Принципы разработки в среде ДОМИНО

- Модель данных
- Справочники
- Документы
- Регистры накопления
- Регистры сведений
- План счетов и регистры бухгалтерского учета
- Поддержка нескольких систем учета
- Справочники

Модель данных

Разработка приложения в среде Домино отличается от разработки в универсальных системах разработки программного обеспечения тем, что приложение на платформе ДОМИНО разрабатывается в рамках проблемно-ориентированной модели с применением метаданных, что значительно упрощает, ускоряет разработку и позволяет разработчикам сосредоточиться на решаемой задаче и больше похоже на проектирование.

При обычном программировании необходимо описать целый ряд сущностей и способы их регистрации и взаимодействия. При проектировании на платформе Домино разработчик использует объекты (сущности), заранее описанные в платформе и соответствующие модели данных решаемой бизнес задачи.

Если решаемая задача соответствует модели данных, то это не только в несколько раз ускоряет разработку новых приложений (новой функциональности существующих приложений), но и снижает затраты/риски отклонения от общей модели данных в дальнейшем.

Модель данных платформы ДОМИНО хорошо проработана на системной уровне с точки зрения производительности, реализации методов обработки и транзакционной целостности данных и при ее использовании в дальнейшем не будет возникать неожиданные проблемы с обработкой данных даже на очень больших базах данных. Есть многолетний положительный опыт использования у крупных клиентов и в больших сетях магазинов баз данных с объемом более 2-х терабайт.

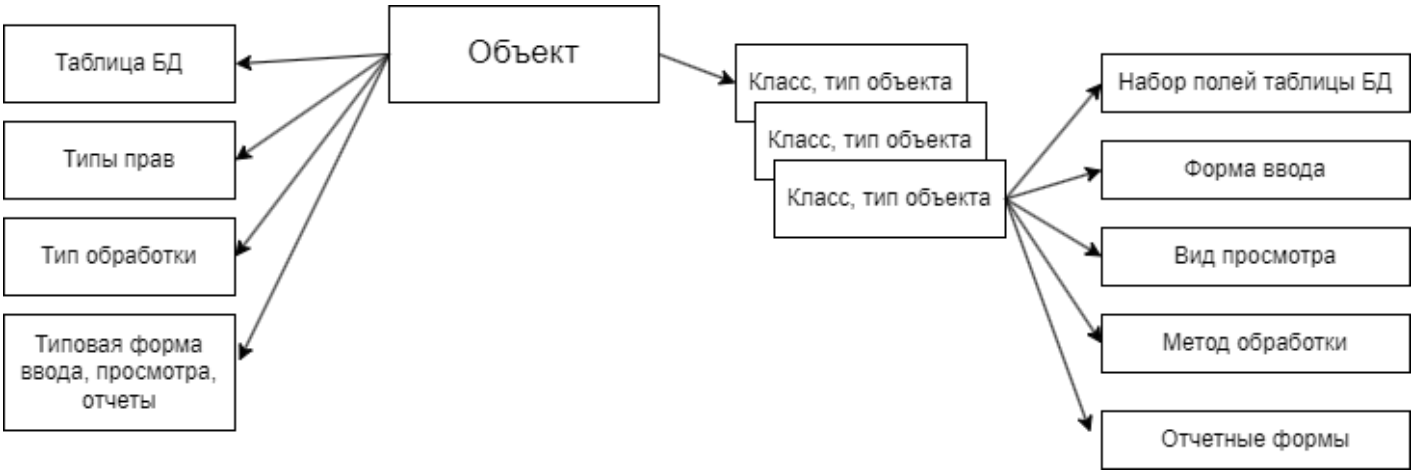
Выделение объектов, типов и классов определялось следующими соображениями:

- понятны назначение и роль объекта в модели данных
- типизация и классификация должны обеспечивать унификацию разработки прикладных решений
- понятны способы применения объектов, классов и типов

Полученные наборы объектов, классов и типов описаны в платформе и все вместе составляют метаданные, которые в значительной степени определяют структуру

проектирования приложения. Каждое приложение Домино основывается и опирается на структуру метаданных. При этом, непосредственно в приложении можно добавить свои объекты, типы и классы.

