

## №1

В библиотеке `random` генерируются псевдослучайные числа используя некоторую последовательность, которая встроена в систему, но пользователь про нее не знает (то есть используется некоторый детерминированный алгоритм, в котором вероятность не играет роли). Из данной последовательности выбирается произвольное число и далее просто выводятся элементы этой последовательности при следующем вызове.

`Seed` – отправная точка для того, чтобы выводить числа, то есть по `seed` выбирается начальный элемент последовательности. Для отладки кода удобно использовать какой-то фиксированный `seed`, чтобы исключить случайность, а если нужно в каждый момент времени начинать отсчет с нового элемента, то используют `seed(time(nullptr))`.

Генератор – это объект, который образует последовательность, из которой берутся элементы (например `mt19937_64`). Распределение – некоторый объект, который определенными преобразованиями приводит последовательность чисел из генератора к нужному распределению, например `std::uniform_int_distribution<int> distr(0, 150)`; преобразует элементы к равномерному распределению на отрезке 0, 150.

## №2

Итератор – это объект, который позволяет обращаться к элементам STL контейнера, не зная при этом его реализации. Например, в контейнере `list` итератор является основным способом для получения доступа к элементам списка.

Диапазон – пара итераторов, первый из которых является началом диапазона, а второй концом. При этом первый итератор указывает на элемент, который лежит раньше второго. Стандартные алгоритмы работают с итераторами, а не напрямую с контейнерами, так как алгоритмы универсальны и подходят для любого контейнера, у которого есть свой итератор. Поэтому не нужно делать новые алгоритмы для каждого нового контейнера. И, более того, можно писать собственные контейнеры с итераторами, к которым будут применимы алгоритмы.