Принципы работы контейнера std::vector из библиотеки <vector>

Представление в памяти. ... . У вектора есть метод .size(), это количество реальных объектов; и .capacity(), это количество объектов, под которые зарезервирована память.

Capacity увеличивается приблизительно на 30% от первого значения, которое вышло за пределы capacity.

```
#include <iostream>
#include <vector>

void print_vector_info(std::vector<int> vec) {
    std::cout << vec.size() << " ";
    int cap = vec.capacity();
    std::cout << cap << std::endl;
    for (int i = 0; i < vec.size(); i++) {
        std::cout << vec[i] << "(" << &vec[i] << ") ";
    }
    std::cout << std::endl;
}

int main() {
    std::vector<int> vec(1);

    for (int i = 0; i < 50; i++) {
        std::cout << "Size / Capacity:   " << vec.size()
<< " ";
        std::cout << vec.capacity() << std::endl;
        vec.push_back(1 * (i + 1)); }
    /*print_vector_info(vec);

    for (int i = 0; i < 5; i++) { vec.push_back(111 * (i
+ 1)); }
    print_vector_info(vec);
    vec.erase(vec.begin() + 4);
    print_vector_info(vec);
    for (int i = 0; i < vec.size(); i++) {
        vec.erase(vec.begin());
    }
    print_vector_info(vec);*/
    system("pause");
    return 0;
}
```

