

# Технологии разработки Internet- приложений

ASP.NET приложения –  
обработка ошибок страниц и  
приложения, Global.aspx

# Вывод информации об ошибках

Ошибки, которые трудно предупредить в ходе функционирования приложения, целесообразно контролировать внутри блока `try ... catch`. В C# объекты отслеживаются сборщиком мусора среды CLR, а не компилятором, поэтому обработка исключений организуется очень эффективно и не сказывается на производительности программы.

---

Вывод пользовательских сообщений об ошибках включается декларативно, с помощью установки, задаваемой в разделе `<customErrors>` файла **web.config**:

```
<configuration>
  <system.web>
    . . .
    <customErrors mode = "Remoteonly" />
  </system.web>
</configuration>
```

Атрибут `mode` определяет, будет ли вывод пользовательских сообщений об ошибках включен, отключен или включен только для удаленных клиентов. Данный атрибут является обязательным. По умолчанию он имеет значение `Remoteonly`, при котором удаленные пользователи видят стандартную страницу с минимально информативным сообщением об ошибке, не говорящем ни о чем, кроме того, что на сервере что-то не так. Локальные пользователи при этом получают сообщения ASP.NET с детальными описаниями ошибок. Если установить `mode = "Off "`, то подробные сообщения об ошибках увидят и локальные и удалённые пользователи.

# Пользовательские файлы ошибок

Для того, чтобы выводить более профессиональные и дружелюбные пользователю сообщения, которые были согласованы с общим интерфейсом сайта, необходимо включить режим `mode = "On"` в файл `web.config`:

```
<configuration>
  <system.web>
    <customErrors mode="On" defaultRedirect="GenericError.htm" >
      <error statusCode="403" redirect="NoAccess.htm"/>
      <error statusCode="404" redirect="FileNotFound.htm"/>
    </customErrors>
  </system.web>
</configuration>
```

После этого, какой бы ни была ошибка, ASP.NET станет переадресовывать пользователя на страницу **GenericError.htm**, содержимое которой всецело определяются разработчиком. Это происходит благодаря необязательному атрибуту `defaultRedirect`, в котором задается страница с сообщением об ошибке. Если вы хотите отдельно обработать некоторые ошибки со стандартными кодами сервера, то их обработчики можно указать в дополнительных тегах `<error ...>`

Если атрибут `mode` установлен в `On`, то локальные и удаленные пользователи перенаправляются на одну и ту же страницу с сообщением об ошибке. Если атрибут `mode` установлен в `Remoteonly` и имеется атрибут `defaultRedirect`, то удаленные пользователи перенаправляются на указанную вами страницу, а локальные – на выводимую по умолчанию.

# Исключения приложения и страниц

Различают исключения уровня страницы и уровня приложения. Первые обрабатываются обработчиками **Page\_Error** каждой страницы, а вторые – обработчиком **Application\_Error** в файле **Global.asax**.

Исключения уровня страницы называются внутренними. Информация о них находится в свойстве **InnerException** класса **Exception**. Необработанные внутренние исключения вызывают исключения уровня приложения – **HttpUnhandledException**.

На уровне приложения, например, в файле **Global.aspx** можно работать как с внутренними исключениями, так и с исключениями всего приложения. Для этого необходимо зафиксировать исключение в какой-либо переменной, например,

```
Exception exc = Server.GetLastError();
```

Теперь, через переменную **exc**, мы получаем доступ к исключениям уровня приложения.

Для доступа к внутренним исключениям необходимо указать конструкцию

```
exc.InnerException
```

Исключения уровня приложения всегда необходимо удалять:

```
Server.ClearError();
```

Если подобное исключение не удалить, то клиент получит сообщение об ошибке приложения.

Для генерации внутреннего исключения на уровне приложения необходимо использовать следующий конструктор:

```
Exception exc = new myException(); // создание внутр. пользователь-го исключения  
exc.Data.Add("message", "Внутренняя ошибка...", ); // информация об исключении  
throw new Exception("Ошибка приложения ...", exc); // генерация исключения
```

При этом внутреннее исключение будет иметь сообщение **"Внутренняя ошибка..."**, а исключение уровня приложения – **"Ошибка приложения ..."**

# Обработка ошибок уровня приложения

Если для всех страниц приложения алгоритм обработки ошибок один и тот же, то его следует реализовывать на уровне приложения. Такой обработчик будет перехватывать все исключения которые не были обработаны на страницах приложения при условии, что в разделе `<customErrors>` файла `web.config` атрибут `mode = "Off"` или отсутствует `defaultRedirect`.

Добавьте в корне приложения файл `Global.asax` (см. [события приложения](#)) и создайте следующий обработчик:

```
protected void Application_Error(object sender, EventArgs e)
{
    // определяем исключение приложения – необработанное исключение HttpUnhandledException
    Exception ex = Server.GetLastError();
    if (ex.InnerException != null) // если оно содержит внутренние исключения,
    {
        // анализируем внутренние исключения:
        if (ex.InnerException is NotImplementedException)
            Server.Transfer("~/errorpages/notImplementedExeption.htm");
        else
            Server.Transfer("~/errorpages/appError.htm");
    }
    if (ex is HttpException) { // Ошибки уровня HTTP-запросов для .aspx-ресурсов (не .html)
        Server.Transfer("~/errorpages/HttpException.htm");
    }
    // удаляем ошибку
    Server.ClearError();
}
```