Лабораторная работа № 9
Возможности диаграммы классов

Назначение диаграммы

Class diagram (диаграмма классов) — основная диаграмма для создания кода приложения. При помощи диаграммы классов создается внутренняя структура системы, описывается наследование и взаимное положение классов друг относительно друга. Здесь описывается логическое представление системы. Именно логическое, так как классы - это лишь заготовки для создания объектов, на основе которых затем будут определены физические объекты.

Таким образом, диаграмма классов описывает общее представление системы и является противоположной Collaboration diagram, в которой представлены объекты системы. Однако такое разделение не является строгим правилом и возможно смешанное представление классов и объектов.

Диаграмма классов используется не только для создания логического представления системы, Rational Rose позволяет на основе диаграммы классов создавать исходный код приложения. А так как описание классов создается на языке UML, то по диаграммам, созданным в едином стиле, возможна генерации исходного кода на любом языке программирования, который поддерживается генератором кода Rational Rose.

Обычно диаграмма классов создается для всех классов системы, в отличие от диаграммы объектов, которую проектировщики создают для отдельных объектов со сложным поведением и взаимодействием.

Диаграмма классов содержит значки, представляющие классы, интерфейсы и их связи. Классы могут представлять любые C++ классы: простые, параметризированные или метаклассы. Интерфейсы — это некоторый набор действий или операций, который обслуживает взаимодействие реализаций классов.

Возможно создание одной или нескольких диаграмм классов, которые описывают классы верхнего уровня в текущей модели. Также возможно создание одной или более диаграмм классов, которые описывают классы, содержащиеся в контейнерах. Так, диаграмма классов сама по себе является контейнером для классов модели, но можно выделить дополнительные контейнеры для логической группировки классов.

Посредством диаграммы классов возможно изменение в любой момент свойств любого класса или его связей, и при этом диаграммы или спецификации, связанные с изменяемым классом, будут автоматически обновлены.

Диаграмма классов может быть использована как при анализе готовой системы, так и при разработке новой.

Создание диаграммы

Главная диаграмма классов (Main) уже присутствует во вновь созданной пустой модели, но возможно создание дополнительных диаграмм при помощи уже знакомых способов посредством контекстного меню Local View в окне Browse, при помощи пункта Browse в главном меню или при помощи кнопки.

Строка инструментов

При активизации диаграммы строка инструментов приобретает следующий вид (рис. 65).

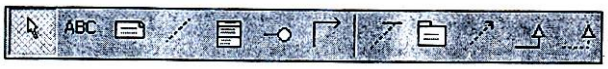

Рис.65. Строка инструментов для диаграммы классов

Class(класс)

Данный инструмент позволяет создать новый класс в диаграмме и модели. Понятие класса в Rational Rose аналогично понятию класса в C++. Класс — это установки структуры и шаблона поведения для некоторого множества реальных объектов, которые в дальнейшем будут определены в программе на основе данного шаблона. Класс — это некоторая абстракция, реального мира. Когда эта абстракция принимает конкретное воплощение, она называется объектом. Для детализации модели поведения классов, даются диаграммы состояний и действий, рассмотренные ранее.
Класс в UML нотации изображается как прямоугольник, разделенный на 3 части (рис.66). В верхней части записывается название класса, в редине — атрибуты, в нижней части — операции.

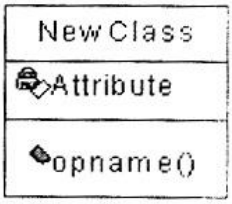

Рис.66.Изображение класса

Interface (интерфейс)

Значок Interface позволяет создать объект Interface, который указывает на видимые извне операции класса или компонента. Обычно интерфейс создается только для некоторых строго определенных классов или компонентов и предназначен скорее для логического отображении системы но может присутствовать как на диаграмме классов так и на диаграмм компонентов.

В диаграмме классов Interface обычно отображается как значок класса со стереотипом «interface».

Unidirectional Association (однонаправленная связь)

Значок Unidirectional Association позволяет создать однонаправленную связь класса с классом или класса с интерфейсом Это общий и самый слабый вид связи.

Association Class (ассоциация класса)

Значок Association Class позволяет связать классы ассоциативной связью. Это свойство сохраняется в классе, и для того чтобы его установить необходимо создать класс и связать класс реляцией с другим при помощи этого значка.

Package (контейнер)

Значок позволяет создать элемент, который используется для группировки элементов. Может быть использован для физической или логической группировки. В нашем случае контейнер удобнее всего использовать для физической группировки кода. Но для небольшой системы, которой является тепличное хозяйство, мы не будем использовать контейнеры.
созданного класса в диаграмму

Dependence of instantiates (зависимость реализации)

Значок Dependence of instantiates позволяет создать связь Dependence of instantiates, при этом генератор кода C++ Rational Rose создает код класса, включающий определения зависимого класса путем генерации директивы #include. Установка этого типа связей показывает, что класс использует другой класс как параметр в одном из методов.

