

Контрольные вопросы:

- ☐ (5 б.) В каких ситуациях применяются типы `std::pair` и `std::tuple`?
- ☐ (5 б.) Когда следует использовать контейнер `std::array`?
- ☐ (5 б.) Когда следует использовать контейнер `std::vector`?
- ☐ (5 б.) Когда следует использовать контейнер `std::deque`?
- ☐ (5 б.) Когда следует использовать контейнер `std::list`?
- ☐ (5 б.) Когда следует использовать контейнер `std::forward_list`?
- ☐ (5 б.) Какие адаптеры контейнеров есть в стандартной библиотеке?
- ☐ (5 б.) Когда следует использовать контейнер `circular buffer` из Boost?
- ☐ (5 б.) Почему контейнер `circular buffer` из Boost не может войти в стандарт?
- ☐ (5 б.) Какие типы данных для работы с многомерными массивами вы можете назвать?

Упражнения:

- ☐ (25 б.) Проанализируйте систему выделения памяти в векторе. Для этого определите с помощью вызовов функции-члена `capacity`, во сколько раз изменяется емкость вектора при нехватке памяти для размещения новых элементов. Также определите, как увеличивается емкость вектора, если задать ее начальное значение вручную с помощью вызова функции-члена `reserve`. Дополнительно определите, как осуществляется выделение памяти в предельном случае, когда вектор уже запросил большой объем памяти и на выполнение следующего запроса у ОС может не хватить ресурсов. В комментариях в коде опишите все свои наблюдения.
- ☐ (25 б.) Одно из основных действий над последовательностями – их сортировка. Определите время сортировки для каждого из пяти последовательных контейнеров. Выполните сортировку как с помощью соответствующей функции-члена (если таковая имеется у контейнера), так и с помощью алгоритма `std::sort`. Используйте ваш хронометр на базе идиомы RAII. Обеспечьте контейнеры одинаковыми наборами данных и выполняйте измерения в равных условиях. Составьте таблицу с результатами измерений и выявите лидера.
- ☐ (25 б.) Реализуйте тело шаблона функции `fill_multi_array`, заполняющего многомерный массив типа `boost::multi_array` значениями из многомерного контейнера типа `std::vector`. Заготовка в приложении.