|  |
| --- |
| EPAM Systems, RD Dep. |
| Практическое задание  NET.C#.07 Обработка исключений |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| REVISION HISTORY | | | | | |
| Ver. | Description of Change | Author | Date | Approved | |
| Name | Effective Date |
| 1.0 | Initial version | Анжелика Кравчук |  |  |  |
| 1.1 | Review and corrections. | Владимир Тихон |  |  |  |
| 1.2 | Added details | Oleg |  |  |  |

**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 возращает нулевую матрицу заданнаго размера.
* Объявить и применить свой тип исключения:
  + В сообщении исключения должна быть информация о размерах матриц.
  + Также размеры матриц должны быть в полях исключения.
  + Нужно создать 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.

**Рекомендации к выполнению:**

* 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 её размеры матрицы после её создания.

**Важно (!):**

* Handling and Throwing Exceptions

<http://msdn.microsoft.com/en-us/library/5b2yeyab.aspx>

* + Обработка и создание исключений

<http://msdn.microsoft.com/ru-ru/library/5b2yeyab.aspx>

* Exception Class and Properties

<http://msdn.microsoft.com/en-us/library/z4c5tckx.aspx>

* How to: Use Specific Exceptions in a Catch Block

<http://msdn.microsoft.com/en-us/library/3tca6706.aspx>

* How to: Explicitly Throw Exceptions

<http://msdn.microsoft.com/en-us/library/xhcbs8fz.aspx>

* How to: Create User-Defined Exceptions

<http://msdn.microsoft.com/en-us/library/87cdya3t.aspx>

* Remarks and Examples for Exception Class

<http://msdn.microsoft.com/en-us/library/system.exception.aspx>

* NET Framework Development Guide: Best Practices for Handling Exceptions

<http://msdn.microsoft.com/en-us/library/seyhszts.aspx>

* **Design Guidelines for Exceptions (!)**

<http://msdn.microsoft.com/en-us/library/ms229014.aspx>

* + **Правила разработки исключений (!)**
  + <http://msdn.microsoft.com/ru-ru/library/ms229014.aspx>

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.