

## Контрольные вопросы:

1. (10 б.) Для чего нужны ключевые слова `try`, `throw`, `catch`, `noexcept`? Почему все исключения в C++ унаследованы от `std::exception`? Как называется метод у `std::exception` и его потомков, позволяющий получить информацию об ошибке?
2. (10 б.) В каких случаях следует рассматривать применение исключений? Почему, вообще говоря, их следует применять как можно реже?
3. (10 б.) Что означает отсутствие транзакционности у блока `try` и к каким проблемам это может привести? Как эти проблемы решены в языке Java?
4. (10 б.) Какие действия следует предпринимать при взаимодействии с чужим кодом (встраивании, вызове), выбрасывающим исключения? Назовите строго два варианта. Поясните, как осуществить выбор между ними.

## Упражнения:

1. (25 б.) Модифицируйте класс `Дроби` из позапрошлого задания, добавив обработку ошибки, связанной с нулевым знаменателем. Используйте наиболее подходящее исключение из стандартных или разработайте своё (удостоверьтесь, что оно ловится и обрабатывается точно так же, как и стандартные). Продемонстрируйте работу кода в функции `main`.
2. (75 б.) Разработайте класс `Даты`, хранящий месяц, год и день. Реализуйте один конструктор, принимающий все три аргумента, а также геттеры полей. Добавьте оператор сравнения дат и оператор потокового вывода. В конструкторе реализуйте проверку введённых данных на корректность. Если потребуется, разработайте один или несколько классов исключений. Корректным считается любой целочисленный год, в т.ч. отрицательный. Напоминалка: функция получения количества дней в месяце в домашке уже фигурировала.