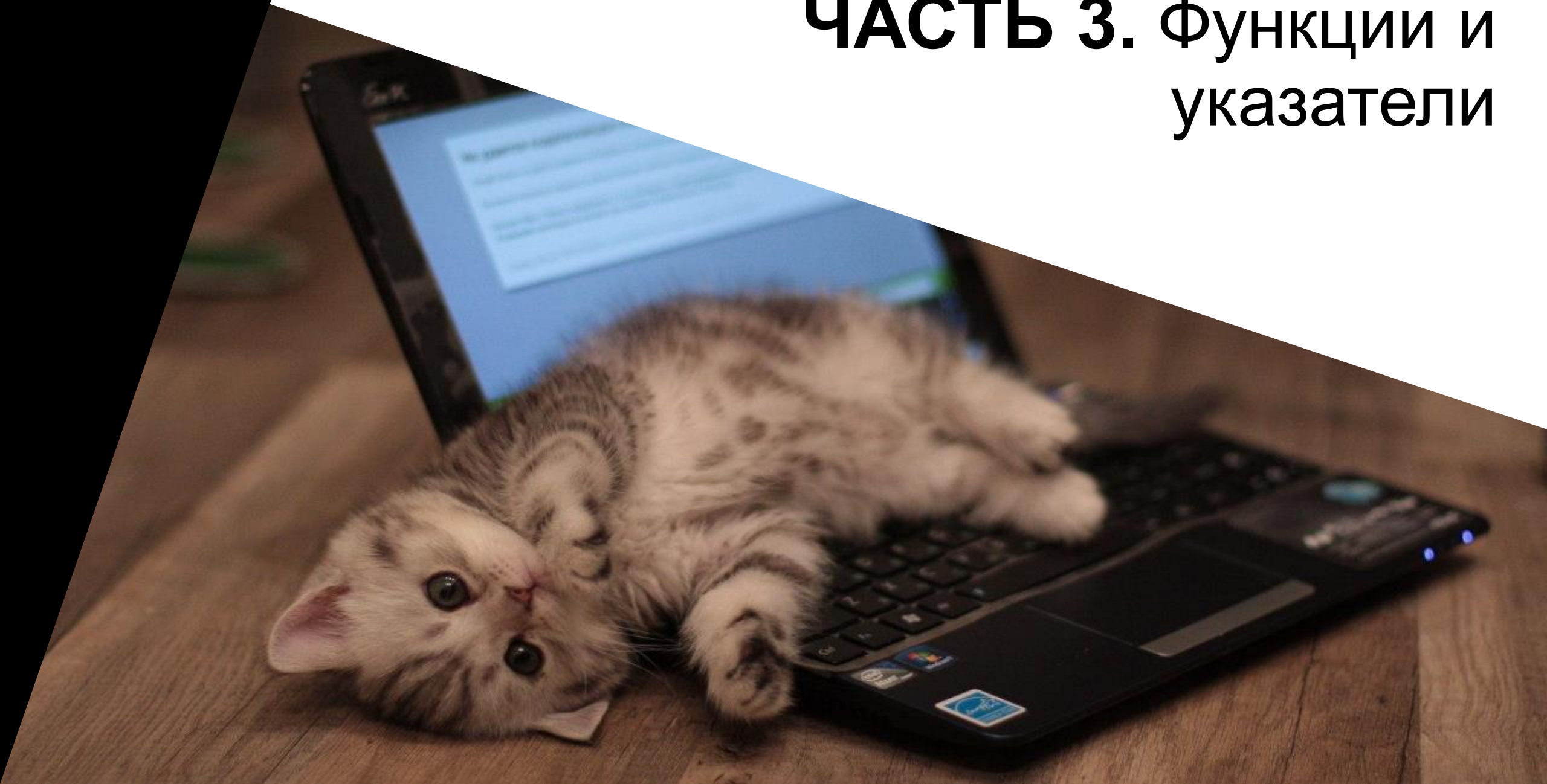
Арифметика указателей

```
int arrSize = 5;
int someArr[arrSize] = {1, 2, 3, 4, 5};
int *pSomeArr = someArr;
cout << "Арифметика указателей:" << endl;
cout << *someArr << endl; // Первый элемент массива
cout << *(someArr+1) << endl; // Второй элемент массива
// А вот в такой форме записи уже используется разыменование
```

