

## Лабораторная работа № 10

### Спецификации класса

Rational Rose позволяет устанавливать значительное количество свойств класса, которые, в том числе, влияют на генерацию кода класса, поэтому, чтобы лучше ориентироваться в дальнейших действиях, разберем вкладку окна спецификаций. Возможно, контроллер параметров среды вам уже порядком надоел, поэтому для разбора возможностей спецификаций возьмем класс датчика температуры — TemperatureSensor.

#### Вкладка General (главная)

При выборе из контекстного меню датчика температуры пункта Open Specification открывается диалоговое окно, показанное на рис.71. Спецификации класса имеют несколько вкладок, и первой активизируется вкладка General (главная).

Это окно позволяет задать главные свойства класса, такие как его имя, тип, определить стереотип класса и доступ к нему, когда класс находится в контейнере. Так же как и во всех других диаграммах, здесь можно задать документацию к классу.

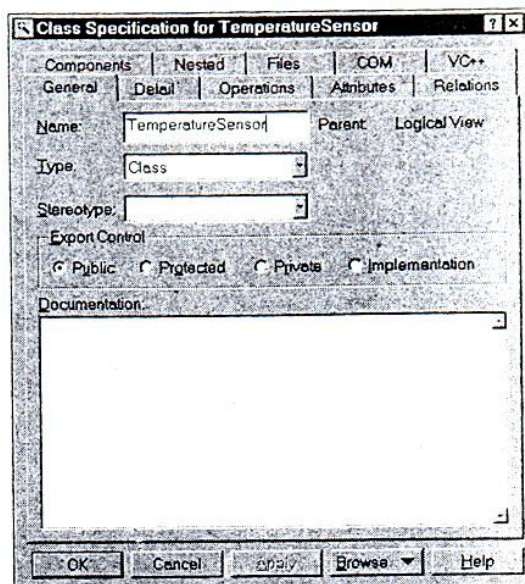


Рис.71.Вкладка General спецификаций датчика температуры

Перечислим поля, которые находятся на этой вкладке:

- Name предназначено для задания имени класса;
- Type предназначено для задания типа класса. В нашем случае - это «класс», но можно выбрать значение «метакласс», «параметризованный класса» и т.д.;
- Stereotype задает стереотип класса;
- Export Control предназначен для определения доступа к классу, когда он расположен в контейнере. При этом Public определяет, что элемент виден вне контейнера, в котором он определен и его можно импортировать в другие части создаваемой модели; Protected - элемент доступен только для вложенных классов, классов с типом friends и собственно внутри класса; Private обозначает защищенный элемент класса; Implementation – элемент виден только в том контейнере, в котором определен.

#### Вкладка Detail (детализация)

Вкладка Detail позволяет указывать дополнительные установки класса, такие как ожидаемое количество создаваемых объектов класса, ожидаемый расход оперативной памяти и т.д. (рис. 72).

Перечислим поля, которые находятся на этой вкладке:

- Multiplicity (множественность) позволяет задать ожидаемое количество объектов, которые будут созданы на основе данного класса. Обычно данный параметр удобно задавать для связанных классов;
- Space показывает количество оперативной памяти, необходимой для создания объекта данного класса. Поле может быть задано напрямую или формулой, описывающей требования по памяти и значение должно учитывать накладные расходы на создание объекта плюс размер всех объектов, входящих в данный;
- Persistence определяет время жизни объекта класса. Если установлен флажок Persistence, то объект должен быть доступен в течение всей работы программы или для доступа других потоков или процессов;
- Concurrency обозначает поведение элемента в многопоточковой среде. Установка такого поля в операции не должна противоречить установке в самом классе. Данная установка может принимать следующие варианты:
  1. Sequential (по умолчанию) — работа класса обеспечивается только для одного потока. Только один поток может быть запущен при помощи методов класса в один момент времени;
  2. Guarded - класс обеспечивает работу с несколькими потоками. Такой класс обеспечивает взаимодействие между потоками - для достижения непротиворечивой работы потоков, является арбитром потоков, предоставляя работу конкретному потоку в конкретный момент времени;
  3. Active - класс является классом отдельного потока;
  4. Synchronous - класс обеспечивает работу нескольких потоков, синхронизируя их.
- Abstract adornment обозначает, что класс является абстрактным, т.е. базовым классом, который должен быть наследован подклассами, и без такого наследования не имеет смысла. Также в классе могут быть определены абстрактные операции, которые представляют собой шаблон для создания операций в классах, наследуемых из данного, и сами не выполняют никаких действий.
- Format Arguments заполняется только для параметризованных классов и утилит классов. Для обычных классов данное поле недоступно.

