

Учебный курс  
**Технологии и средства разработки  
корпоративных систем**

Лекция 12

**Основные возможности корпоративных  
приложений**

Лекции читает

**кандидат технических наук, доцент**

**Зыков Сергей Викторович**

***Библиотека классов***  
***Enterprise Library***

ИНТУИТ, 2009

# Содержание

## 1. Введение

- Основные понятия и назначение
- Функциональный состав и структура
- Сценарии использования

## 2. Основные блоки

- Ядро
- Кэширование
- Безопасность (криптозащита, политики, журнал операций, ...)
- Доступ к данным
- Проверка корректности (зависимости, исключения)

## 4. Пример – построение приложения для доступа к данным

## 5. Библиография

# Что такое Enterprise Library?

Enterprise Library состоит из повторно используемых, расширяемых и модифицируемых компонент для решения задач, повторяющихся от проекта к проекту.

Enterprise Library объединяет несколько функциональных блоков и ядро, отвечающее за конфигурацию, управление и создание объектов, используемых в функциональных блоках (ObjectBuilder).

Начиная с версии 2.0, Enterprise Library целиком базируется на .NET Framework. Кроме того, Enterprise Library включает в себя Configuration Console и Security Database Console.

## Назначение / цели:

- Последовательность (Consistency).
- Расширяемость (Extensibility).
- Простота в использовании (Ease of use).
- Интеграция (Integration).

## Основные сценарии использования

1. Реализация типовых нефункциональных требований к широкому спектру корпоративных приложений, создаваемых на платформе Microsoft .NET;
2. Основа создания пользовательских библиотек. Разработчики могут использовать точки расширения функциональных блоках и расширять функциональность стандартных блоков новыми библиотеками-провайдерами. Т.к. Enterprise Library поставляется с полными исходными текстами, можно расширять функциональность блоков и на более низком уровне (включая создание новых функциональных блоков на основе реализованной в библиотеке инфраструктуры)

## Основные сценарии использования

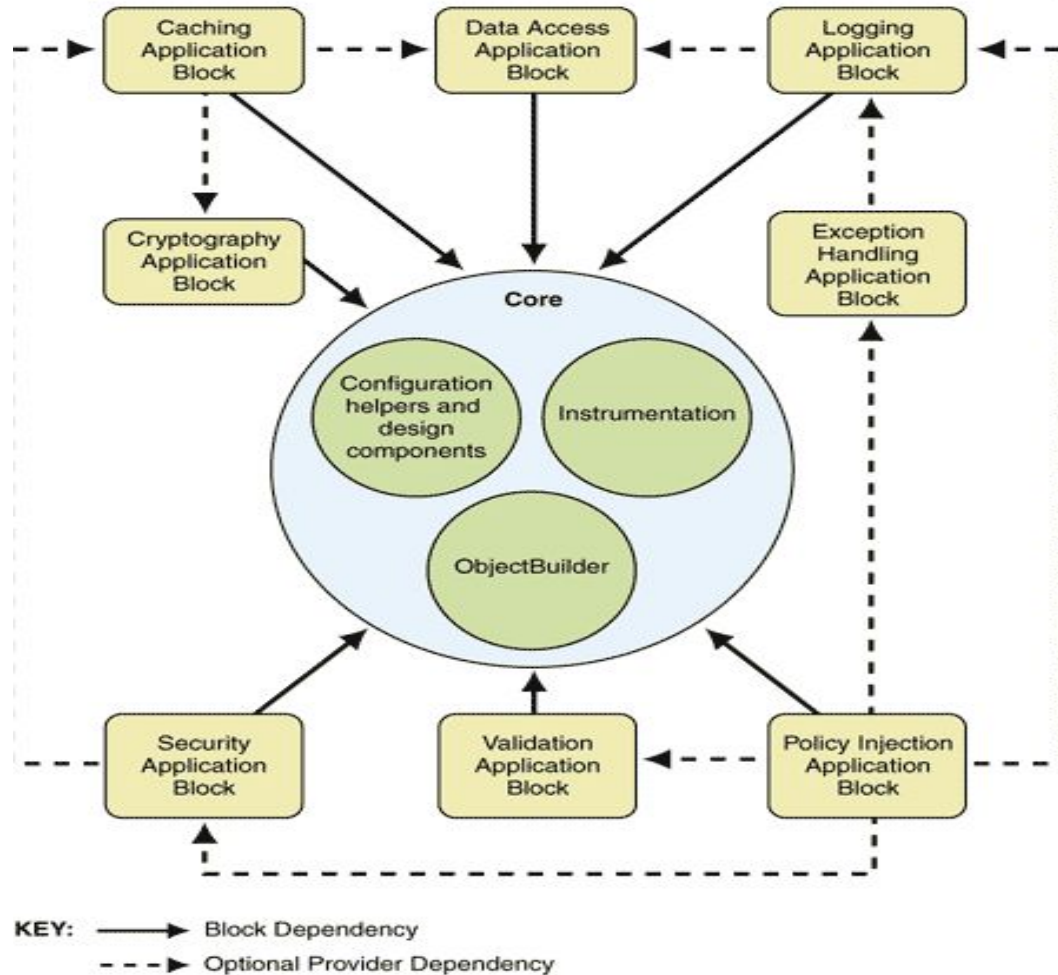
3. Отсутствие необходимости использования всех функциональных блоков в приложение включаются лишь блоки, необходимые для решения задачи
4. Доступность исходного кода позволяет включать функциональные фрагменты в пользовательские библиотеки;
5. Основа для изучения архитектурных принципов, дизайна корпоративных приложений на платформе Microsoft .NET, примеры разработки согласно рекомендованным практикам написания кода.

