

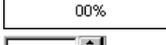
Моделирование в Stratum Jet

Оглавление

ОГЛАВЛЕНИЕ	2
ВВЕДЕНИЕ	6
МОНИТОРИНГ.....	6
МОДЕЛИРОВАНИЕ.....	6
ИССЛЕДОВАНИЕ.....	7
УПРАВЛЕНИЕ.....	7
ПРОЕКТИРОВАНИЕ.....	7
ОРГАНИЗАЦИЯ.....	8
ОБУЧЕНИЕ И ТРЕНИНГ.....	8
ИНСТАЛЛЯЦИЯ СИСТЕМЫ	8
РЕГИСТРАЦИЯ СИСТЕМЫ	9
ПРАВОВАЯ ИНФОРМАЦИЯ И СОГЛАШЕНИЕ НА ИСПОЛЬЗОВАНИЕ	9
ИНТЕРФЕЙС STRATUM FOR WINDOWS 10	
MENU ITEMS.....	11
TOOL BARS.....	11
АРХИТЕКТУРА ПРОЕКТА	11
ИМИДЖИ И ОБЪЕКТЫ.....	11
СОСТАВИМИДЖА.....	11
БИБЛИОТЕКА КЛАССОВ.....	12
СОЗДАНИЕ ПРОЕКТА	12
СХЕМА, ЛИСТ, VDR-ФАЙЛ	13
РЕДАКТИРОВАНИЕ ГРАФИЧЕСКИХ ОБЪЕКТОВ НА СХЕМЕ.....	14
<i>Draw a line</i>	14
<i>Draw a rectangle</i>	14
<i>Draw a Rounded Rectangle, Circle, Sector and Arc</i>	14
<i>Insert a text</i>	14
<i>Insert a PolyLine</i>	14
<i>Insert a Picture</i>	15
<i>Changing objects properties</i>	15
<i>Selecting objects</i>	15
<i>Grouping objects</i>	15
<i>Moving and Copying objects</i>	15
<i>Deleting objects</i>	15
<i>Scaling and Rotation of objects</i>	15
<i>Editing polylines</i>	16
<i>Views of a Scheme</i>	16
ДОБАВЛЕНИЕ ИМИДЖЕЙ НА СХЕМУ.....	16
ЗАПУСК МОДЕЛИ СХЕМЫ НА ВЫЧИСЛЕНИЕ.....	18
ПРОВЕДЕНИЕ СВЯЗЕЙ.....	19
ИЗМЕНЕНИЕ ПОРЯДКА ВЫЧИСЛЕНИЙ.....	20
УСТАНОВКА ПАРАМЕТРОВ СХЕМЫ (ПЕРЕМЕННЫХ ИМИДЖЕЙ).....	21
<i>Установка параметров по умолчанию</i>	22
<i>Установка начальных значений параметров</i>	22
<i>Установка значений всех параметров проекта с одинаковым именем</i>	22
<i>Запись/чтение состояний модели</i>	22
МОДЕЛЬ. ИМИДЖ	22
АРХИТЕКТУРА СЧЕТА.....	23

ОСТАНОВКА МОДЕЛИ ПО УСЛОВИЮ.....	26
УПРАВЛЕНИЕ РАБОТОЙ ОТДЕЛЬНЫХ ИМИДЖЕЙ МОДЕЛИ.....	26
ПЕРЕМЕННЫЕ ИМИДЖЕЙ МОДЕЛИ.....	26
<i>Типы переменных.....</i>	<i>26</i>
FLOAT.....	27
COLORREF.....	27
HANDLE.....	27
STRING.....	27
LOCAL.....	27
<i>Предопределенные переменные.....</i>	<i>27</i>

СТАНДАРТНЫЕ БИБЛИОТЕКИ ИМИДЖЕЙ 28

БИБЛИОТЕКА UNIT.....	28
 <i>GraphicSpace</i>	28
 <i>GraphicSpaceEx</i>	28
 <i>LGSpace</i>	28
 <i>OSCSpace2D</i>	29
 <i>Gistogramma</i>	29
 <i>MessageBox</i>	29
 <i>InputDialog</i>	29
 <i>FileDialog</i>	30
 <i>ColorDialog</i>	30
 <i>MatrixView</i>	30
 <i>TableEdit</i>	30
 <i>VKeyState</i>	31
 <i>MouseState</i>	31
 <i>ChangeModel</i>	31
 <i>FileString</i>	31
 <i>Table</i>	31
 <i>BaseOpen</i>	31
 <i>DrawFile</i>	32
БИБЛИОТЕКА CONTROLS.....	32
 <i>WinButton</i>	32
 <i>Button</i>	32
 <i>Lamp</i>	32
 <i>Slider</i>	32
 <i>Gauge</i>	33
 <i>NumberIn</i>	33
 <i>NumberView</i>	33
 <i>StringEdit</i>	33
 <i>ComboBox</i>	33
 <i>Check Box</i>	33
 <i>Radio Button</i>	33
БИБЛИОТЕКА GRAPH2D.....	34
1. <i>Графические объекты</i>	34

	Line.....	34
	Arrow.....	34
	Triangle.....	34
	Rectangle.....	34
	Polyline.....	34
	Pie.....	34
	Ellipse.....	35
	Text.....	35
	Picture.....	35
	Video.....	35
	ColorBar.....	35
	2. Вспомогательные илджи.....	35
	Object2d.....	35
	ShowObject.....	35
	ScanObject.....	35
	DragObjects.....	36
	Color.....	36
	БИБЛИОТЕКА GRAPH3D.....	36
	1. Илджи перемещения, вращения и масштабирования.....	36
	Move3d.....	36
	Resize3d.....	36
	Rotate3d.....	36
	ObjectColor3d.....	36
	2. Трехмерные примитивы.....	36
	Bar3D.....	36
	Sphere3D.....	37
	Cylinder3D.....	37
	Cone3D.....	37
	Tube3D.....	37
	Tore3D.....	37
	Pyramid3D.....	37
	Custom3D.....	37
	Teapot3D.....	37
	Grid3D.....	37
	3. Трехмерная визуализация.....	37
	Material3d.....	37
	OSC3D.....	37
	MakeFace3D.....	37
	Histogramm3D.....	37

4. Трехмерные функции – имиджи.....	38
БИБЛИОТЕКА NETWORK.....	38
 Network_Client_Object.....	39
 Network_Server_Object.....	39
ГИПЕРССЫЛКИ	39
СОЗДАНИЕ ГИПЕРССЫЛКИ.....	39
ОТКРЫТИЕ ОКНА.....	40
НАСТРОЙКА ПАРАМЕТРОВ СТРАНИЦЫ.....	41
Группа параметров «Scroll».....	42
Группа параметров «Style».....	42
Группа параметров «Size».....	42
НАСТРОЙКА ПАРАМЕТРОВ ГИПЕРССЫЛКИ.....	43
Выполнение приложений Windows.....	43
Запуск другого проекта.....	43
Ничего не делать.....	45
Системная команда.....	45
ИСПОЛЬЗОВАНИЕ ФУНКЦИЙ ГИПЕРПЕРЕХОДА.....	45

Введение

Среда предназначена для мониторинга, моделирования, исследования, управления, проектирования, организации систем, а также обучения и тренинга.

Использование среды полезно при быстром прототипировании будущих систем, получении принципиального ответа на вопрос о работоспособности идеи, так как позволяет быстро собрать и модифицировать систему, определить принципиальную возможность ее реализации.

Особенно целесообразно использование среды там, где нет готовой системы автоматизации. При этом пользователь должен иметь представление о будущей системе, знать ее законы функционирования, чтобы построить самому ее вычислительную модель. Среда не предполагает знание пользователем программирования для построения компьютерного прототипа системы.

Среда является интерпретатором математических, алгоритмических и структурных выражений. Среда является средством проектирования и моделирования. Среда имеет в своем составе набор редакторов. Объекты, создаваемые редакторами, сохраняют свою структуру (свойства), интерпретируются средой. Объекты модифицируются, что позволяет легко изменять первоначальный проект.

Мониторинг

С помощью этой среды вы можете собирать информацию с периферийных устройств. Это достигается подключением к PC каких-либо систем сбора информации, например, P-NET (фирмы Process Data, протокол P-NET), STEP7 (фирмы Siemens, протокол MPI), Internet (протокол TCP/IP). В среде устанавливаются соответствующие имиджи, дающие возможность видеть переменные этих устройств. Таким образом, из модели, построенной пользователем, можно легко управлять любым из устройств, подключенным к системе сбора информации, или получать информацию от него.

Если оборудование имеет протоколы, отличные от перечисленных, то следует иметь соответствующие dynamic link library (dll) для подключения новых систем сбора информации.