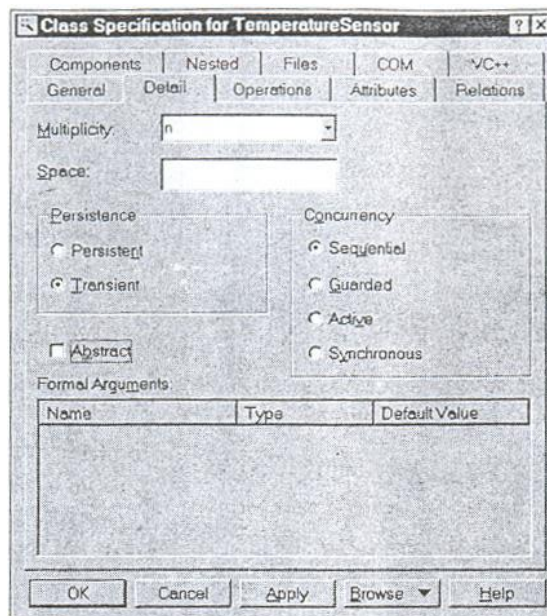


Рис.72. Вкладка Detail спецификаций датчика температуры

*Вкладка Components (компоненты)*

Вкладка Components отражает компоненты, с которыми ассоциирован класс (рис. 73). На вкладке помечены красным маркером компоненты, которые включены в текущую модель, и могут быть показаны остальные компоненты модели. Если у вас не показан компонент MFC 6.0, то или необходимо включить флажок Show all components, или еще не была импортирована библиотека MFC.

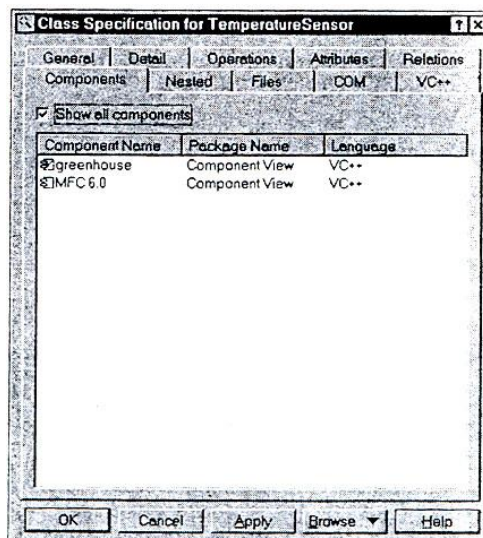


Рис. 73. Вкладка Components спецификаций датчика температуры

Из этой вкладки по двойному нажатию мыши на компоненте можно получить доступ к его спецификациям.

#### Вкладки Attributes (атрибуты)

Данная вкладка позволяет добавлять, удалять, редактировать атрибуты класса (рис.74)

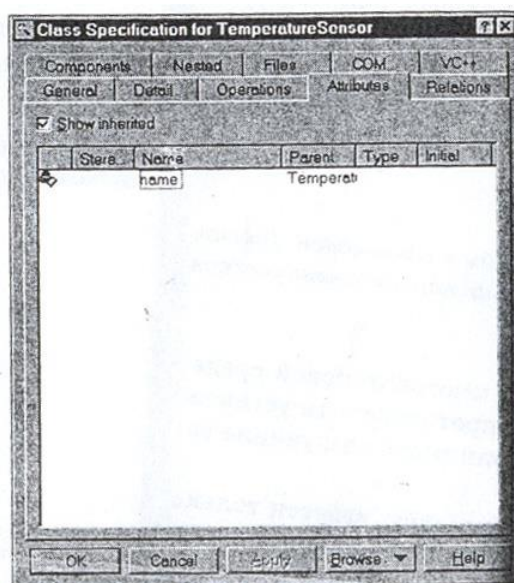


Рис.74. Вкладка Attributes спецификаций датчика температуры

На данной вкладке представлен список атрибутов класса, который можно редактировать при помощи контекстного меню. Флажок Show inherited позволяет скрыть или показать доступные атрибуты родительских классов.

Для того чтобы добавить атрибут, необходимо из контекстного меню выбрать пункт Insert. По двойному нажатию мыши на атрибуте или из контекстного меню Rational Rose предоставляет доступ к диалоговому окну спецификаций атрибутов (рис.75).



Здесь пользователь может изменить название атрибута (Name), его тип (Type) и стереотип (Stereotype) задать начальное значение (Initial value) и тип доступа атрибуту (Export Control).

