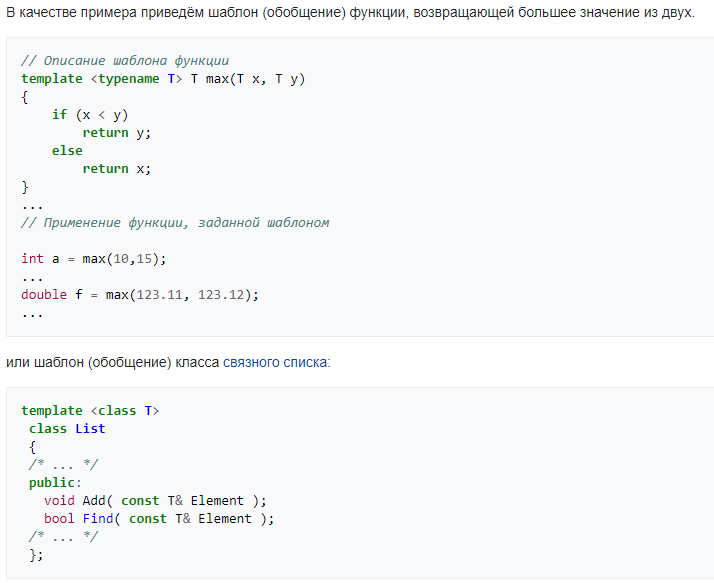
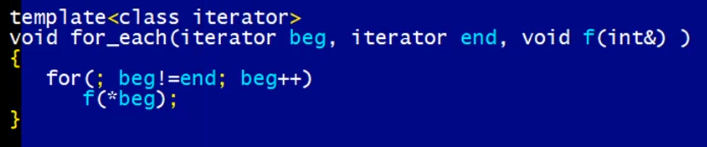
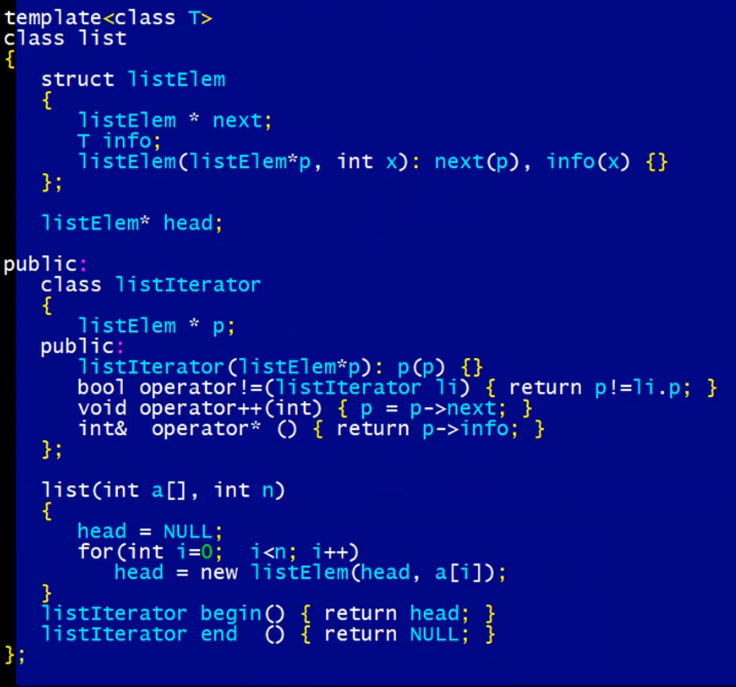
**Обобщённое программирование** (англ. generic programming) — парадигма программирования, идея которого состоит в написании алгоритмов в терминах абстрактных типов данных; когда алгоритм используется для конкретных типов данных, создается экземпляр этого алгоритма с типами данных, переданными в качестве параметров. Такой стиль программирования позволяет использовать универсальный код для похожих заданий, имеющих дело с разными типами данных, и таким образом уменьшить дублирование кода.

Обобщённое программирование широко используется для реализации универсальных контейнеров и алгоритмов. Так, стандартная библиотека шаблонов STL в С++ предоставляет набор контейнеров (динамический массив, связный список, очередь, множество и т.д.) и алгоритмов, применимых к этим или пользовательским контейнерам.

В основе обобщенного программирования лежит использование так называемых шаблонов функций и шаблонов классов. Применение таких шаблонов связано с тем обстоятельством, что алгоритмы обработки данных часто слабо зависят от типа данных, которые они обрабатывают. Информацию о типе данных, которые следует обрабатывать, в этом случае удобно передавать через параметры, которые носят название обобщенных типов.



**Итератор** — это объект, разработанный специально для перебора элементов контейнера (например, значений массива или символов в строке), обеспечивающий во время перемещения по элементам доступ к каждому из них.

Контейнер может предоставлять различные типы итераторов. Например, контейнер на основе массива может предлагать прямой итератор, который проходится по массиву в прямом порядке, и реверсивный итератор, который проходится по массиву в обратном порядке.

В стандартной библиотеке С++ началом последовательности или диапазона является первый элемент. Конец последовательности или диапазона всегда определяется как элемент, следующий за последним элементом. Глобальные функции begin и end возвращают итераторы в указанный контейнер.

Основные требования к итераторам — наличие операторов разыменования и инкремента.