Моделирование

С помощью этой среды вы можете анализировать собранную информацию, обработать ее, задав собственный алгоритм или формулы для этого. С помощью этой среды Вы можете построить модель любого реального или планируемого объекта, системы, задачи или процесса.

Конструирование достигается средствами визуального проектирования элементов системы. Среда имеет достаточно широкий набор таких элементов, а также инструментов для их производства в случае необходимости его расширения самим пользователем.

Среда позволяет собирать самые сложные системы, соединяя элементы связями в необходимые структуры. Допускается также иерархическая сборка подсистем из элементов как вверх, так и вниз.

Среда позволяет легко модифицировать собранную систему, изменив направление связей, содержание элементов, их структуру. Модификация компьютерной модели систем позволяет без программиста подобрать режим их наилучшего

функционирования, структуру. Для этого пользователю при проектировании достаточно придерживаться обычного объектного конструкторского мышления.

Среда рассчитывает содержимое элементов, организует влияние их по связям друг на друга и, таким образом, интерпретирует поведение и свойства всей системы, наглядно отражает поведение и свойства в двухмерном и трехмерном пространстве.

Исследование

С помощью этой среды вы можете исследовать поведение систем. Для этого имеется возможность изменять начальные условия системы, менять любые параметры и переменные в ходе расчета, изменять структуру за счет модификации связей, количества и содержимого элементов. Вы можете управлять моделью системы программно, данными из файла, вручную через установленные интерфейсные элементы, с помощью других моделей или обратных связей. Есть возможность взаимодействия моделей друг с другом, в том числе находящихся на разных РС, в локальной или глобальной сети. Управление может происходить от периферийных устройств и датчиков.

Можно исследовать свойства систем и процессы, многократно запуская расчет проектируемой системы и сравнивая результаты между собой для нахождения наилучшего варианта.

Результаты моделирования можно отражать графически, запоминать или выдавать на периферийные устройства.

Управление

С помощью этой среды вы можете управлять реальными системами. Для этого, установив типовые элементы связи с периферийным оборудованием на своем проекте, результаты расчетов можно подавать на входы промышленного оборудования как управляющие сигналы. Управление может быть реализовано как автоматическое или под наблюдением человека оператора, который наглядно контролирует процесс на экране и может вмешаться в процесс управления.

Обратная связь позволяет построить реальную систему управления внешним объектом, а средства визуализации и записи организовать мониторинг любого сложного объекта, в том числе распределенного или удаленного.

Управление может производиться с нескольких точек, различных РС и локальной или глобальной сети.

Проектирование

С помощью этой среды Вы можете сами проектировать компьютерную реализацию будущей системы. Для проектирования достаточно представлять себе примерно – как будет выглядеть будущая система, и рисовать ее структуру на листе проекта, наполняя по необходимости отдельные элементы содержанием. Создавать надо только оригинальные элементы системы, так как к услугам пользователя имеется множество стандартных элементов. Элементы могут иметь любое количество переменных, описание (текст модели), изображение (двухмерное, трехмерное), данные и состоять из схемы в виде новых элементов и связей между ними.

В среде есть механизмы сообщений, доступ к базам данных и файлам, методы расчета стандартных моделей, области пространства, структуры данных и другие полезные инструменты.

Элементы могут быть подвижными или статичными. Элементы могут связываться друг с другом связями, сообщениями, через файлы, матрицы и динамические структуры данных.

Для описания содержимого элементов пользователю достаточно математических средств и функций (более 2000). При наличии навыков программирования можно использовать запись на языке подобном СИ++ или писать дополнительные к среде модули dll.

Организация

С помощью этой среды Вы можете сами организовать свой труд. Среда позволяет создать прототипы органайзера, контролировать ход проектирования. В ходе работы Вы создадите собственный конструктор элементов, который будет полезен для последующих проектов. Неотъемлемое свойство среды – обязательное уменьшение трудоемкости каждого следующего проекта за счет использования ранее созданных элементов. Предусмотрена возможность комментирования проекта.

Среда хорошо сочетается с другими продуктами под Windows – редакторами графики, видео, текста и так далее. Проект можно соединять с другими проектами посредством гиперссылок, в том числе на другие программные средства. В проект можно включать средства мультимедиа – видео, графику, звук. В сравнении с типовыми проигрывателями мультимедиа, среда позволяет управлять объектами мультимедиа.

Обучение и тренинг

Среда позволяет использовать модели для тренажа персонала. Многократное исследование модели в различных режимах обеспечивает лучшее понимание свойств объектов их пользователями.

Моделирование на сети позволяет проводить деловую игру с любым количеством участников.

Правильно оформленный проект, дополненный средствами навигации, помощи и тестирования, может служить целям обучения.

Инсталляция системы

Система Stratum Jet имеет три варианта поставки:

1. поставка на дискетах 3,5" (4-5 дискет, в зависимости от версии);
2. поставка на компакт диске (1 диск);
3. поставка через Internet (файл установки).

Независимо от формы поставки используется единая программа установки. Чтобы начать установку системы, найдите и запустите файл **setup.exe**. В появившемся диалоговом окне программы установки укажите устройство и папку (Destination Directory), в которую вы предполагаете установить систему. Дополнительно программа установки может вывести ярлык на рабочий стол, для этого поставьте флажок в пункте Put shortcut on the desktop. Далее программа установки автоматически скопирует

необходимые файлы и создаст в меню Пуск ярлыки для запуска программы. Система готова к работе.

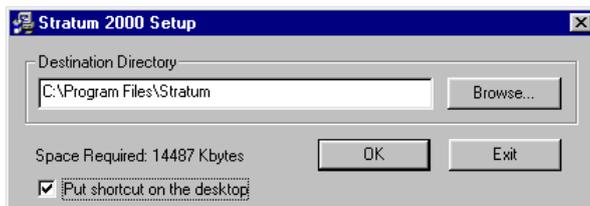


fig. 1 Диалоговое окно программы установки

Регистрация системы

При первом запуске системы появится диалоговое окно About, в котором, нажав кнопку Register, можно зарегистрировать свою копию системы. Если вы пользуетесь не зарегистрированной копией, то система через определенные промежутки времени будет предлагать зарегистрироваться. Для регистрации необходимо заполнить все поля окна регистрации.

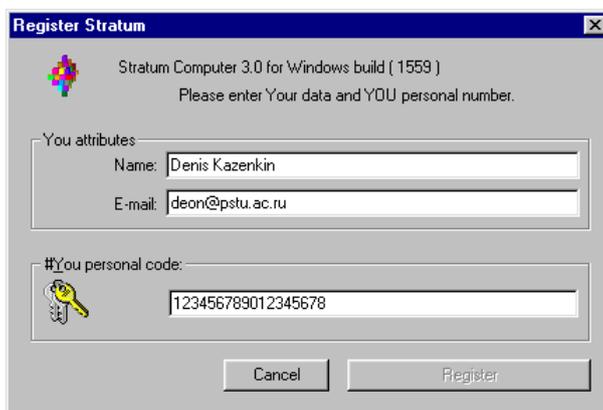


fig. 2 Окно регистрации системы

Если ваш персональный код введен верно, то кнопка Register станет доступна и систему можно зарегистрировать.

Правовая информация и соглашение на использование

Авторское право от 1991 г. ©Stratum Computer. Государственный регистрационный номер №50910000204.

Авторское право от 1998 г. ©Stratum for Windows. Государственный регистрационный номер №980665.

Авторы системы:

Мухин Олег Игоревич

Носков Владимир Анатольевич

Шелемехов Александр Анатольевич

Все права сохранены. Программный продукт запрещается воспроизводить, копировать или распространять в любом виде или любыми средствами целиком или частично без письменного разрешения авторов. Авторы предупреждают о своем праве защищаться

от несанкционированного копирования как компьютерными средствами, так и в судебном порядке в соответствии с авторским правом. Программное обеспечение может быть установлено одновременно только на том количестве компьютеров, которое описано в поставке. Перенос продукта на другие технические средства должен предусматривать демонтаж системы с предыдущего места базирования.

Продукт продан по системе “как есть”. Мы попытаемся устранить ошибки, которые Вы обнаружите в системе, если Вы обеспечите способ воссоздать проблему на компьютерах, специфицированных в пункте Технические требования данного руководства.

Зарегистрированный пользователь имеет право на консультации по системе. Для регистрации Вас как пользователя заполните и вышлите карточку регистрации в наш адрес. Региональный центр информатизации Пермского государственного технического университета осуществляет обучение и сертификацию пользователей Stratum по отдельному договору.

Авторы не предполагают ответственности и обязательств любого рода за последствия ошибок, вызванных неправильной работы системы.