Описание объекта включает такую информацию, как используемая таблица базы данных, поля таблицы, типовые интерфейсные элементы, форму ввода, виды просмотра и наборы прав, которые будут использоваться в системе разграничения прав доступа. На основе имеющихся объектов разработчик может создать свои собственные объекты. Эти новые объекты с момента создания будут наследовать свойства от объекта-родителя. При необходимости разработчик может указать особенности, которыми должна обладать новая сущность в отличие от родителя.



В платформе ДОМИНО в модель данных уже заложен полнофункциональный набор объектов, который позволяет, не добавляя никаких новых (по сравнению с заложенной в платформе) объектов, быстро и удобно поддерживать реализовать прикладную задачу или выполнить доработку существующей.

Имеются следующие объекты:

- Кодификаторы
- Каталоги/Справочники
- Товары
- Партнёры /Подразделения
- Документы
- Строки документов
- Проводки
- Учетные регистры

- Протоколы и журналы

При необходимости, помимо имеющихся объектов, разработчик приложения ДОМИНО может описать свои объекты, которые сможет использовать в разрабатываемом приложении с возможностью дальнейшего наследования.

Справочники

Описания таких сущностей, как товары, контрагенты, валюты, склады, объединяет наличие таких общих свойств, как внутренняя идентификация объекта в системе, необходимость поддержки иерархии и группировки элементов, необходимость поддержки вложенных таблиц. Информацию об этих объектах надо хранить, они задействованы в хозяйственных операциях предприятия и т.д. В "Домино" все такие сущности объединены в общий класс "справочник", для которого перечисленные выше свойства и возможности поддерживаются на уровне платформы.

Разработчик может задать состав необходимых полей справочника, задать связи между ними, сконструировать экранную форму и самостоятельно, с помощью специального конструктора экранных форм придав ей в точности такой вид и свойства, которые соответствуют особенностям решаемой задачи, необходимы по эргономическим соображениям и т.д.

Документы

Документы — счета, накладные, заказы и т.п. — фиксируют различные события, происходящие в хозяйственной жизни
организации. Важным свойством документа является его привязка ко времени. В "Домино" для этих объектов поддерживается идентификация самого события хозяйственной жизни, отражение в учетных механизмах, контроль последовательности и отражение событий в реальном времени. Такой набор функциональности заложен в систему и обеспечивает быструю разработку разнообразных документов.

Состав полей документа и набор его строк целиком зависят от того бизнес-процесса, в котором участвует каждый конкретный тип документа. Очевидно, что состав и структура разных документов различается. Так условная приходная накладная существенно отличается от условного платежного поручения, но для все документов используется стандартный механизм описания их структуры и конкретных методов обработки.

Однако сам по себе документ только описывает некоторый факт происшедший в хозяйственной жизни. В бизнес-приложениях эти факты необходимо еще *учитывать* — отражать движения ресурсов (товаров, финансов и т.п.) в различных системах учета.

Для этого документ необходимо *акцептовать*. С точки зрения разработчика проведение означает вызов соответствующей
обработки, выполнение алгоритма на встроенном языке "Домино", описывающего отражение события в различных
системах учета. Акцепт документа возможен как по специальной команде , так и автоматически при сохранении документа.

Регистры накопления

Механизм многомерных регистров накопления "отвечает" за учет движений ресурсов (финансов, товаров, материалов и т.д.) и позволяет автоматизировать такие направления, как складской учет, взаиморасчеты, планирование. В регистрах накопления хранится информация о поступлении и расходе тех или иных ресурсов, а заложенная в платформу "Домино" функциональность этих регистров предоставляет возможности получения остатков на определенный момент времени, расчета итогов, кэширования итогов и т.д.

