

остановке также остановить события, вызываемые события сеанса отладки уведомляет диспетчер (SDM) и интегрированная среда разработки которой отлаживаемого приложения на остановку выполнения кода. `IDebugBreakpointEvent2` и `IDebugExceptionEvent2` интерфейсы примеры остановить события.

Режим шага

Режим шага происходит, когда шаги программы в следующей строке кода или в над или из функции. Шаг выполняется путем вызова метода `IDebugProcess3::Step`. Этот метод требует `DWORDs`, задающее `STEPUNIT` и `STEPKIND` перечисления в качестве входных параметров. Если программы шаги в следующей строке кода или в функцию или выполняются до курсора или на точке останова, `DE` переходов автоматически обратно в режим приостановки выполнения.

УДК 381

Мацур Е.В.

МОДЕЛИ ДАННЫХ В СУБД

БНТУ, Минск

Научный руководитель: Астапчик Н.И.

Хранимые в базе данные имеют определенную логическую структуру, то есть представлены некоторой моделью, поддерживаемой СУБД (система управления базами данных – совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных).

К числу важнейших относят следующие модели данных: иерархическая; сетевая; реляционная; объектно-ориентированная.

Иерархическая модель данных – это модель данных, где используется представление базы данных в виде древовидной

структуры, состоящей из объектов (данных) различных уровней. Между объектами существуют связи, каждый объект может включать в себя несколько объектов более низкого уровня. Базы данных с иерархической моделью одни из самых старых, они стали первыми системами управления базами данных для мейнфреймов. Преимуществами такой модели является простота, использование отношений предок/потомок, быстрое действие. Если структура данных оказывалась сложнее, чем обычная иерархия, простота структуры иерархической СУБД становилась её недостатком.

Сетевая модель означает представление данных в виде произвольного графа. Разница между иерархической и сетевой моделями в том, что в иерархических структурах запись-потомок должна иметь в точности одного предка, а в сетевой – любое количество. Достоинствами модели данных является возможность её эффективной реализации по показателям затрат памяти и оперативности. Недостатком – сложность и жесткость схемы БД, построенной на ее основе.

Не будучи хронологически первой, наиболее популярной с начала 80-х гг. была и до сих пор остается реляционная (с англ. relation – отношение, создана Э.Коддом в 1970 году) модель данных. Если можно говорить об основной идее использования реляционного подхода в СУБД, то это именно предсказуемость результатов работы с данными, обеспечиваемая математическим аппаратом в основе этого подхода. В реляционной модели считается, что все данные представлены в виде двумерной таблицы, привычной для человека. Очевидными достоинствами реляционной модели являются простота и наглядность представления данных, развитый математический аппарат для работы с таблицами, удобные высокоуровневые языки для описания и обработки данных. Вместе с тем вплоть до начала 80-х годов реляционная модель данных считалась недостаточно эффективной, поскольку отслеживание связей,

заданных значениями данных, требовали более высокой трудоемкости по сравнению с предшествующими моделями. Этот основной недостаток реляционной модели был преодолен путем бурного развития возможностей вычислительной техники и совершенствования средств быстрого поиска данных. В настоящее время основным предметом критики реляционных СУБД является не их недостаточная эффективность, а присущая этим системам некоторая ограниченность при использовании в так называемых нетрадиционных областях.

Объектно-ориентированные БД объединяют в себе две модели данных, реляционную и сетевую, и используются для создания крупных БД со сложными структурами данных. К достоинствам объектно-ориентированной модели обычно относят возможность для пользователя системы определять свои сколь угодно сложные типы данных, наличие наследуемости свойств объектов, повторное использование программного описания типов объектов при обращении к другим типам, на них ссылающимся. К недостаткам объектно-ориентированной модели можно отнести отсутствие строгих определений, отсутствие общепотребимых стандартов, позволяющих связывать конкретные объектно-ориентированные системы с другими системами работы с данными, разное понимание терминов и различия в терминологии, как следствие – эта модель не исследована столь тщательно математически, как реляционная.

Использование модели данных при работе с БД неизбежно по нескольким причинам. Во-первых, модель дает общий язык пользователям, работающим с данными. Во-вторых, модель может обеспечить предсказуемость результатов работы с данными. Становится возможным объяснить пользователю, почему он получил конкретный результат при просмотре или изменении данных, и наоборот, работающий с базой может предвидеть, какого сорта он получит результат.