Авторы оставляют за собой право изменять с целью улучшения потребительских свойств как текст публикации, так и программный продукт без дополнительного персонального предупреждения пользователей (сохраняя при этом преемственность программного обеспечения в рамках одной операционной среды).

Интерфейс Stratum for Windows

To start Stratum Jet simply click the Start Button (it is placed in the bottom left corner of the screen), move the mouse pointer on the line Programs , then on the line Stratum for Windows. Now select the Stratum Jet application by clicking the left mouse button on it. The screen you have got looks as follows:

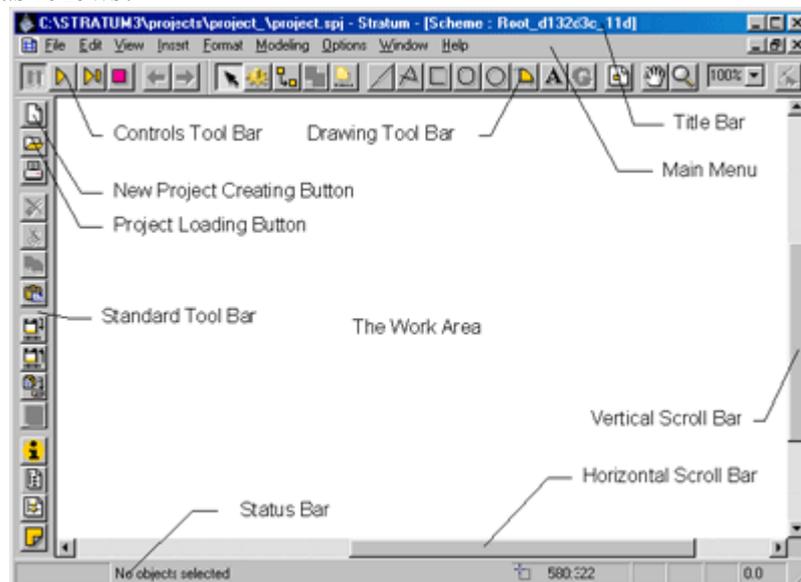


fig. 3 Окно программы и интерфейсные элементы

The interface consists of the Title Bar, the Menu Bar and the Tool Bar also known as the Button Bar and Work Area.

Menu Items

Click the menu name to open it and look through its options. Click the option name to choose the option.

File – Commands for the projects and files: create, open, save, print and other.

Edit – Commands for editing graphics and texts.

View – Commands for interface control and show of special windows.

Insert – Commands for new objects creation on the scheme and for the objects import.

Format – Commands for editing various project objects.

Modeling – Commands for the control of modeling.

Options – Commands for set of environment and project preferences .

Window – Commands for control of windows appearance and layout.

Help – Help system.

Tool Bars

The Tool Bar is below the Menu Bar over the Work Area. These buttons make your work with Stratum Jet easily. Just position the mouse cursor in front of the button to get the descriptions of its functions.

Standard Tool Bar – contain the most often used commands – load/save, edit, show in special windows.

Drawing Tool Bar – contain tools used for drawing.

Control Tool Bar – contain commands controlling start, pause and stop of simulation.

Архитектура проекта

Моделируемая система состоит из элементов и связей. Система создается на листе проекта.

Схема – модель системы, имиджи, соединенные связями.

Имидж – модель элемента (может быть схемой).

Лист – двухмерное графическое пространство для изображения схемы.

Проект – совокупность файлов для обеспечения функционирования схемы.

Схема может быть объявлена имиджем и использована в составе схемы верхнего уровня.

Имиджи и объекты

Следует отличать имиджи и объекты. Объекты – это графические, звуковые фрагменты, возможно файлы, базы данных и так далее. Имидж – это модель, выраженная в виде текста, который может быть исполнен средой, проинтерпретирован ею. Имидж может управлять объектом.

Состав имиджа

Модель элемента (имидж) содержит текст модели, перечень использованных для этого переменных и их значения, изображение элемента (типовая иконка или оригинальное

двухмерное изображение), ссылку на подсхему, ссылку на другой элемент, описание, название, является объектом с номером и координатами на листе проекта.

Библиотека классов

Созданные однажды вами имиджи автоматически хранятся в библиотеке проекта. Имидж, установленный на схему проекта – экземпляр класса. Все экземпляры одного класса имеют одинаковое имя. Изменение текста или изображения класса ведет к аналогичным изменениям во всех экземплярах класса. Экземпляр, установленный на схеме, имеет собственные уникальные данные.

В стандартной поставке среды содержатся стандартные библиотеки (см. Стандартные библиотеки)

Создание проекта

Проект – целостная конструкция, реализующая идею пользователя. Создается посредством манипуляций пользователя по проектированию его составляющих. Проект создается на рабочем поле. Одновременно может быть загружен только один проект. Отдельный проект рекомендуется помещать в отдельную директорию. К проекту может относиться ряд файлов – vdr, stt, mat, графические, базы данных. Структура проекта записывается в файле *.spj. Проект состоит из листов. На листе может располагаться одна схема или (и) один (или несколько) vdr-файлов.

Для того, чтобы создать новый проект, выполните команду File | New | Project.

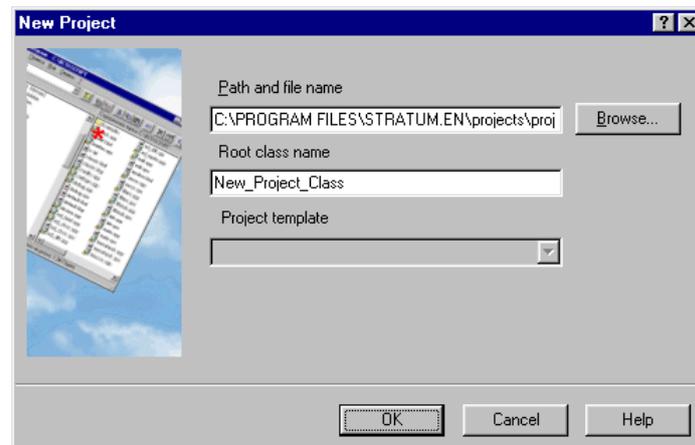


fig. 4 Диалоговое окно создания проекта

В появившемся окне New project кнопкой Обзор выберите директорию, в которой будет располагаться Ваш проект. Можете дать файлу, в котором будет храниться проект, имя (по умолчанию project) с расширением spj. Так же выберите имя для главного листа проекта (Root class name).

Если имя и директория вам не принципиальны, на инструментальной панели системы Stratum нажмите кнопку  – "Быстрое создание нового проекта". Появится рабочее поле, подписанное сверху "Схема: ...".

Для сохранения проекта вызовите File | Save All. Проект будет сохранен. Когда необходимо Вы сможете снова загрузить изготовленный проект. После запуска

системы в верхней строке вызовите File | Open. Если проект использовался недавно, то его можно открыть из списка последних проектов в меню File.

Схема, лист, VDR-файл

Проект состоит из совокупности схем, каждая из которых, отображается в отдельном окне. Одна из схем обязательно корневая (главная). Схема содержит модель и графические объекты для ее визуализации. Графические объекты в рамках одной схемы объединены в лист, или, говорят, расположены на листе. Лист обладает индивидуальными свойствами – размеры листа, фон, разметка, система координат и так далее. Схема существует только в рамках проекта. Лист может являться частью схемы, но может выступать как отдельный элемент и храниться в графическом vdr-файле. Обычно понятия схема и лист рассматриваются как идентичные и взаимозаменяемые, хотя, точнее, лист – это графическое пространство для расположения схемы и (или) объектов.

Графические объекты имеют форму, тип, имя, графические инструменты, используемые для их создания, положение, стиль, цвет и ряд других свойств. Объект визуально проявляется на окне схемы. Внешний вид и положение графических объектов может изменяться из модели. Объекты могут объединяться в группы. Группа, в свою очередь, есть тоже графический объект.

