# Проверяемое задание № 1

**Тема 1. Обработка исключений**

Лекция 1.1. Обработка исключений

**Цель работы**

Сформировать навыки использования конструкций try/catch/finally, позволяющих реализовать в приложениях структурированную обработку исключений, и научиться разбираться в ключевых концепциях, которые позволяют создавать и вызывать исключения.

**«Задание**

* Разработать тип для работы с матрицами.
* Реализовать методы, позволяющие выполнять операции сложения, вычитания и умножения матриц, предусмотрев возможность их выполнения, в противном случае должно генерироваться исключение.
* Реализовать метод GetEmpty(…), который возвращает нулевую матрицу заданного размера.
* Объявить и применить свой тип исключения:

- в сообщении исключения должна быть информация о размерах матриц;

- размеры матриц должны быть в полях исключения;

- должны быть созданы разные виды конструкторов по примеру базового исключения System.Exception.

* В качестве UI-интерфейса использовать консольное приложение с интерфейсом командной строки, WinForms или WPF-приложение.
* Создать unit-тесты для тестирования методов разработанного типа». [2]

**Рекомендации по выполнению**

Старайтесь обрабатывать только известные вам исключения. Если вы все же обрабатываете все исключения, то или, проведя необходимую обработку, генерируйте исключение повторно, чтобы его могли обработать последующие фильтры, или выводите пользователю максимально полную информацию об ошибке. Если целью перехвата исключений является очистка ресурсов после сбоя, лучше воспользоваться секцией finally.

* Доступ к чтению и записи данных матриц должен осуществляться только через индексатор. Не должно быть доступа к данным матрицы в обход индексатора.
* Ввод и вывод элементов матрицы должен выполняться без дублирования кода.
* Тип «матрица» не должен ничего выводить на «экран» и ничего считывать с «экрана».
* Нельзя изменять размеры матрицы после её создания.

**Важные материалы для выполнения задания (!)**

1. Handling and Throwing Exceptions – <http://msdn.microsoft.com/en-us/library/5b2yeyab.aspx>
2. Обработка и создание исключений – <http://msdn.microsoft.com/ru-ru/library/5b2yeyab.aspx>
3. Exception Class and Properties – <http://msdn.microsoft.com/en-us/library/z4c5tckx.aspx>
4. How to: Use Specific Exceptions in a Catch Block – <http://msdn.microsoft.com/en-us/library/3tca6706.aspx>
5. How to: Explicitly Throw Exceptions – <http://msdn.microsoft.com/en-us/library/xhcbs8fz.aspx>
6. How to: Create User-Defined Exceptions – <http://msdn.microsoft.com/en-us/library/87cdya3t.aspx>
7. Remarks and Examples for Exception Class – <http://msdn.microsoft.com/en-us/library/system.exception.aspx>
8. .NET Framework Development Guide: Best Practices for Handling Exceptions <http://msdn.microsoft.com/en-us/library/seyhszts.aspx>
9. **Design Guidelines for Exceptions (!) –** <http://msdn.microsoft.com/en-us/library/ms229014.aspx>

**Требования к предоставлению результатов (!)**

**К отчету следует приложить:**

* листинг программ, написанных на языке программирования C#;
* скриншоты и краткое описание работы классов и методов приложений.