|  |  |  |
| --- | --- | --- |
| **Концепция** | **Метаданные** | **Код** |
| **int TypeCode**  Код типа для быстрой идентификации объекта переноса данных (data transfer object) | IDataTransferObject  + int TypeCode { get; set; } | Класс реализует интерфейс косвенно.  Свойства интерфейса не отображаются на данные. |
| **class PersistentObject** (primary key)  Используется для моделирования сохраняемых в базе данных сущностей. Каждая сущность должна иметь первичный ключ (простой или составной). | IPersistentObject<TPrimaryKey> : IDataTransferObject  + TPrimaryKey PrimaryKey { get; set; } | Простой первичный ключ (Guid, int и т.п.) используется ссылочными типами данных, а составной - значимыми. |
| **class ReferenceObject** (ссылочные типы)  Ссылочный тип данных используется для  При этом первичным ключом такой ссылки может быть только простой (значимый, ValueType) тип данных (Guid, int и т.п.).  Равенство объектов этого типа между собой определяется равенством их кодов типа и значений первичного ключа.  Для представления этого объекта в интерфейсе пользователя может использоваться строковое значение. Доступ к этому значению может быть организован через переопределение метода ToString класса object. | ReferenceObject : IPersistentObject<Guid>  Перегрузка операторов == и !=    this.TypeCode == test.TypeCode  &&  this.PrimaryKey == test.PrimaryKey  Переопределение метода string object.ToString()  {TypeCode:PrimaryKey} | Класс реализует чтение как публичное свойство, а запись косвенно через интерфейс. |
| **class ClientReferenceObject** | ClientReferenceObject : ReferenceObject  + string View { get; set; }  Переопределение метода string object.ToString() |  |
| **class ValueObject** (значимые типы)  Тип данных для моделирования сохраняемых в базе данных объектов, имеющих составной первичный ключ.  Равенство объектов данного типа между собой определяется равенством значений всех их свойств между собой (значимые типы).  Как правило такие объекты не могут быть типами данных для свойств других классов. | ValueObjectPrimaryKey  ValueObject : ValueObjectPrimaryKey, IPersistentObject<ValueObjectPrimaryKey>  ~ нет членов | Первичный ключ реализуется как класс или интерфейс. Пользовательский класс может наследовать от ключа или реализовывать интерфейс, указанный в качестве параметра обобщённого интерфейса. |
| **Version** (rowversion)  Реализация оптимистичной стратегии блокировки данных.  Для этих целей на уровне SQL Server используется поле типа rowversion (timestamp).  В некоторых случаях может использоваться несколько обычных полей данных таблицы SQL Server. | IVersion  + byte[] Version { get; set; } | Класс реализует интерфейс косвенно. |
| **FieldPurpose** (составные типы данных)  1С:Предприятие имеет концепцию составных типов данных, которая по сути своей является универсальным типом данных Variant из COM.  На уровне СУБД значения таких типов хранятся в нескольких полях таблицы базы данных, каждое из которых имеет своё предназначение (purpose). | public enum FieldPurpose  {  Value, // Обычное значение для  // не составного типа данных  Locator, // Указатель на поля значений  Boolean, // Хранилище значения Булево  String, // Хранилище значения Строка  Number, // Хранилище значения Число  Binary, // Хранилище бинарного значения  DateTime, // Хранилище значения Дата  Object, // Хранилище значения Ссылка  TypeCode // Код типа ссылки (int)  } | Свойства класса этого типа имеют тип значения ReferenceObject. 1C# ограничивает использование примитивных типов данных и ссылочных одновременно. |
| **enum PersistentState**  Внутреннее состояние объекта переноса данных (new, changed, original, deleted, virtual, loading).  Управление состоянием может быть распределено между разными классами: сам объект, при изменении значений его свойств, классы отображения свойств объекта на структуры хранения данных, сериализаторы и т.п. | IPersistentState  + PersistentState State { get; set; }  + void OnStateChanged(StateEventArgs args);  + event StateChangedEventHandler StateChanged;  Реализует концепцию self-tracking entity.  Состояние loading используется для поддержки концепции lazy load. | Класс реализует интерфейс косвенно.  Событие реализуется как публичное свойство.  Свойства интерфейса не участвуют в отображении данных. |