

СЕМИНАР ПО ПРОГРАММИРОВАНИЮ НА C++
6 ОКТЯБРЯ 2015Г.

1. Реализуйте шаблонный функциональный объект

```
template <typename T>
struct TLessCmp {
    static unsigned int CmpCount;
    bool operator()(const T& lhs, const T& rhs);
};
```

который определяет меньше ли **lhs** чем **rhs**. Реализуйте подсчет количества сравнений. Создайте массив из **N** случайных целых чисел (**N** вводится с клавиатуры), отсортируйте его с помощью стандартной функции `std::sort` и `TLessCmp`. Сравните отношение количества сравнений во время сортировки к количеству элементов массива. Продемонстрируйте работу алгоритма.

2. Реализуйте шаблонную структуру

```
template <unsigned int N, unsigned int K>
struct C {
    static const unsigned int value = ...;
};
```

которая сохраняет в **value** количество сочетаний из **N** по **K**. Продемонстрируйте работу алгоритма.

3. Реализуйте шаблонный класс

```
template <typename T>
class TStack {
private:
    ...
public:
    T& Top();
    void Pop();
    void Push(const T& element)
};
```

который представляет стек. Свои элементы класс должен хранить в динамической куче. Реализуйте эффективную стратегию выделения памяти блоками.

4. Реализуйте шаблонный класс

```
template <typename T, unsigned int Capacity>
class TLocalStack {
private:
    ...
public:
    T& Top();
    void Pop();
    void Push(const T& element)
};
```

который представляет стек и хранит свои элементы в локальном стеке. Память под элементы класс должен выделять один раз при создании экземпляра. Продемонстрируйте совместную работу классов из задания 3 и 4.