Generalization (обобщение)

Значок Generalization позволяет создать связь Generalization, для которой Rational Rose создает код наследования, то есть создается подкласс для соединенного этой связью класса, наследуемого из родительского класса.

Realize (выполнять)

Значок Realize позволяет создать связь Realize между классом и интерфейсом или между компонентом и интерфейсом. Этот тип связи используется для того, чтобы показать, что класс выполняет операции, предоставляемые интерфейсом.

Помещение класса на диаграмму

Для создания нового класса и помещения его на диаграмму классов можно воспользоваться соответствующим значком из строки инструментов или меню Menu=>Tools=>Create=>Class. Мы уже создали некоторые классы системы, но не поместили их на диаграмму. Для того чтобы поместить уже созданный класс, например, EnvironmentalController на диаграмму классов, есть несколько путей:

- перетащить нужный класс мышкой из окна Browse;
- воспользоваться Menu=>Query=>Add Classes и в диалоговом окне (рис. 67) выбрать необходимые классы для включения в диаграмму.

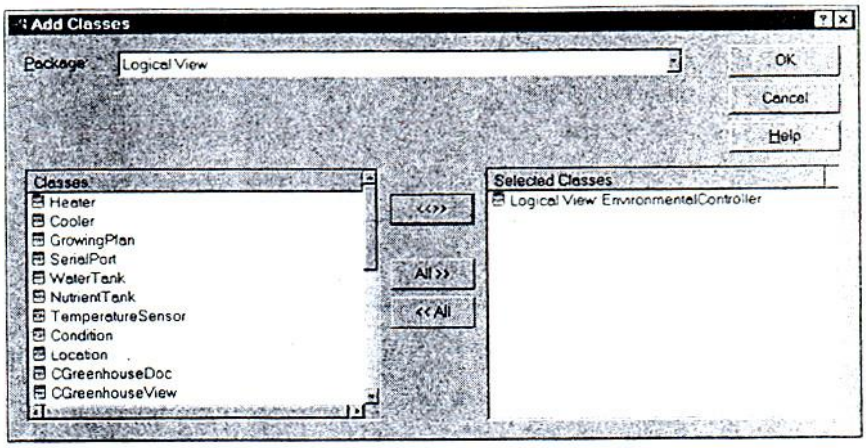

Рис. 67. Добавление созданного класса в диаграмму

Первый способ быстрый, но не для всех диаграмм подходит. Но если необходимо включить в диаграмму несколько уже имеющихся классов, то второй способ определенно лучше.

Понятие «стереотип» для класса

Мы уже встречались с понятием стереотипа в других диаграммах, но для диаграммы классов необходимо более подробное рассмотрение данного понятия.

Стереотип позволяет указывать дополнительные особенности для разрабатываемой модели, которые не поддерживаются языком UML. Понятие стереотипа позволяет легче добавлять информацию в новые элементы модели путем выбора стереотипа для этих элементов из уже заданных и представляет собой дополнительную классификацию элементов.

Некоторые стереотипы уже определены в Rational Rose, но всегда можно добавить новые стереотипы пользователя, которые сохраняются в файле стереотипов. В качестве примера использования стереотипов можно привести следующий. Вы можете использовать для классов, которые предназначены для хранения данных, стереотип «Storage», для классов, которые предназначены для показа данных — стереотип «View», для классов, которые предоставляют пользователю возможность контроля за выполнением программы — . «Controller».

Стереотип может быть показан для класса или скрыт при помощи пункта Options контекстного меню класса.

На рис.68 показан пример различных вариантов визуализации стереотипов. Слева направо: None, Label, Decoration, Icon.

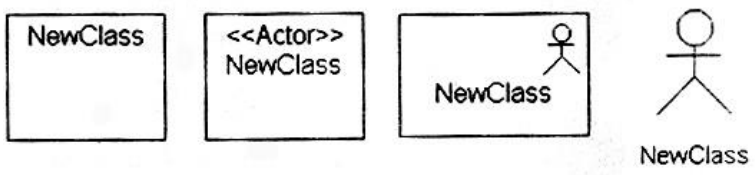

Рис. 68. Демонстрация различных вариантов визуализации стереотипов

Для демонстрации в предыдущем примере использован встроенный стереотип Actor.

В Rational Rose доступны следующие встроенные стереотипы:

- Actor (исполнитель);
- boundary (граница);
- business actor (бизнес-исполнитель);
- business entity (бизнес-сущность);
- business worker (работник);
- control (управление);
- entity (сущность);
- Interface (интерфейс).

Необходимость в стереотипах особенно ощущается при использовании диаграмм Use Case, разработка которых гарантирует, что система будет представлять собой именно то, что задумал пользователь, но и в диаграмме классов стереотипы могут использоваться для дополнительной детализации описания классов.