В данном случае производится уже не только описание структур данных и форм их представления, но и определение бизнес-логики работы приложения. Для описания этой бизнес-логики уже необходимо программирование на встроенном языке "Домино" — многообразие вариантов учета в зависимости от ситуации, вида деятельности и особенностей работы конкретного предприятия очень велико и его целесообразно описывать алгоритмически. . На практике такие алгоритмы, как правило, сложны — они могут, например, автоматически рассчитывать скидки, поддерживать те или иные способы списания материальных ценностей (по средней стоимости, LIFO, FIFO), обеспечивать контроль наличия товаров на складе или выдачи товаров покупателю в зависимости от его задолженности и т.п., тем не менее все они могут быть реализованы как элементы проекта конкретного приложения механизмами, имеющимися в платформе.

Регистры сведений

Регистры сведений предназначены для хранения многомерных сведений о значениях различных величин, которые сами
по себе не имеют объектной семантики. Такими значениями могут быть, например, курсы валют или цены на товары конкурентов по состоянию на определенную дату. Эта информация может быть как статической, так и изменяющейся с течением времени — в этом случае для нее предусмотрено хранение истории изменений.

Функциональность регистров сведений в "Домино" обеспечивает задание произвольной периодичности хранения,
возможность получения "срезов" информации на определенный период и т.д.

План счетов и регистры бухгалтерского учета

Система двойной записи бухгалтерского учета представляет собой отдельную модель учета со своей спецификой, поэтому в "Домино" план счетов и регистры бухгалтерского учета выделены в отдельные классы сущностей.

Практика применения "Домино" для автоматизации учета как в России, так и в других странах, очень широка и пока еще не возникало ситуации, чтобы разработчикам прикладных решений не хватило возможностей механизмов бухгалтерского учета, реализованного в "Домино". Причем данные механизмы никак не навязывают разработчику собственно принципов ведения бухгалтерского учета.

Заметим, что создание подобного инструмента с нуля является весьма непростой задачей, даже если использовать хотя бы часть возможностей из тех, что реализованы в "Домино", а это:

- многоуровневый план счетов с фиксированной или переменной разрядностью кодов;
- многоуровневый и многомерный аналитический учет;
- многовалютный учет;
- учет по нескольким планам счетов;
- учет по нескольким организациям (юридическим лицам);
- опциональное ведение количественного, суммового и валютного учета по отдельным разрезам аналитики и т.д.

При этом система предоставляет разработчику инструмент для манипулирования итогами, который сводит построение весьма сложных запросов по всем перечисленным "степеням свободы" всего к нескольким строчкам.

Поддержка нескольких систем учета

В каждом прикладном решении на платформе "Домино" может поддерживаться несколько систем учета.
Например, расходная накладная может одновременно отметить в системе учета товаров, что товаров на нашем складе стало меньше, отметить в системе взаиморасчетов увеличение задолженности клиента, которому мы отгрузили товар, отметить в системе бухгалтерского учета изменение остатков по счетам и т.д.

Между документами и системами учета (регистрами) поддерживаются связи типа "многие ко многим" — один документ
может включать записи в различные регистры, поддерживаемые прикладным решением, и наоборот — записи в один регистр могут формироваться при проведении документов различных типов.

Справочники

Описания таких сущностей, как товары, контрагенты, валюты, склады, объединяет наличие таких общих свойств, как внутренняя идентификация объекта в системе, необходимость поддержки иерархии и группировки элементов, необходимость поддержки вложенных таблиц. Информацию об этих объектах надо хранить, они задействованы в хозяйственных операциях предприятия и т.д. В "Домино" все такие сущности объединены в общий класс "справочник", для которого перечисленные выше свойства и возможности поддерживаются на уровне платформы.

Разработчик может задать состав необходимых полей справочника, задать связи между ними, сконструировать экранную форму и самостоятельно, с помощью специального конструктора экранных форм придав ей в точности такой вид и свойства, которые соответствуют особенностям решаемой задачи, необходимы по эргономическим соображениям и т.д.