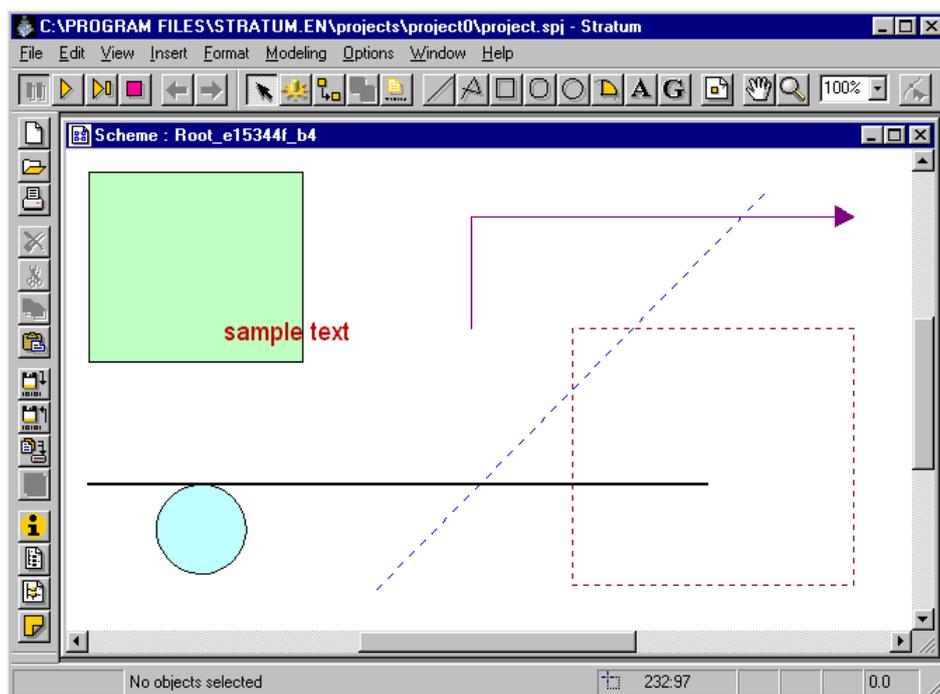


fig. 5 Пример листа схемы с двухмерными объектами

Для создания отдельного листа в меню “Файл” выберите пункт “Новый” и далее подпункт “Векторный рисунок (vdr)”. Откроется окно листа, внешне схожее с окном схемы. Характерной особенностью vdr-файла является то, что его объекты являются только рисунком. Свойства объектов vdr-файлом не интерпретируются (для этого нужна схема).

Внимание !

Разделив схему и лист, в последствии вы не сможете создать на листе с vdr-файлом схему. Потребуется дополнительные усилия по обращению схемы к объектам vdr-файла. Неопытным пользователям рекомендуем создавать лист и схему одновременно: File | New | Project.

Отдельный лист или лист схемы можно сохранить в графический файл (vdr-файл) с помощью команд Save active window or Save active window as из меню File.

Информация по обработки графики из следующей главы применима как к листам, так и к схемам.

Редактирование графических объектов на схеме

Use the **Drawing Tool Bar** to draw something on the scheme:

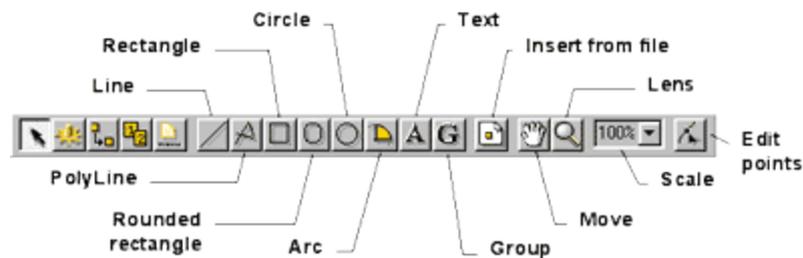


fig. 6 Drawing Tool Bar

Draw a line

Press the line creation button  on the toolbar. The mouse pointer is changed, showing that now we can draw a line. Click the left mouse button and holding it down draw a line. Then release the button.

Draw a rectangle

Press the rectangle creation button  on the toolbar. The mouse pointer is changed, showing that now we can draw a rectangle. Click the left mouse button and holding it down draw a rectangle. Then release the button.

Draw a Rounded Rectangle, Circle, Sector and Arc

Draw a Rounded Rectangle, Circle, Sector and Arc using the same operations as stated above.

Insert a text

Press the text creation button  on the toolbar. The mouse pointer is changed, showing that now we can insert the text. Click the left mouse button to add the text on the scheme. The Text dialog box is open. Here you can input your text and change its properties. Then press Ok button.

Insert a PolyLine

Press the polyline creation button  on the toolbar. The mouse pointer is changed, showing that now we can draw a polyline. In order to edit the points of the polyline click the left mouse button first (the object is getting selected) and click the right mouse button. Then choose the Edit points line. Now you can edit any point you want.

Insert a Picture

Choose the Insert on the main menu and then the line From File or press the **insert from file** button . The dialog box Open file is open. Now choose the picture you like. The *.bmp, *.gif, *.jpg, *.pcx, *.wmf and other graphic formats are supported.(You can also insert picture from the file by clicking the right mouse button on the scheme and choosing the line Insert From File or press the Insert From File button on the control panel).

Changing objects properties

To change object properties you can double-click on the object or click the right mouse button and then choose the line Object Properties as well. The Properties of 2d object dialogue is open. Here you can change sizes, coordinates, color and other properties. Then press Ok button.

Selecting objects

To select a group of objects, position the mouse pointer in front of the first object you wish to select and then press and hold down the left mouse button. Now drag the mouse until the black highlighting has covered all the objects you wish to select before releasing the left button.

Grouping objects

Press the group creation button  on the toolbar. The mouse pointer is changed, showing that now we can create a group. Now pressing and holding down the left mouse button select objects. Then release the button.

Moving and Copying objects

In order to move or copy a selection of objects, we make use of what is known as the paste buffer or clipboard. Select an object and then click the left mouse button on Edit – Copy: This copies the selected object to the clipboard. Then, move the mouse pointer to anywhere you like on your scheme and select Edit – Paste. You can paste the object from the clipboard as many times as you wish to produce multiple copies of that object. The process of moving object is much the same, except that you use Edit – Cut instead of Copy. Note that the editing buttons illustrated on the left can be used instead of using the menu commands.

There is another way. Position the mouse pointer in front of the object or the group of objects. Now drag the mouse pointer to anywhere on your scheme when pressing and holding down the left mouse button. Then release the left button. The selected object is moved to the new place. This process is called as drag-and-drop. To copy an object position the mouse pointer in front of it. Then move it anywhere you like when pressing the [Ctrl] key and holding down the left mouse button.

Deleting objects

Select the object or the group of them and press the [Delete] key to remove them altogether.

Scaling and Rotation of objects

Click the left mouse button on any object, the frame will appear:

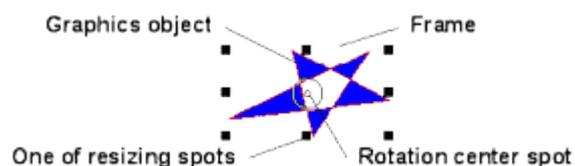


fig. 7 Scaling and rotation of object

In order to change the size of the object position the mouse pointer in front of one of resizing points. Then move the mouse pointer holding down the left mouse button. Now release the button. The size of the object is changed.

To rotate the object press the [SpaceBar] key on the keyboard. The spot of rotation center appears. By default the spot is placed in the frame centre. But you can move the centre to another place holding down the left mouse button. Now you can turn the object using drag-and-drop on frame spots.

Editing polylines

Select object – polyline. Press the edit points button  on the toolbar. All points will be marked by small squares. You can move this points using the drag-and-drop operation. If you need to add a point, press the [Ctrl] key and holding it down click the left mouse button on the required polyline segment. To delete points press the [Shift] key and holding it down click the left mouse button at the required point.

Views of a Scheme

There are many ways in Stratum Jet to view your scheme during the editing process.

Move button  – enables a fast review of scheme mode. To disable the mode press this button once again or click the right mouse button.

Lens button  – enables fast rescaling. Click the left mouse button to zoom the scale, and the right – to decrease it. To disable the mode press this button once again.

Scale Box – enables inputting the exact scheme's scale.

Добавление имиджей на схему

Scheme is known as a set of information the linked objects. So it maps the researched system structure. Object is a prototype of an object from the real world, model, or template. We divide system into some elements to deal easily with them. That's why it is useful to split complex systems. The splitting of the system into units should be made so that it would be possible to compound these units in other combination in another system. And to make that the main amount of links remain inside elements. It permits to achieve the greatest integrity of elements, which can be used later as complete objects. But mind 2 extremes! It is much more easily to test elementary objects, but much more difficult to set them into one system – because of links amount. The objects, which are too large, are also inconvenient – it's difficult to create something new of their combinations or to update the system without changing components.

The Drawing toolbar is applied to operation with scheme elements.



fig. 8 Drawing toolbar

Press the new object creation button  on the toolbar. The Class properties dialogue is open. For the first time you do not have to fill in everything, but only the most important.