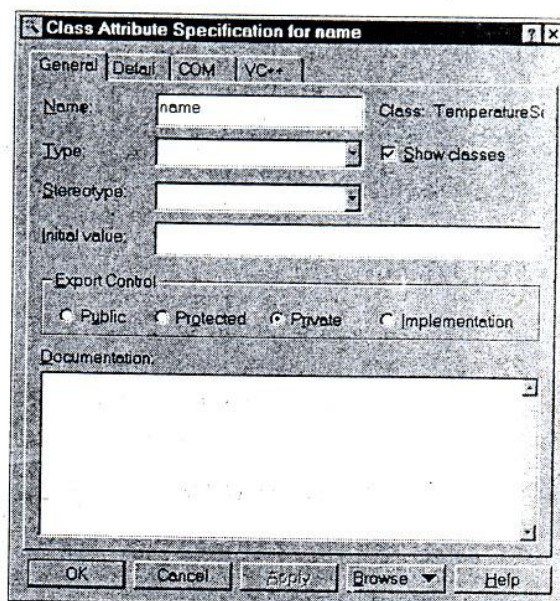


Рис. 75. Вкладка General спецификаций атрибутов класса

Дополнительная вкладка Detail спецификаций атрибутов класса (рис.76) позволяет задать тип хранения атрибута в классе:

- By Value- по значению;
- By Reference - по ссылке;
- Unspecified - не указано.

Также пользователь может указать, что атрибут является Static (статическим) или Derived (производным).

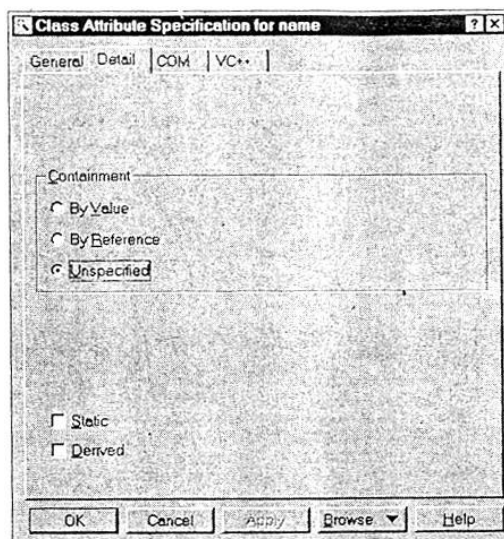


Рис. 76. Вкладка Detail спецификаций атрибутов класса

#### *Вкладка Operations (операции)*

Вкладка Operations позволяет добавлять, удалять, редактировать операции класса (рис.77).

На этой вкладке представлен список операций класса, который можно редактировать при помощи контекстного меню. Для того чтобы добавить операцию, необходимо из контекстного меню выбрать пункт Insert. По двойному нажатию

мышью на операции или из контекстного меню Rational Rose предоставляет доступ к диалоговому окну спецификаций операции.

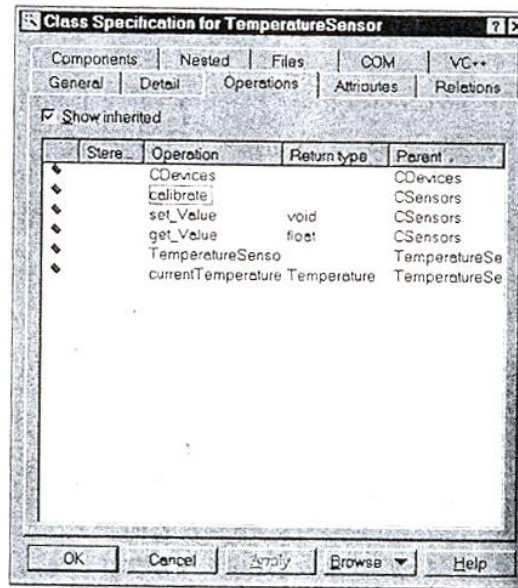


Рис. 77. Вкладка Operations спецификаций атрибутов класса

Вкладка Detail спецификаций операций позволяет устанавливать дополнительные свойства операции (рис. 78).

