

Третья небольшая домашка посвящена использованию исключений в вашем коде.

## 1 Задание

1. В имеющийся класс `subvector` необходимо добавить работу с исключениями.
2. Исключения, которые кидаете вы из недр класса, должны быть продолжением стандартной иерархии исключений. Логично начать с класса типа `VectorException`, отнаследованного от `std::exception`, и далее уже работать со своим классом.
3. Так как класс `subvector` – это в основном служебный класс (вспомните, как вы писали матрицу на нем), то по исключению должно быть понятно, что случилось и где случилось. Это облегчает отладку и делает работу с кодом удобнее. Попробуйте в описании ошибки дать подробное описание произошедшей исключительной ситуации.
4. Везде, где это возможно, постарайтесь сделать строгую гарантию безопасности, попробуйте пользоваться принципом критической линии.
5. Будьте внимательны с методом `push_back`.

Интерфейс для класса `subvector` лежит в файле `interface.cpp`. Напишите небольшой `main` с использованием `try+catch`, демонстрирующий корректную работу с исключениями.

**Напоминаю**, что класс `subvector` должен быть спроектирован так, что он удовлетворяет правилу пяти, для этого мы писали делали прошлую лабу.

## 2 Оценивание

1. Своя иерархия классов исключений с корректным наследованием - 2 балла
2. Все методы со строгой гарантией безопасности - 4 балла

Итого за лабу получается  $2 + 4 = 6$  баллов. Здесь дополнительных заданий не будет.