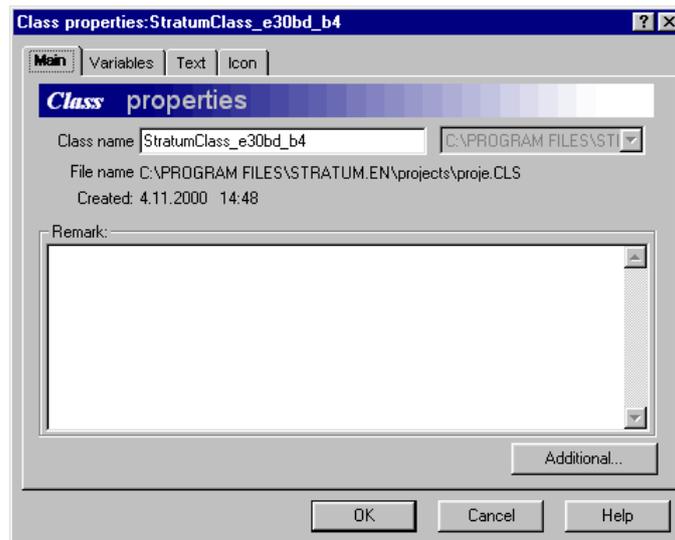


fig. 9 Class properties dialogue. Main

Class name

In the line "Class name" type for example linear_func.

Class text

Click the left mouse button (click) on the tab "text" then click the text field. In the first line you can type:

$X := X + 1$

then press ENTER. In the next line type:

$Y := \sim X * 2 + 2$

Pay attention to characters := "to assign" and ~ "Tilde"!

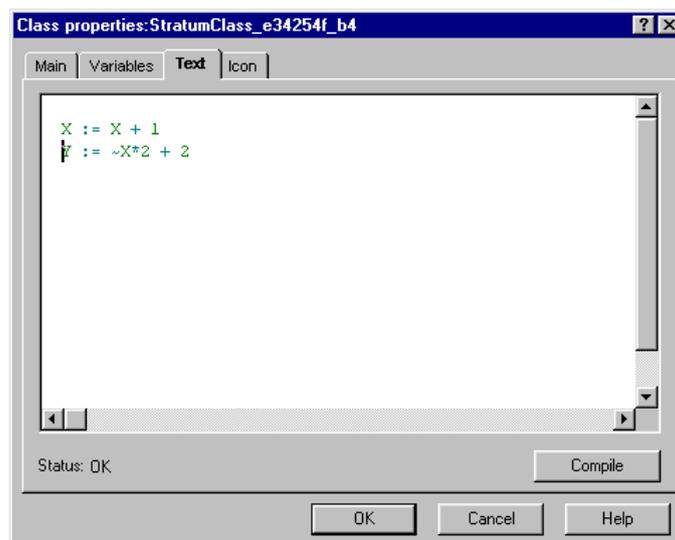


fig. 10 Class properties dialogue. Text

The first line will change X value: -10, -9,..., 8, 9, 10,... to one every step until the project is not stopped (by the Stop button).

The second line will change Y value depended on a current X value. Values of X and Y will be calculated at every step.

Click the Compile button. The message OK will appear, else you have misprinted. Check the text and correct the error. Then press OK on the dialogue box bottom, it will disappear and class picture, containing your expressions, will arise on the project worker field.

Initial value. For correct calculating the initial value of X is needed. Click the right mouse button (right-click) on the class icon, click "Class properties" and then click tab "Variables". You will have the list of variables. Click cell on "Data" column and the X line intersection. Type there -11. Now press the OK button. You are ready for calculating.

Запуск модели схемы на вычисление

For calculation of the scheme the Control toolbar is used.

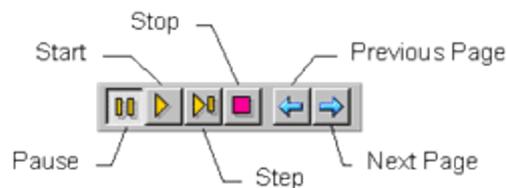


fig. 11 Control toolbar

Press the Step button. System does one step and stops after it. The system calculates all the expressions of the scheme at every step. After a step the system can be frozen in a pause state for any long time. Check this step results. Right-click the object icon. Put the mouse pointer at the line "Object variables". The object variables list and their current values is open. At the time $X=-10$ and $Y=-18$. If the Step button is pressed next time then Y value will be calculated for $X=-9$. By repeating the actions you can check the correct values of X and Y .

Test the Start, Pause and Stop buttons. To repeat experiment press the Stop button at first. The values of all variables will be set by default. Press the Start button to calculate the scheme continuously and the Pause button to stop it.

Let us stop the calculations after 20 steps ($X = 10$) automatically. It is really difficult to stop the system after the exact step number in the Run mode. This is caused by high calculating speed of simple models. You can use Stop the command to stop system in Run time. The syntax is as follows: Stop(Flag). The system will pause if Flag value is more than zero. Add the following line in the Linear_func class text:

`Stop(~X>9)`

Then press the Start button. The scheme will do 20 steps and pause. Open "Object properties" dialog box for Linear_func object. On the variables tab you can see the X value that equals 10.

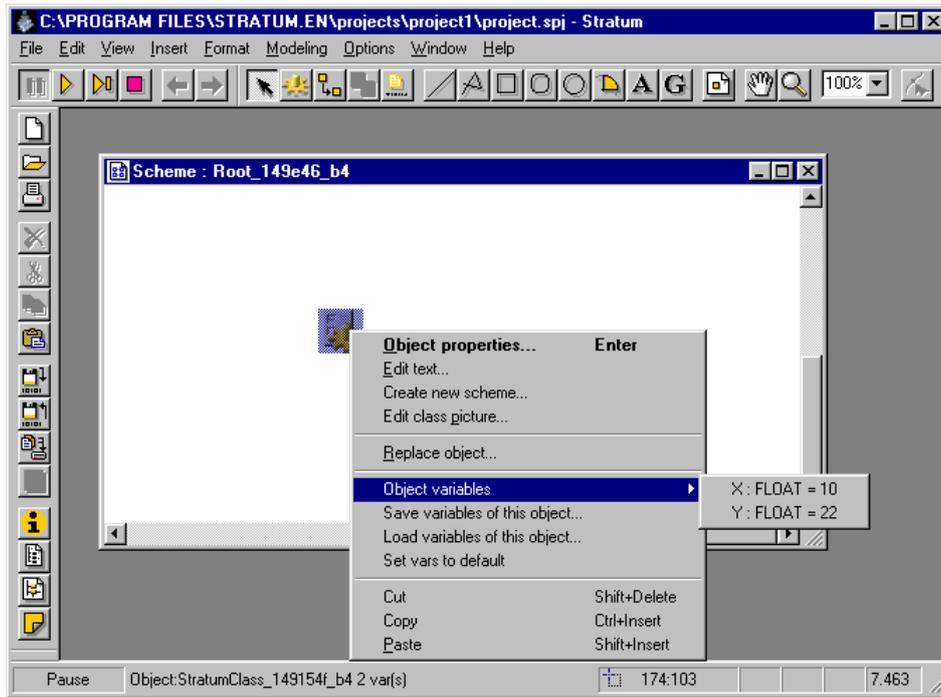


fig. 12 Значение переменных на конец расчета

Проведение связей

In order to link between objects you should press the Create link button  on the Drawing panel. Mouse pointer will look like the lighting. Put the crosshair on the first object and click. Then move the mouse pointer holding the bottom down to the second object and release the button. The red line will link these objects (it means that the link is settled). At the same time the box for describing this link will appear. You need to link the variables of one object with those of the other. Press Add button so many times, as many variables you need to link.

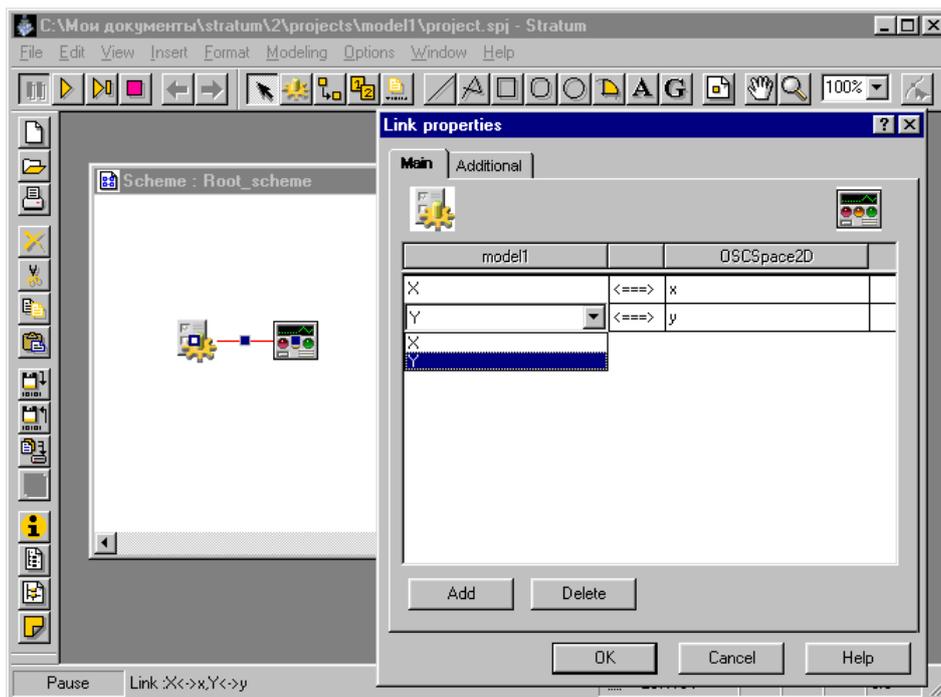


fig. 13 Проведение связи. Диалоговое окно Link properties

For example you should link the variables "X" and "Y" of the left object with the same of the right object. For that click the variable on the left (you will have all variables of the object in the list) and then choose the variable you need (Y). Repeat the same for the right list.