Контекстное меню класса

После добавления класса в диаграмму становится доступно контекстное меню класса. Содержание меню может изменяться при ассоциации класса с разными языками программирования. Пункты меню, относящиеся к языку программирования VC++, мы рассмотрим позднее, а сейчас ознакомимся с возможностями меню для класса, не ассоциированного с каким либо языком программирования (рис.69).

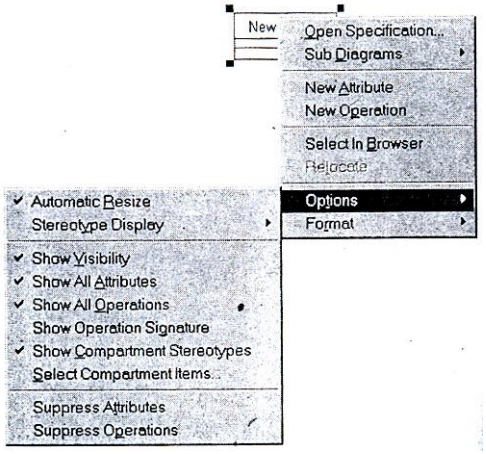

Рис.69. Контекстное меню

Перечислим назначение отдельных пунктов:

- Open Specifications — открытие диалогового окна заполнения спецификаций;

- Sub Diagrams - позволяет создавать к текущему классу диаграммы активности и состояний или перейти на диаграммы класса;
- New Attribute - позволяет добавлять новый атрибут класса;
- New Operation - позволяет добавлять новую операцию для класса;
- Select In Browser - позволяет выделить класс в окне;
- Relocate - позволяет переместить класс в новый контейнер или на местоположение;
- Options - вызов подменю настройки значка класса;
- Format - вызов подменю настройки шрифта, цвета, заливки диаграммы.

Меню Options (свойства)

Меню Options позволяют управлять отображением класса в диаграмме классов и состоит из следующих пунктов:

- Automatic Resize - автоматическая настройка размера значка, для того чтобы вместить весь введенный текст названия, атрибута или операции. Данная функция удобна для начального заполнения названий атрибутов и операций и включена по умолчанию. В дальнейшем, когда данный класс уже связан с другими и занимает свое место на диаграмме классов, ее можно выключить;
- Stereotype Display - позволяет показать или скрыть стереотип для данного класса;
- Show Visibility - позволяет показать тип доступа для операторов и атрибутов, таких как Public, Protected, Private, Implementation.
- Show All Attributes - показывает или скрывает атрибуты класса;
- Show All Operations - показывает или скрывает все операции класса;
- Show Operation Signature - показывает или скрывает так называемую сигнатуру операции, т.е. параметры и возвращаемое значение;
- Show Compartment Stereotypes - эта установка позволяет показывать или скрывать имя стереотипа для операции или атрибута класса;
- Select Compartment Items позволяет активизировать окно выбора пунктов Операций или атрибутов для показа, в том случае если нужно скрыть не все атрибуты или операции, а только некоторые. Для этого необходимо активизировать окно Select Compartment Items и выбрать необходимые для показа атрибуты и операции (рис.70);

При этом из левой части окна, где присутствуют все реквизиты, необходимо переместить в правую только те, которые необходимы для показа, после чего нажать ОК.

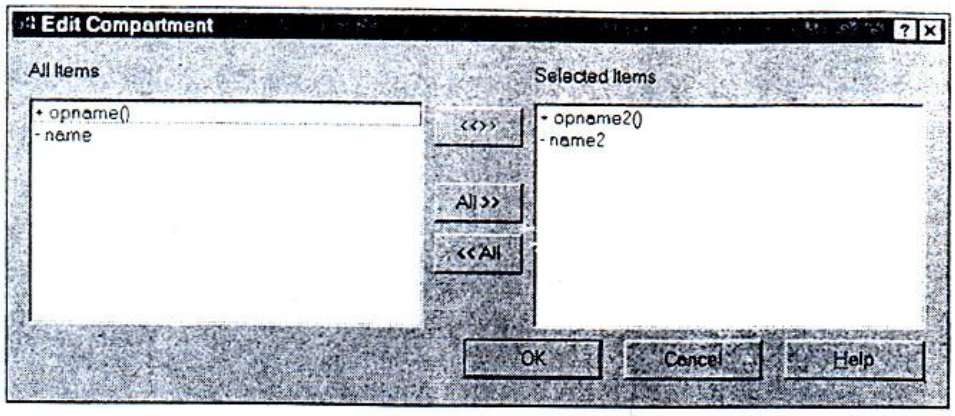

Рис.70. Установка необходимых для показа атрибутов и операций

- Suppress Attributes позволяет скрыть все атрибуты, даже если они были выбраны при помощи окна Select Compartment Items. Этот пункт интересен тем, что при его выборе не только скрываются атрибуты, но и закрывается пункт меню Attributes, что не позволяет ввести новые;
- Suppress Operations позволяет скрыть все операции аналогично атрибутам в предыдущем пункте.