

allocator

Условие

Реализовать свой аллокатор памяти, который позволит выполнять операцию `reserve` для контейнера `std::map`. Аллокатор должен параметризоваться количеством выделяемых за раз элементов. Освобождение конкретного элемента не предполагается - аллокатор должен освобождать всю память самостоятельно. Аллокатор работает с фиксированным объёмом.

Опционально реализовать расширяемость аллокатора.

Реализовать свой контейнер, который по аналогии с контейнерами `stl` параметризуется аллокатором. Контейнер должен иметь две возможности - добавить новый элемент и обойти контейнер в одном направлении.

Опционально реализовать совместимость с контейнерами `stl`.

Прикладной код должен содержать следующие вызовы:

- создание экземпляра `std::map<int, int>`
- заполнение 10 элементами, где ключ это число от 0 до 9, а значение - факториал ключа
- создание экземпляра `std::map<int, int>` с новым аллокатором ограниченным 10 элементами
- заполнение 10 элементами, где ключ это число от 0 до 9, а значение - факториал ключа
- вывод на экран всех значений (ключ и значение разделены пробелом) хранящихся в контейнере
- создание экземпляра своего контейнера для хранения `int`
- заполнение 10 элементами от 0 до 9
- создание экземпляра своего контейнера для хранения `int` с новым аллокатором ограниченным 10 элементами
- заполнение 10 элементами от 0 до 9
- вывод на экран всех значений хранящихся в контейнере

Самоконтроль

- во время заполнения стандартного контейнера со стандартным аллокатором должно быть последовательно выделено 10 раз каждый объёмом около 48 байт (зависит от компилятора)
- стандартного контейнера с реализованным аллокатором - 1 раз объёмом около 480 байт
- реализованного контейнера со стандартным аллокатором - 10 раз каждый объёмом N байт

- реализованного контейнера с реализованным аллокатором - 1 раз объёмом $10 * N$ байт
- необходимо уметь объяснить величину N
- для каждого выделения памяти должно присутствовать её освобождение
- обратить внимание на правильность использования `rebind`
- в реализованном контейнере все операции по выделению, освобождению, конструированию и деструктурированию должны выполняться методами аллокатора
- аллокатор откладывает выполнение операции освобождения памяти
- пакет `allocator` содержащий исполняемый файл `allocator` должен быть опубликован на `bintray`
- отправлена на проверку ссылка на страницу репозитория

Проверка

Задание считается выполненным успешно, если после просмотра кода, подключения репозитория, установки пакета и запуска бинарного файла командой:

```
$ allocator
```

на экран выведется содержимое контейнеров с новым аллокатором.
Команда

```
$ ltrace -e malloc -e free allocator
```

покажет снижение операций выделения памяти.