Изменение порядка вычислений

По умолчанию имиджи вычисляются в порядке их установки на схему. Порядок вычисления имиджа – сверху вниз по строкам текста модели. В строке порядок вычислений – слева направо после знака $=$ с учетом скобок, приоритетов операций и вложений. После выполнения строки вычисленное значение присваивается переменной, стоящей слева от знака $=$.

Для изменения порядка расчета имиджей используется кнопка **Order of calculation** button .

Для изменения порядка расчета строк имиджа используются операторы цикла и условные операторы.

Для изменения порядка расчета внутри строки используются скобки.

Имиджи с переменной `_enable`, равной 0, из расчета исключаются.

Имиджи, помеченные как временно удаленные, из расчета исключаются.

Если имидж содержит схему, то после передачи ему управления, оно передается первому имиджу вложенной схемы. После окончания расчета последнего имиджа подсхемы управление возвращается в имидж верхнего уровня.

После расчета последнего имиджа схемы вычисленные данные доступны для просмотра и отображаются в окнах интерфейсных элементов.

После расчета такта вычисленные переменные становятся доступны (передаются по связям) переменным, соединенных с ними имиджей. То есть на следующем такте переменные могут иметь новые значения, получив их по связям.

Во время расчета значения переменных можно наблюдать в окне Watch. Для этого следует пометить необходимые переменные имиджа в закладке Переменные (в левом столбце) и открыть окно Watch пунктом главного меню Вид/Просмотр переменных.

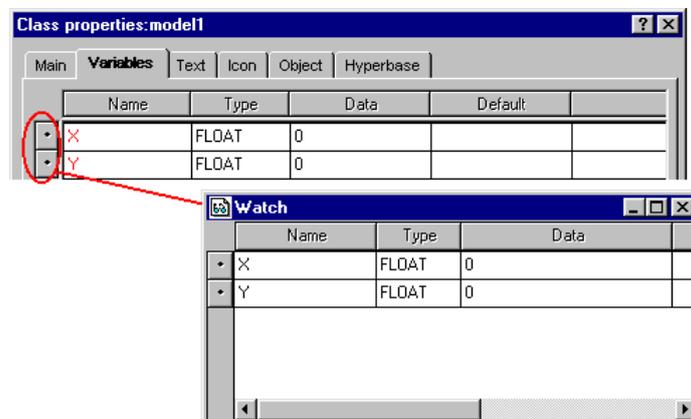


fig. 14 Помеченные переменные и окно Watch

Если в строке текста имиджа встретится оператор `Execute`, то управление будет передано программе (исполняемому модулю), указанной в данном операторе. Возврат

управления будет осуществлен в точку за оператором `Execute` после окончания исполнения программы.

Вы можете контролировать поведение модели, наблюдая за изменением состояния переменных или интерфейсных элементов или управляя элементами ввода. В режиме **Один шаг** вы можете изменить значения любых переменных модели после окончания такта или просто внимательно ознакомиться с ними. Для продолжения расчета следует снова нажать кнопку **Один шаг**.

С помощью механизма сообщений порядок вычислений может быть изменен.

С помощью оператора `Pause` процесс вычислений может быть приостановлен.

Набор значений всех переменных всех имиджей схемы называется **состоянием**. На каждом такте набор переменных имеет новое состояние. Любое состояние может быть сохранено кнопкой **F2** и иметь оригинальное имя. Любое из сохраненных состояний может быть загружено кнопкой **F3**. В этом случае процесс расчета можно повторить заново кнопкой **Запуск**.

Внимание!

Во время работы схемы состояние ее переменных наблюдается в окне `MainWindow` средствами стандартных имиджей из библиотеки `Controls` типа `NumberView` . Для появления этого окна следует на схему установить в обязательном порядке имидж `LGSpace`  из стандартной библиотеки `Unit`.

Установка параметров схемы (переменных имиджей)

Значение любой переменной модели можно изменить как до расчета, так и во время расчета, приостановив его кнопкой **Пауза**. Для доступа к переменным щелкните правой кнопкой мышки на объекте и откройте диалог Свойства объекта, закладка `Переменные`. Здесь представлены все переменные некоторого имиджа, их типы и значение. Для изменения переменной введите новое значение в поле `Data`.

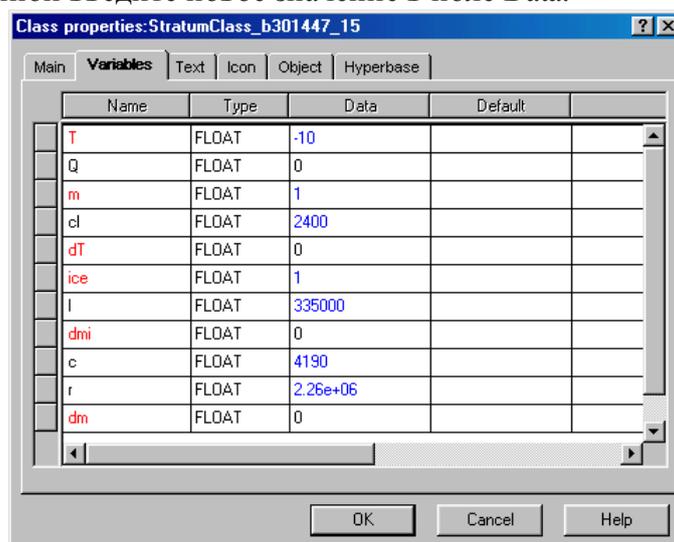


fig. 15 Установка значений переменных (параметров)

Красным цветом выделены вычисляемые моделью переменные, чьи значения зависят от вычислений и других переменных. Черным цветом – независимые, значения которых следует задать. При отсутствии задания значения, им будет присвоен ноль.

Установка параметров по умолчанию

Это значения, которые автоматически присваиваются всем переменным класса в момент загрузки проекта или при нажатии на кнопку Стоп. Переменные хранятся вместе с текстом модели. Для их установки достаточно ввести значения в режиме остановки схемы в диалоге Свойства объекта – Переменные в столбце Default.

Установка начальных значений параметров

Начальные значения параметров хранятся в отдельном файле переменных, загружаются и устанавливаются перед запуском проекта. Они выделяются синим цветом и запоминаются автоматически.

Начальные значения параметров переопределяют значения параметров по умолчанию.

Установка значений всех параметров проекта с одинаковым именем

Для установки значений всех параметров проекта, имеющих одинаковое имя, необходимо в меню Modeling выбрать пункт Set vars value. В диалоговом окне Setting variable выберите из списка переменных имя нужной переменной и укажите ее значение. Чтобы значение переменной автоматически устанавливалось при загрузке проекта, выберите пункт As setted.

Запись/чтение состояний модели

В любой момент времени Вы можете сохранить значения всех переменных схемы в файл командой Modeling | Save State. Затем в любой момент времени вы можете восстановить эти значения командой Modeling | Load State. Можно также сохранить и восстановить значения переменных отдельных объектов, если щелкнуть по нему правой кнопкой мышки и выбрать команды Save variables of this object и Load variables of this object. Значения переменных состояния модели выделяются зеленым цветом.

Модель. Имидж

Теперь наша основная задача - создать текст модели, которую следует рассчитать.

Простейшей задачей является модель из операторов присвоения. Этот тип моделей соответствует численным преобразованиям входных переменных по известным формулам в выходные.

Инженер, проектируя устройство, чертит на бумаге цепочку каскадов, показывающую, как изменяется сигнал, проходя через блоки. Блоки преобразуют сигнал. Экономист, в виде блоков и стрелок показывает последовательность расчетов (формул) своей методики. Блоки преобразуют данные. Чтобы схема проимитировала эту деятельность, разбейте свою задачу на такие блоки.