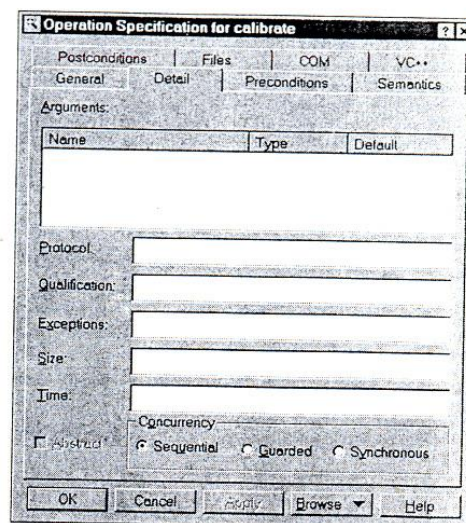


Рис. 78. Вкладка Detail спецификаций операций

Перечислим поля, представленные на этой вкладке:

- Arguments (аргументы) позволяет устанавливать список аргументов для операции с их типами и значениями по умолчанию;
- Protocol (протокол) позволяет задавать список установок, который предоставляется клиенту для вызова;
- Qualification (квалификация) позволяет идентифицировать зависящие от языка возможности, которые позволяют квалифицировать метод. Данная квалификация необходима, если вы используете Common Lisp Object System (CLOS);
- Exceptions (исключения) позволяет задавать список исключений, которые могут быть вызваны операцией. Здесь необходимо ввести имя одного или нескольких классов, обрабатывающих исключительные состояния;
- Size (размер) позволяет задать размер памяти, требуемой для выполнения операции;

- Time (время) позволяет задать время выполнения операции;
  - Concurrency (конкуренция) отражает для многопоточковой программы тип выполнения операции:
1. Sequential (по умолчанию) — только один поток должен выполняться а одно и тоже время;
  2. Guarded - возможно выполнение нескольких потоков, управляемых некоторым классом;
  3. Synchronous - операции могут идти параллельно.

Пользователь может устанавливать конкуренцию выполнения для класса в спецификациях класса. Это поле недоступно для редактирования утилит классов, параметризованных классов и реализаций утилит классов.

Вкладки Preconditions, Postconditions, Semantics, позволяют задавать дополнительные описания процессов подготовки и завершения операции, а также описание алгоритма операции. Кроме текстового описания здесь же можно задать имя Interaction диаграммы, которая описывает указанные действия.

#### *Вкладка Relations (связи)*

Вкладка Relations добавлять, удалять, редактировать связи класса (рис.79).

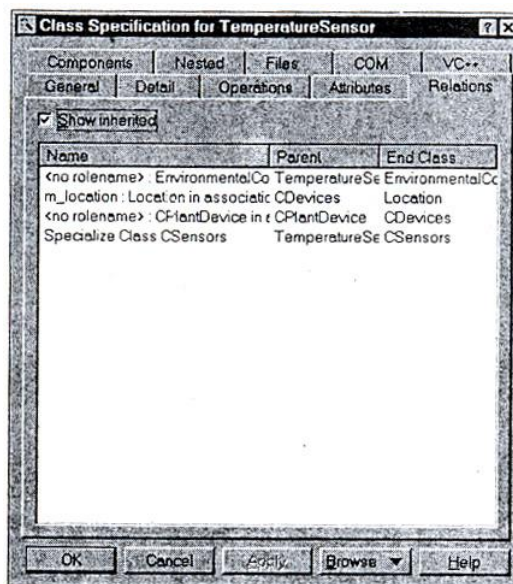


Рис.79.Вкладка Relations спецификаций класса

На данной вкладке представлен список связей класса, который можно редактировать при помощи контекстного меню. Для добавления связи лучше всего воспользоваться соответствующим инструментом из строки инструментов, а для удаления — контекстным меню.

#### *Вкладка VC++*

Вкладка VC++, которая появилась после ассоциации класса с языком Visual C++, предназначена для изменения свойств, связанных с данным классом. Поля данной вкладки не предназначены для редактирования поэтому не будем их рассматривать.

#### *Вкладка COM*

Вкладка COM позволяет устанавливать свойства для классов, которые предназначены для создания COM объектов в модели. В случае если такие объекты импортируются в модель, в них также появляется такая вкладка (рис. 80).



Рис.80 Вкладка COM спецификаций класса

- Generate - свойство, определяющее необходимость генерации исходного кода класса;
- kind - свойство, которое идентифицирует тип объекта, который может быть следующим: coclass, enum, record, module, interface, dispinterface, alias, union, max;
- uuid - свойство которое задает строку идентификатора для класса или библиотеки, которая идентифицирует COM объект в системе;
- version - версия COM объекта;
- helpstring, helpcontext предназначены для задания строки и идентификатора файла помощи для объекта;
- attributes - задает такие атрибуты для объекта как control, hidden, restricted, licensed, appobject, nonextensible или oleautomation;
- dllname - задает имя файла DLL, в котором находится объект;
- alias - задает тип, для которого создается псевдоним.