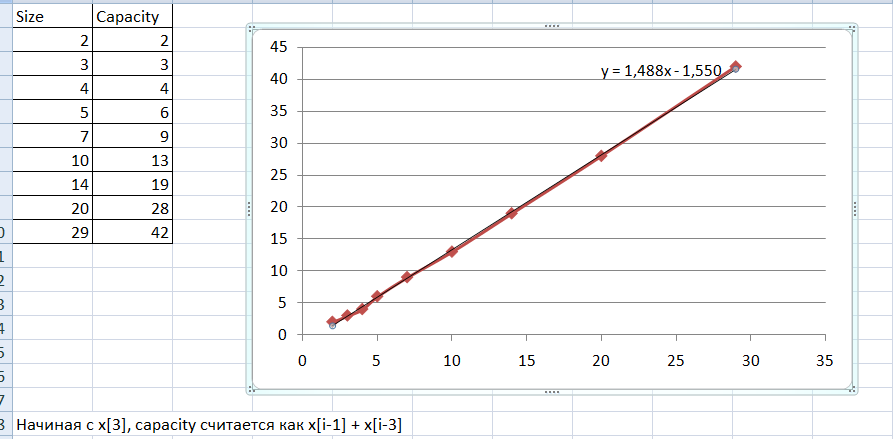
**Отчет STL: Умные указатели**

*Вариант 2*

Вместимость массива увеличивается, когда его размер достигает текущей вместимости. Это означает, что для вектора вместимость увеличивается по мере необходимости.



Мы наблюдаем, что вместимость вектора увеличивается всякий раз, когда размер достигает текущей вместимости. Однако вместимость не всегда увеличивается на фиксированное значение. Это связано с тем, что вектор использует алгоритм экспоненциального роста для определения новой вместимости.

Если бы вектор был реализован «наивно» и увеличивал свою вместимость на 1 каждый раз, когда не хватает места, то при выполнении *push\_back* 1000 раз потребовалось бы 999 выделений памяти.

Умные указатели — это классы-шаблоны в C++, которые обеспечивают безопасное и удобное управление динамически выделенной памятью. Они автоматически освобождают память, когда она больше не нужна, предотвращая утечки памяти и ошибки использования памяти.

Использование умных указателей:

* Чтобы создать умный указатель на объект, используйте функцию make\_shared():
* shared\_ptr<MyClass> ptr = make\_shared<MyClass>();
* Умные указатели действуют как обычные указатели, поэтому вы можете использовать оператор стрелки (->) или точку (ptr->member) для доступа к членам объекта.
* Когда умный указатель выходит из области видимости, он автоматически освобождает память, на которую он указывает.

Ключевое слово *auto* в C++ является неявным типом. Это означает, что компилятор автоматически определяет тип переменной на основе ее инициализатора. В нашем случае переменная p инициализируется результатом функции *foo()*, которая возвращает *shared\_ptr<Test>*. Поэтому компилятор автоматически выводит тип *auto* как *shared\_ptr<Test>*.

Использование *auto* может сделать код более лаконичным и понятным, поскольку вы не указываете явно тип переменной. Однако важно помнить, что вы должны быть уверены, что компилятор правильно выведет тип, чтобы избежать ошибок.

Умные указатели поддерживают концепцию множественного владения, что означает, что несколько умных указателей могут указывать на один и тот же объект. Когда последний умный указатель, владеющий объектом, выходит из области видимости, объект уничтожается.

Вывод:

Умные указатели являются мощным инструментом для управления динамически выделенной памятью в C++. Они обеспечивают автоматическое освобождение памяти, предотвращая утечки памяти и ошибки использования памяти.