ВОЗЬМИТЕ ЛИСТ ДЛЯ ПРОЕКТИРОВАНИЯ,	Кнопка «Быстрое создание нового проекта»	
Заведите на каждый блок имидж	Кнопка «Создать и вставить новый имидж»	
и впишите в него формулы.	Откройте имидж (двойной щелчок мышкой по имиджу), закладка «Текст», впишите формулы	

Анализатор автоматически разберет формулы и выделит из них переменные.	Нажмите кнопки «Компилировать», «ОК». Посмотрите закладку «Переменные».	
Соедините связью имиджи в требуемом порядке – выходную переменную одной формулы с входной переменной последующей формулы.	Кнопка «Проведение связи». Укажите мышью на выходной имидж, удерживая левую клавишу мыши, доведите указатель до входного имиджа. В открывшемся окне нажмите кнопку «Добавить» (по количеству соединяемых пар переменных), установите между соединяемыми переменными из списка входных и выходных переменных, нажмите «ОК».	
Задайте значение входных переменных.	Откройте имидж (двойной щелчок мышкой по имиджу), закладка «Переменные», укажите строку нужной переменной в столбце «Данные», введите число. Другие начальные числа аналогично. Нажмите «ОК».	
Запустите проект на счет.	Нажмите кнопку «Запуск»	
Остановите счет.	Нажмите кнопку «Пауза»	
Посмотрите ответ.	Откройте нужный имидж (двойной щелчок мышкой по имиджу), закладка «Переменные», смотрите строку нужной переменной в столбце «Данные». Нажмите «ОК».	

Архитектура счета

Расчет схемы осуществляется циклически повторяясь, если нажата кнопка Пуск или одноразово, при нажатии кнопки Один шаг. Один проход схемы (всех имиджей) интерпретатором называется шагом (тактом). Реальное время, за которое был выполнен такт (или количество тактов в секунду), указывается в правом нижнем углу статусной строки. Доступ переменных одного имиджа к значению переменных другого имиджа осуществляется (в общем случае) один раз в конце такта в виде передачи по связям.

Порядок вычислений в имидже - по строкам текста сверху вниз с учетом операторных скобок (if - endif). Строки, помеченные знаками // и /* */, в расчете не учитываются, так как воспринимаются как комментарий. В строке порядок вычислений - слева направо с учетом круглых скобок. Функции (операторы), имеющие флаг, вычисляются (выполняются) в общем порядке, но только в том случае, если значение флага больше единицы на момент их выполнения.

Порядок вычисления имиджей соответствует порядку их установки на схему. Нажмите на панели рисования кнопку Порядок вычислений. Около каждого имиджа появится специальный знак (красный квадрат с цифрой), указывающий его номер в схеме. Более ранние номера вычисляются первыми. Следует отметить, что порядок установки имиджей в основном оказывает влияние на связанные переменные, упоминаемые в правых частях со знаком тильда.

Номер имиджа можно изменить, если указать курсором на интересующий имидж и нажать левую клавишу мыши. Указывая на имиджи по порядку, пронумеруйте их снова, начиная с 1.

Если требуется изменить номер отдельного имиджа, нажмите и удерживайте клавишу Shift, укажите курсором на требуемый имидж, нажмите левую клавишу мыши. В появившуюся подсказку введите требуемый номер и нажмите ОК.

Если имидж имеет схему, то в момент его обработки происходит переход на обработку имиджей его схемы, которые обрабатываются согласно собственному порядку. После вычисления последнего из них система возвращается к расчету верхнего уровня, продолжая с прерванного места.

В тексте модели могут встретиться уравнения. Различают локальные и глобальные системы уравнений. Локальные уравнения помечаются как $\langle \text{неизвестное} \rangle, \dots = ?$, содержатся в тексте одного имиджа и разрешаются по мере их встречи, как только количество прочитанных системой уравнений и неизвестных совпадет. Корни становятся доступными немедленно. Глобальные уравнения разрешаются после выполнения всех присваиваний, и помечаются как $? = \langle \text{неизвестное} \rangle, \dots$.

В тексте модели могут встретиться дифференциальные уравнения. Их расчет производится после расчета присваиваний, но перед расчетом уравнений.

Последними на такте выполняются присвоения, помеченные как $: :=$.

После вычисления левых частей всех выражений во всех имиджах на текущем такте их значения становятся доступными по связям переменным других имиджей или переменным в правых частях выражений, которые будут использованы для вычислений на следующем такте.

В силу сказанного следует иметь в виду, что значения входных и выходных переменных в имидже, использующем присвоения, могут запаздывать на один шаг. Новое значение является следствием и будет доступно другим переменным только на следующем шаге.

НАПРИМЕР,

```
q:=s+5
b:=q+3
```

Зададим начальные условия данных 4 для s , 0 для q , 0 для b .

На начало второго такта значение переменной q будет равно 9. На начало второго шага значение b будет определено с учетом того, что q было равно 0 на начало шага, так как новое значение q для b пока не доступно. То есть b будет равно 3.

На начало третьего шага значение q равно 9, а значение b будет равно 12. В динамике этот процесс задержки и проталкивания результата можно представить в следующем виде (таблица 1).

Переменная	Значения на начало такта 1	Значения на начало такта 2	Значения на начало такта 3
s	4	4	4
q	0	9	9
b	0	0	12

Таблица 1 Значение переменных s, q, b на начало 1, 2 и 3 тактов

Таким образом, с учетом двух задержек между s и b, правильный результат для b будет вычислен за два такта.

Рассмотрим этот же пример, используя приставку тильда (~).

$$q := \sim s + 5$$

$$b := \sim q + 3$$

Зададим начальные условия данных 4 для s, 0 для q, 0 для b.

Теперь вычисленное значение q будет сразу доступно на этом же шаге правой части второй строки, как в прочем и другим имиджам по связи. Результаты представлены в таблице 2.

Переменная	Значения на начало такта 1	Значения на начало такта 2	Значения на начало такта 3
s	4	4	4
q	0	9	9
b	0	12	12

Таблица 2 Значение переменных s, q, b на начало 1, 2 и 3 тактов

Как видно, в данном случае задержки не происходит. Разумеется, если переставить строки местами

$$b := \sim q + 3$$

$$q := \sim s + 5$$

то тильда не сумеет компенсировать задержку и эффект будет аналогичен первому примеру.

Во многих случаях задержку использовать целесообразно для имитации причинно-следственных отношений, но если не учитывать эффект запаздывания, могут получаться неверные результаты. Например, если делить на переменную, чье значение еще не вычислено на текущем такте, то произойдет деление на ноль.

Рекомендуем предусматривать такие ситуации, во-первых, путем установления безопасных чисел в начальные значения промежуточных переменных. Во-вторых, можно установить в меню Параметры пункт Параметры среды закладка Вычисления при обработке ошибки параметр "Не замечать". В-третьих, возможно в формулах, где ожидаются подобные ошибки, установить нормирование, например делить не на X, а на $(X+1*\delta(X))$. Функция $\delta(X)$ вырабатывает значение 1, когда X равно 0, таким образом, все выражение никогда не может быть равным 0. В-четвертых, можно установить формулу внутри оператора условия if. Пока значение не благоприятное, вычислять по одной формуле или не вычислять вовсе, а задавать значение. Когда значение будет соответствовать требуемому, то исчислять по другой формуле. Переключение формул установить по условию, которое записывается как условие оператора If. В любом случае рекомендуется дождаться установившегося результата, когда причины дойдут до самих отдаленных последствий в схеме и только после этого использовать результаты расчета.

Для достижения установившегося результата кнопкой Один шаг "протолкать" схему на требуемое число шагов. Рекомендуем запомнить полученное состояние кнопкой Записать состояние в default.stt и в дальнейшем использовать именно его как начальное (кнопка Загрузить состояние из default.stt).

см. Топик системы помощи Calculate order.

Остановка модели по условию

Для того, чтобы остановить схему при достижении по условию, добавьте в модель функцию Stop(условие). Например, `Stop(t > 10)`. Когда условие станет истиной, расчет проекта будет автоматически остановлен.

Управление работой отдельных имиджей модели

Для управления работой имиджей используются специальные переменные:

`_disable` – если установлена в 1, математическая модель объекта и всех вложенных в него подсхем не обрабатывается. Если в 0, объект работает как обычно.

`_enable` – если установлена в 0, математическая модель объекта и всех вложенных в него подсхем не обрабатывается. Если в 1, объект работает как обычно.

Для управления этими переменными достаточно упомянуть ее имя в контексте модели. Переменная будет доступна в списке переменных.

Переменные имиджей модели

Переменная это последовательность русских или (и) латинских букв и цифр, начинающаяся с буквы длиной не более 255 символов

Заглавные и маленькие буквы неразличимы.

