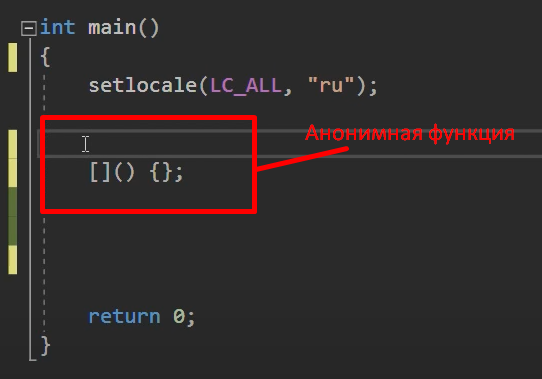
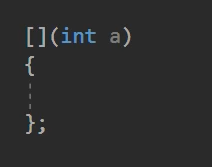
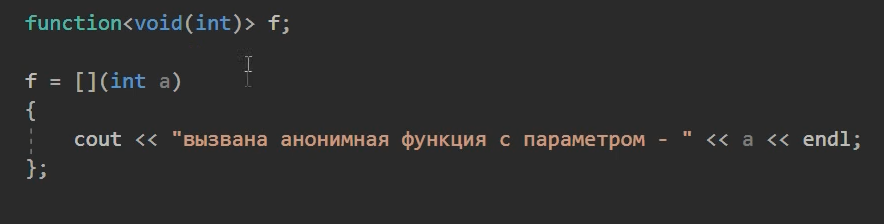
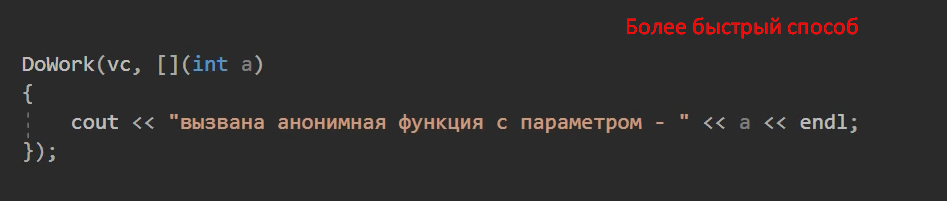
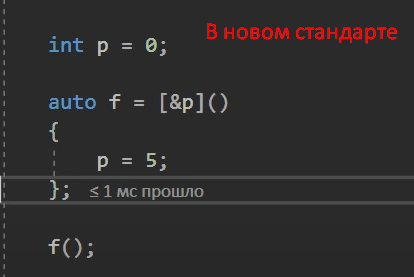
std::функция | Полиморфная оболочка функции(указатель на функцию из мира ООП):

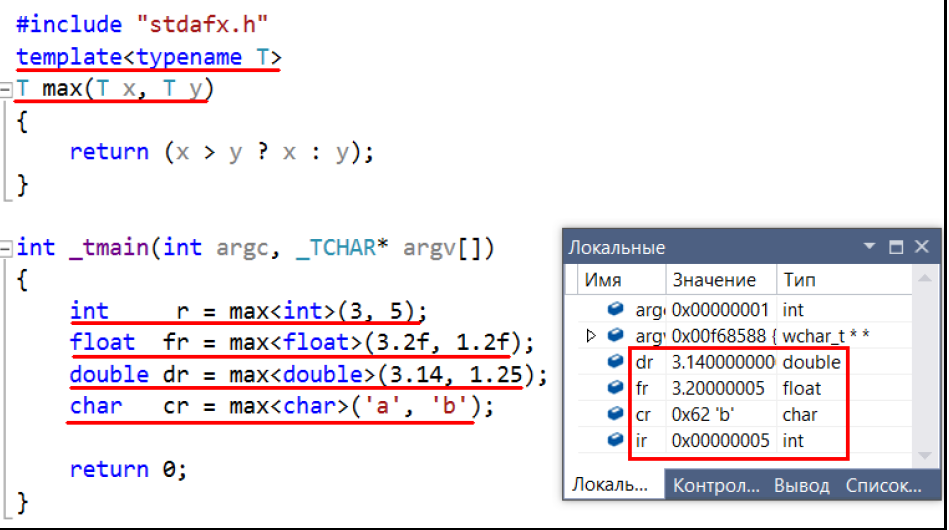
Лямбда-функция:









Шаблоны функций:

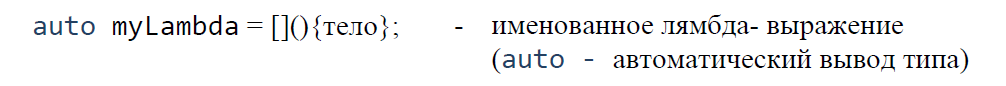
Ответы на вопросы:

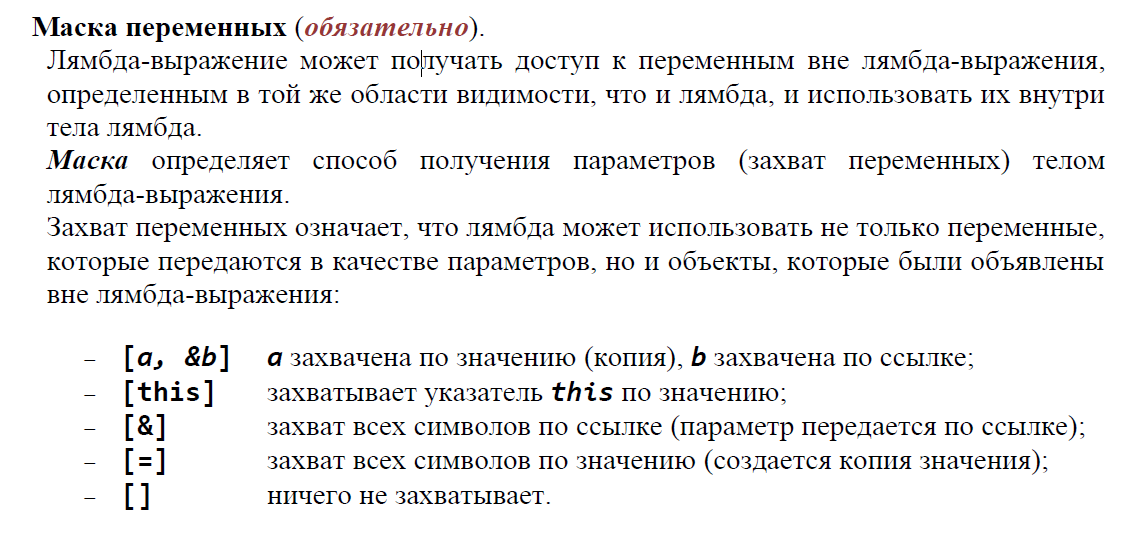
1) **Лямбда-выражение**— типичная для многих языков синтаксическая конструкция для определения анонимной функции.

**Анонимная функция**– особый вид функций, которые объявляются в месте использования и не получают уникального идентификатора для доступа к ним.

Слово «лямбда» или «лямбда-выражения» чаще всего относится к анонимным функциям. Таким образом, в этом смысле лямбда является своего рода функцией, но не каждая функция является лямбда (то есть именованные функции обычно не называют лямбда-выражениями).

2) Основная причина применения **лямбда** **функций** — создание функции, которая **используется** в коде единожды. **Использование** таких функций позволяет снизить число строк кода, которые вам придется написать.

3)

Где [] – 

() – список параметров(аргументов);

{} – тело функции;

!Перед телом возможно указать ключевое слово mutable (**позволяет модифицировать** параметры, переданные по значению);

!!После тела в круглых скобках можно явно указать значения параметров.

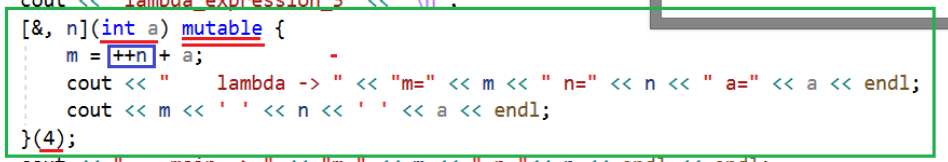
4) Захват переменных:

[] не только определяет (вводит) лямбду, но также содержит список захваченных переменных (может быть пустым). Это называется «список захвата».

Захватив переменную, лямбда создает член-копию этой переменной в типе замыкания. Затем внутри тела лямбды можно получить к этой копии доступ.

5) По значению и по ссылке (==3 вопрос)

6) (==3 вопрос), пример:



**n** передали по значению -> мы не можем её изменять, т.к. тип замыкания является константным, и нельзя изменять захваченные переменные внутри тела лямбда-выражения.

Чтобы изменить это поведение, нужно добавить ключевое слово mutable после списка параметров.