**26.Изключения в .NET . Дефиниране на собствено изключение**.

**Собствени изключения**

В .NET Framework програмистите могат да дефинират собствени класове за изключения и да създават класови йерархии с тях. Това осигурява много голяма гъвкавост при управлението на грешки и необичайни ситу- ации. В по-големите приложения изключенията се разделят в логически в категории и за всяка категория се дефинира по един базов клас, а за конкретните представители на категориите се дефинира по един клас- наследник. Създава се по един абстрактен базов клас за категорията изключения, свързани с клиентите (**CustomerException**) и за категорията изключения, свързани с поръчките (**OrderException**). Наследниците на **OrderException** и **CustomerException** също могат да се подреждат в класова йерархия и да дефинират собствени подкатегории.

При работата на приложението, използващо класовата йерархия от примера могат да се прихващат наведнъж всички грешки, свързани с клиентите или само някои конкретни от тях. Това дава добра гъвкавост при управлението на грешките.

Добре е да се спазва правилото, че йерархиите трябва да са широки и плитки, т.е. класовете на изключения трябва да са производни на тип, който се намира близо до **System.Exception**, и трябва да бъдат не повече от две или три нива надълбоко. Ако дефинираме тип за изключение, който няма да бъде базов за други типове, маркираме го като **sealed**, а ако не искаме да бъде инстанциран директно, го правим абстрактен.

**Дефиниране на собствени изключения**

|  |
| --- |
| За дефинирането на собствени изключения се наследява класът **System. ApplicationException** и му се създават подходящи конструктори и евен-туално му се добавят и допълнителни свойства, даващи специфична информация за проблема. Препоръчва се винаги да се дефинират поне следните два конструктора: MyException(string message); MyException(string message, Exception InnerException);  |