# **КОНТРОЛЬНЫЕ ВОПРОСЫ.**

## Как организован механизм генерации случайных чисел в библиотеке random?

В библиотеке random присутствуют 16 генераторов: объектов, которые, будучи инициализированы некоторым значением, т.н. seed-ом, генерируют случайные величины по заложенному алгоритму, которые затем передаются объекту–распределению, преобразовывающему их с некоторым распределением (своим для каждого класса р.) по заданному интервалу. Таким образом пользователь может получить случайные значения, подчиняющиеся некоторому распределению, например, нормальному р. или р. Пуассона.

## Чем отличаются функциональные объекты от функций и лямбда-выражений?

Объекты обладают как поведением, так и состоянием, т.е. могут хранить информацию.\* Разные объекты имеют разные типы. Передача объекта в функцию по значению зачастую быстрее, чем передача функции, которая производится по указателю.

\*Функции также могут хранить информацию (а лямбды – захватывать), но хранение информации в функциях может быть достигнуто за счёт использования статики. Существует мнение, что использования ключевого слова static следует избегать, так как это противоречит концепции ООП.

## Когда алгоритмам стандартной библиотеки требуются итераторы вставки?

Алгоритмы стандартной библиотеки используют итераторы вставки в тех случаях, когда нет гарантии в том, что коллекция обладает достаточным размером для применения требуемой операции. Использование итераторов вставки позволяет увеличивать размер контейнера до необходимого в процессе вставки в него элементов посредством вызова соответствующих методов, например, push\_front или insert.

## Какая классификация\* предлагается для алгоритмов стандартной библиотеки?

Немодифицирующие алгоритмы: for\_each (немод. версия), count, find, search и др.

Модифицирующие алгоритмы: for\_each, copy, move, transform, fill, generate, replace и др., в т.ч. алгоритмы удаления: remove, unique\*\* (на самом деле не удаляют элементы, а осуществляют необходимые перестановки для применения метода erase).  
\*\* удаляет *подряд* идущие дубликаты

Алгоритмы перестановки: reverse, rotate, shuffle.

Алгоритмы сортировки: sort(комбинированный алгоритм), stable\_sort(сохранение порядка одинаковых элементов), partial\_sort (частичная сортировка, гарантия O(NlogN)), partition (сортировка на кластеры) и др.

Алгоритмы для работы с упорядоченными диапазонами: алгоритмы двоичного поиска: lower\_bound, equal\_range и др.; алгоритмы множеств: intersection, union, difference и др.

Численные алгоритмы: accumulate, partial\_sum и др.

\* Указана классификация, предложенная на семинаре, есть несколько иная классификация, отдельно выделяющая:

Алгоритмы разделения (partition).

Алгоритмы поиска экстремальных элементов.

Алгоритмы сравнения.

Алгоритмы перестановки (permutation) — не содержат reverse, rotate и shuffle, а последние отнесены в модифицирующие.

## Почему алгоритмы стандартной библиотеки предпочтительнее собственных циклов?

Алгоритмы стандартной библиотеки написаны опытными программистами и многократно протестированы, их использование упрощает код и способствует его читаемости: другие программисты, знакомые со стандартными алгоритмами, могут быстрее разобраться в нём.