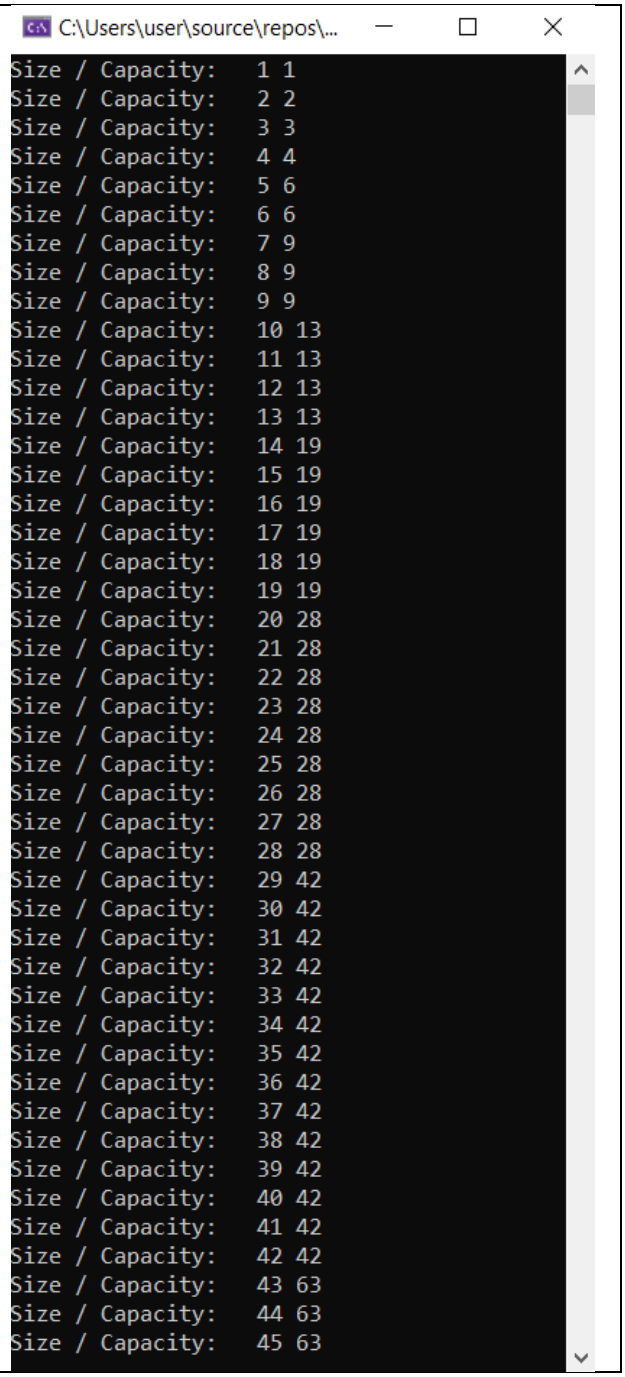


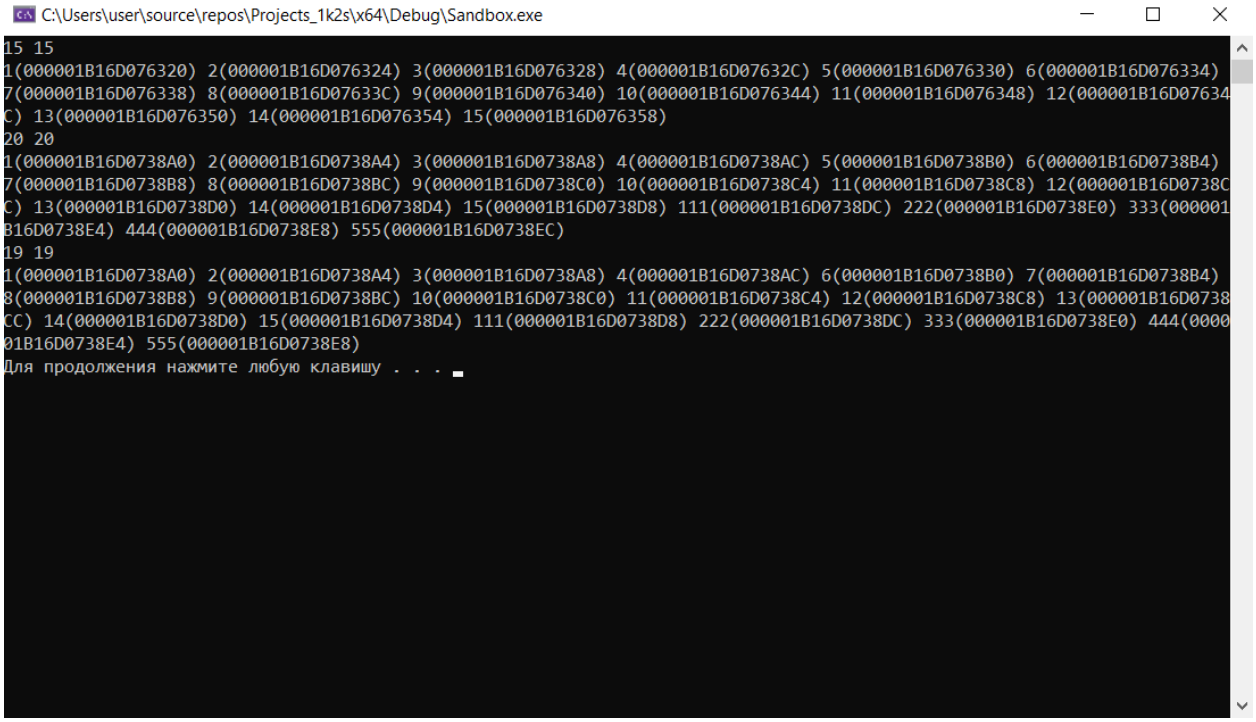
схема представления вектора в памяти

Вставка.

При добавлении элементов адреса всех ячеек меняются.

Удаление.

При удалении элементов адреса ячеек не меняются. Элементы «сдвигаются» вперед.



Сравнение с TVector. …

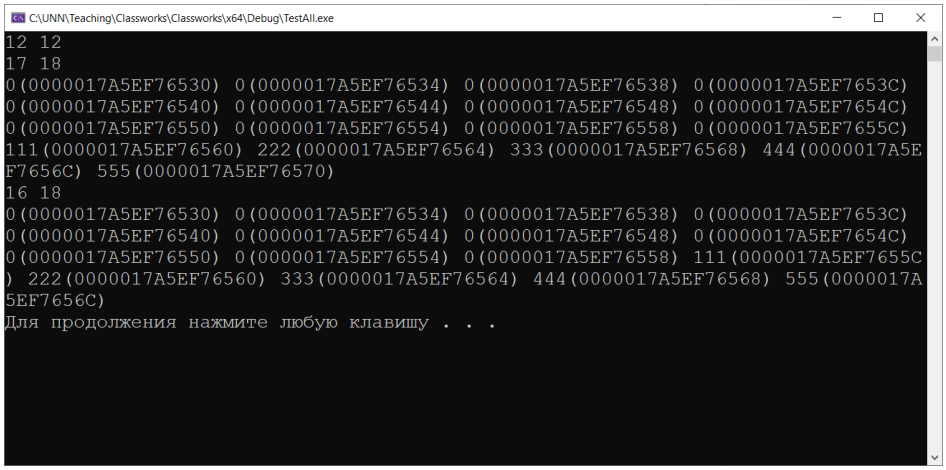


Рис. 1 - Запуск тестовой программы с выводом адресов

Приложение А: проведение эксперимента

```
#include <iostream>
#include <vector>

void print_vector_info(std::vector<int> vec) {
    std::cout << vec.size() << " " << vec.capacity() << std::endl;
    for (int i = 0; i < vec.size(); i++) {
        std::cout << vec[i] << "(" << &vec[i] << ") ";
    }
    std::cout << std::endl;
}

int main() {
    std::vector<int> vec(12);
    print_vector_info(vec);
    for (int i = 0; i < 5; i++) { vec.push_back(111 * (i + 1)); }
    print_vector_info(vec);
    vec.erase(vec.begin() + 4);
    print_vector_info(vec);
    system("pause");
    return 0;
}
```