Output

```
Арифметика указателей:
1
2
```

ЧАСТЬ 3. Функции и указатели



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
```

Добавим указатели

```
void someFunction(int *pSomeNumber) {
    (*pSomeNumber)++; // Разыменование, а затем инкремент
    cout << "Значение переменной В МОМЕНТ исполнения функции: " << *pSomeNumber << 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: 121
```

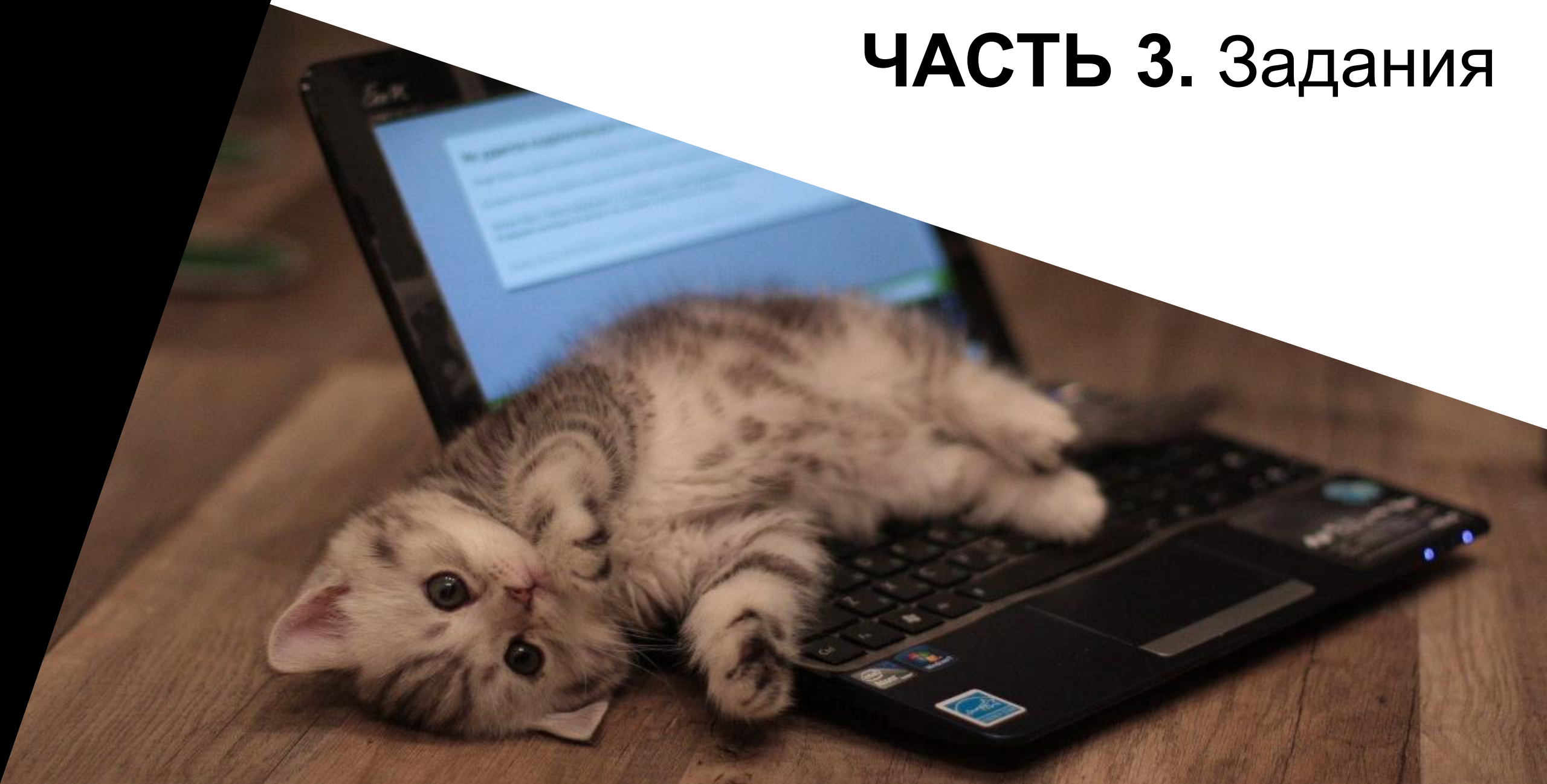
Передача массива в функцию

```
void print(int*);
int main()
{
    int someArr[] = {1, 2, 3, 4, 5};
    print(someArr);
}
void print(int *pSomeArr)
{
    int arrSize = sizeof(*pSomeArr);
    cout << "Получившийся массив: " << endl;
    for (int i=0; i<=arrSize; i++){
        cout <<  pSomeArr[i] << "\t";
    }
}
```

Output

```
Получившийся массив:
1       2       3       4       5
```

ЧАСТЬ 3. Задания



Задание 1

(Вариант выбираем соответственно номеру в списке группы)

Вариант 1: Запросить у пользователя консольный ввод 2х целых чисел. Создать для них указатели. С помощью указателей сравнить их, затем увеличить меньшее в 2 раза, а большему присвоить значение первого+1.

Вариант 2: Запросить у пользователя консольный ввод 2х целых чисел. Создать для них указатели. С помощью указателей сравнить их и большее увеличить в 5 раз, а меньшее уменьшить на 5.

Задание 2

Выполнить Задание 1 **другого варианта**.

Вычисления реализовать **в отдельной функции**.

Задание 3

Взять любую написанную ранее программу, содержащую обработку массивов через функции, и модифицировать ее с использованием указателей.

Благодарю за внимание!