# Функциональные блоки

1. Caching Application Block.
2. Cryptography Application Block.
3. Data Access Application Block.
4. Exception Handling Application Block.
5. Logging Application Block.
6. Policy Injection Application Block.
7. Security Application Block.
8. Unity Application Block.
9. Validation Application Block.



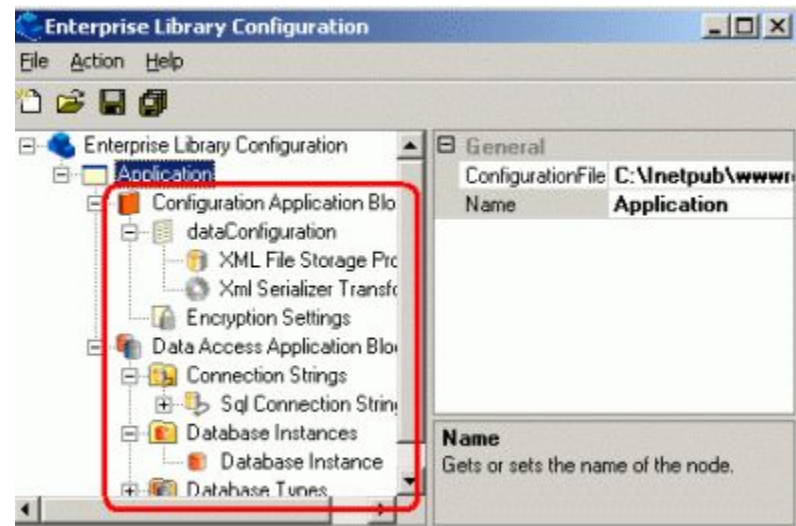
# Структурная схема



# Ядро Enterprise Library

Корпоративные системы  
Корпоративные системы  
Корпоративные системы

- функциональные блоки поддерживают общие механизмы настройки, которые позволяют определять порядок взаимодействия блоков в составе приложения, задавать механизмы расширения и т.п.
- Механизмы конфигурации используют пространство имен System.Configuration из библиотеки .NET Framework.
- Реализованы вспомогательные классы, поддерживающие для каждого функционального блока класса ConfigurationSections сохранение информации в конфигурационных файлах (app.config, web.config). Поддерживаются все возможности System.Configuration, в т.ч. шифрование и использование внешних файлов.



## Ядро Enterprise Library

- В Enterprise Library включена подсистема ObjectBuilder, которая реализована в пространстве имен Microsoft.Practices.ObjectBuilder и отвечает за создание и удаление экземпляров объектов. На уровне Enterprise Library подсистема ObjectBuilder используется для вставки конфигурационных данных в классы функциональных блоков, а также для связи управляющих классов с функциональными блоками. Для использования Enterprise Library не требуется изучения принципов работы ObjectBuilder.
- функциональные блоки могут использовать счетчики производительности, Windows Management Instrumentation (WMI) и протоколы событий. Также можно применять механизмы конфигурации для задания типа управления.

# Caching Application Block

- Блок кэширования служит для реализации локального кэша. Поддерживается кэш в памяти и (опционально) в хранилище данных, которое может быть либо хранилищем, управляемым через блок доступа к данным (Data Access Application Block), либо изолированным хранилищем (Isolated Storage).
- Обеспечивается извлечение, добавление и удаление данных из кэша. Время хранения данных задается соответствующими конфигурационными настройками.
- Локальный кэш поддерживается для одного домена приложения, вследствие чего данный функциональный блок не обеспечивает реализацию кэша, разделяемого между доменами.