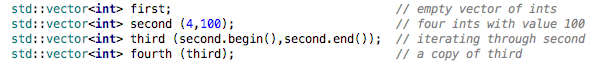
Vector

1. Какие виды векторов позволяют создать конструкторы?

* Пустой
* С n элементами, где каждый элемент - это копия заданного значения
* Вектор, заполненный значениями из диапазона [first, last)
* Копию вектора



2. Как задать размер контейнера?



3. Как определить потенциальный размер контейнера?



4. Как получить доступ к элементу контейнера?

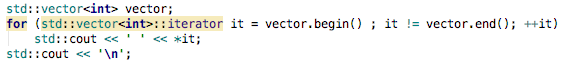


5. Как присвоить значение элементу контейнера?



6. Опишите функциональность методов begin и end

* begin - возвращает итератор, указывающий на первый элемент вектора
* end - возвращает итератор, относящийся к “после последнему” теоретическому элементу вектора



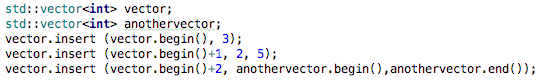
7. Опишите использование метода back и front

* front- возвразает ссылку на первый элемент
* back - возвразает ссылку на последний элемент



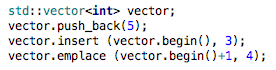
8. Опишите использование метода insert

* используется для вставки новых элементов (нового элемента) перед элементом в указанной позиции



9. Как и куда можно вставить новые элементы в контейнер?

* в любое место контейнера с помощью push\_back, insert, emplace



10. Опишите функциональность методов capacity и max\_size

* capacity - возвращает потенциальный размер вектора
* size - возвращает количество элементов в векторе



11. Зачем нужны методы reserve и resize?

* reserve - измеяет capacity
* resize - изменяте размер вектора



12. Как вставит/удалить элемент в начало контейнера?



13. Как вставит/удалить элемент в конец контейнера



Deque

14. Какие виды двусторонних очередей позволяют создать конструкторы?

* Пустая
* С n элементами, где каждый элемент - это копия заданного значения
* Очередь, заполненная значениями из диапазона [first, last)
* Копию очереди



15. Как получить доступ к элементу контейнера?



16. Как присвоить значение элементу контейнера?



17. Опишите функциональность методов front и back

* front- возвразает ссылку на первый элемент
* back - возвразает ссылку на последний элемент



18. Как вставит/удалить элемент в начало контейнера?



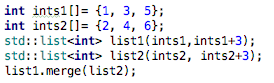
19. Как вставит/удалить элемент в конец контейнера



List

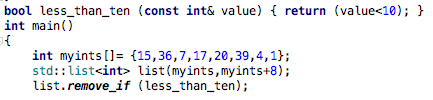
20. Опишите использование метода merge

* A.merge(B) - перемещает все элементы из B в A на соответствующие упорядоченные позиции



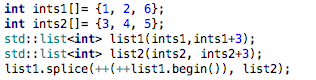
21. Опишите использование метода remove\_if

* удаляет из листа элементы, для которых предикат возвращает значение истинна



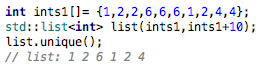
22. Опишите использование метода splice

* A.splice(iterator pos, list& B) - перемещает элементы из B в A в указанную позицию



23. Опишите использование метода unique

* в каждой последовательной группе равных элементов, удаляет все элементы, кроме первого.



Stack

24. Опишите функциональность методов push и top и как она реализуется в зависимости от выбора базового контейнера?

* push - вставляет новый элемент в вершину стека
* top - возвращает ссылку на вершину стека
* push и top вызывают методы push\_back и back базового контейнера



25. Почему стек реализован как адаптер контейнера? Каким образом?

* потому что проще было написать через готовый контейнер
* стек использует методы базового класса и организован по принципу “последним зашел - первым вышел”

26. Каким образом изменить умолчание в выборе базового контейнера?



Queue

27. Опишите функциональность методов push и front и как она реализуется в зависимости от выбора базового контейнера?

* push - вставляет новый элемент в конец очереди
* front- возвращает ссылку на первый элемент очереди
* push и front вызывают методы push\_back и back базового контейнера



28. Почему очередь реализован как адаптер контейнера? Каким образом?

* потому что проще было написать через готовый контейнер
* очередь использует методы базового класса и организована по принципу “первым зашел - первым вышел”

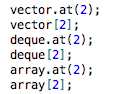
29. Каким образом изменить умолчание в выборе базового контейнера?



Общие вопросы

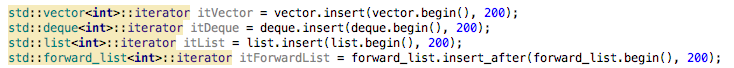
30. У каких контейнеров допустим произвольный доступ к элементам?

* vector
* deque
* array



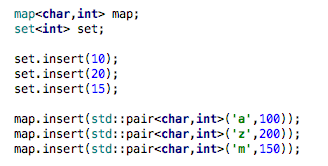
31. Для каких контейнеров сохраняются значения указателей, итераторов после вставки/удаления?

* vector
* deque
* list
* forward\_list



32. Какие последовательные контейнеры поддерживают упорядоченность элементов автоматически?

* map
* set



33. Какие средства можно использовать для сортировки элементов контейнера?

* можно использовать sort из библиотеки algorithm



34. Когда, в каких случаях нужно отдать предпочтение выбору одного из контейнеров?

* выбор контейнера зависит от конкретной задачи. Так, например, если в задаче требуется только вставка элементов, целосообразно использовать list, тк вставка работает за O(1)

35. Какие средства можно использовать и что необходимо реализовать для сравнения элементов контейнеров, если они содержат объекты пользовательских классов?

* можно использовать операторы >, <, ==, !=, <=, >=. Для этого надо перегрузить их для пользовательского класса