Среда автоматически выделяет переменные из контекста модели. Выделенный из контекста список переменных может быть просмотрен в закладке Переменные имиджа. Достаточно упомянуть имя переменной в тексте модели, чтобы она была доступна. Переменной будет присвоен тип FLOAT по умолчанию. Переменной FLOAT будет присвоено значение 0 по умолчанию.

Если требуется изменить тип переменной, то следует указать это явно.

Например, `Colorref P`.

Внимание!

Следует проверять правильность выделения переменных по списку переменных (закладка Данные). В случае опечатки в тексте модели выделение переменных может произойти не верно.

Ранее объявленные переменные (имеющиеся в списке переменных) в тексте модели выделяются цветом.

Переменная является уникальной в пределах одного объекта, текста модели.

Вложенные объекты обладают разными переменными, даже если они названы одинаково.

Если переменные двух объектов связаны, то они указывают на одну область памяти и поэтому их значения идентичны.

Типы переменных

В среде можно использовать базовые типы переменных: FLOAT, COLORREF, HANDLE, STRING.

Остальные типы могут быть спроектированы как составные из базовых.

FLOAT

Тип `FLOAT` присваивается любой переменной по умолчанию. `FLOAT` используется для хранения чисел с плавающей запятой в пределах от $\pm 1.7^{10-308}$ до $\pm 1.7^{10+308}$.

Например, `FLOAT F`
`F := -50.203`

COLORREF

`COLORREF` используется для кодирования цвета в пределах от (0,0,0) до (255,255,255)

Например, `COLORREF C`
`C := RGB(100,200,40)`

HANDLE

`HANDLE` используется для хранения указателей на объекты в пределах от -2147483648 до +2147483647.

Например, `HANDLE H`
`H := #3125`

STRING

`STRING` используется для хранения символьных переменных.

Например, `STRING S`
`S := "ASD"`

LOCAL

Можно ограничить область видимости любой переменной имиджа. Для этого необходимо при определении типа переменной указать дополнительный параметр `LOCAL`. Переменная или группа переменных, определенных таким образом, ограничены одним имиджем и недоступны для связи с другими имиджами.

Например, `FLOAT local A, B // Локальные переменные`
`FLOAT C, D`
`A := 1`
`B := 2`
`C := ~A + ~B`
`D := ~C * ~A`

Предопределенные переменные

Предопределенные переменные это переменные, содержащие свойства объекта. Имя предопределенной переменной зарезервировано. Устанавливаются автоматически только при нажатии на кнопку `Стоп`:

`FLOAT orgx` – значение координаты X изображения объекта на схеме;
`FLOAT orgy` – значение координаты Y изображения объекта на схеме;
`HANDLE _hobject` – дескриптор изображения объекта на схеме;
`STRING _classname` – имя класса объекта,

π – значение числа ПИ, равное 3,14...

Для работы с предопределенными переменными необходимо объявить их в имидже или упомянуть в контексте модели.

Например, `FLOAT orgx, orgy`

Стандартные библиотеки имиджей

Включают базовый набор, наиболее часто применяемых имиджей.

Библиотека Unit

Библиотека содержит имиджи, создающие различные окна для отображения и редактирования графики, текста, таблиц и других данных, а также другие устройства, полезные при работе:



GraphicSpace

Используется для отображения различных графических сцен, созданных непосредственно из модели или считанных из VDR файла.

Имидж создает окно, содержащее графическое пространство, и управляет его параметрами. Графическое пространство создается одновременно с созданием окна, или, если установлена переменная `FileName`, считывается из заранее подготовленного VDR файла.

Необходимым параметром для создания является имя окна `WindowsName`. Если окно с таким именем уже существует, имидж получит его параметры и может им управлять.

Созданное окно располагается поверх остальных окон и имеет размеры, устанавливаемые по умолчанию, если не заданы переменные `Width` и `Height`. В случае случайного закрытия, окно автоматически восстанавливается.



GraphicSpaceEx

Используется для создания окна в окне. В этом случае окно, созданное в другом, (дочернее) имеет относительные координаты от родительского и не может выйти за его пределы. Такие окна могут применяться при необходимости одновременного доступа ко многим пространствам. Используя `GraphicSpaceEx` их можно удобно разместить в одном окне.

Имидж создает дополнительное графическое пространство к уже имеющемуся в некотором окне. Необходимым параметром является имя существующего окна - `MainWindow` и собственного окна - `WindowsName`. В переменную `FileName` можно поместить имя VDR файла или имя класса со схемой для считывания в пространство при его создании.



LGSspace

Используется для отображения различных графических сцен, расположенных непосредственно на схемах модели. Ими могут быть различные панели с элементами управления, диалоги с пользователем, графические меню и другие графические элементы.

Имидж создает окно с графическим пространством, содержащим копию изображения схемы некоторого класса. Отображаемую схему можно задать с помощью переменной `ClassName`, если она не указана, используется схема, в которой расположен имидж `LGSpace`.

Необходимым параметром для создания является имя окна `WindowsName`. По умолчанию устанавливается имя окна `MainWindow`. Если указан какой-либо графический объект (переменные `hObject` – дескриптор или `ObjectName` – имя объекта), то размеры видимой области окна автоматически устанавливаются по его размерам. Иначе размеры устанавливаются по умолчанию или из переменных `Width` и `Height`.



OSCSpace2D

Используется для динамического отображения данных, результатов вычислений в виде графика, осциллограммы. Для отображения в одном окне несколько "лучей" необходимо установить в соответствующих осциллографах одинаковые имена окон (и, если нужно, масштабы).

Имидж создает окно с графическим пространством, содержащим линейные оси и сетку осциллографа. Далее, он отображает траекторию изменения точки с координатами X, Y .



Gistogramma

Используется для динамического отображения данных, результатов вычислений в виде двумерной гистограммы, т.е. изображения интервального вариационного ряда в виде прямоугольников разной высоты.

Имидж создает окно с графическим пространством, содержащим оси гистограммы, и осуществляет строительство в них гистограммы согласно получаемым данным.



MessageBox

Предназначается для вывода небольших информационных сообщений или запросов пользователю в отдельном всплывающем окне. Например – "Вы хотите продолжить?". Размеры окна задаются автоматически в зависимости от содержания. При установке соответствующих флагов возможна установка стандартных иконок и нескольких кнопок для простейших операций выбора.

По умолчанию имидж отключен. Диалоговое окно создается при установке 1 в переменную `_enable`, после чего переменная опять автоматически устанавливается в 0.

Текст сообщения задается с помощью переменной `Info`. Необходимая иконка и кнопки задаются флагами (`flags`). Определить какая кнопка была нажата пользователем можно из переменной `Buttons`.

Примечание: Переменная `flags` имеет тип `FLOAT`. Флаги представляют из себя константы и перечисляются с помощью операции `|` (логическое или).



InputBox

Диалог предназначен для вывода небольших информационных сообщений и ввода текстовой строки (до 255 символов). Например – "Введите имя файла:". Размеры окна задаются автоматически. Возможна установка текста по умолчанию.

По умолчанию имидж отключен. Диалоговое окно создается при установке 1 в переменную `_enable`, после чего переменная автоматически опять устанавливается в 0.

Введенную строку можно получить из переменной `InputString`. При нажатии пользователем на кнопку CANCEL возвращается пустая строка.



FileDialog

Представляет собой стандартный Windows диалог, предназначенный для выбора имени файла при операциях открытия/сохранения. Размеры задаются автоматически. Возможна установка фильтров для расширений файлов.

По умолчанию имидж отключен. Диалоговое окно создается при установке 1 в переменную `_enable`, после чего переменная автоматически опять устанавливается в 0.

В случае успешного выбора, имя файла и путь можно получить из переменной `FileName`, иначе переменная содержит пустую строку. Для инициализации окна в переменную `_enable` необходимо установить 1.



ColorDialog

Представляет собой стандартный Windows диалог, предназначенный для выбора цвета. Размеры задаются автоматически. Возможно задание цвета по умолчанию.

По умолчанию имидж отключен. Диалоговое окно создается при установке 1 в переменную `_enable`, после чего переменная автоматически опять устанавливается в 0.

В случае успешного выбора, значение цвета можно получить из переменной `Color`, иначе переменная содержит нулевое значение.



MatrixView

Предназначается для просмотра и редактирования числовых матриц.

По умолчанию имидж отключен. Диалоговое окно создается при установке 1 в переменную `_enable`, после чего переменная автоматически опять устанавливается в 0.



TableEdit

Предназначается для просмотра и редактирования таблицы базы данных.

По умолчанию имидж отключен. Диалоговое окно создается при установке 1 в переменную `_enable`, после чего переменная автоматически опять устанавливается в 0.

Необходимо установить в переменную `FileName` имя файла с таблицей базы данных. После закрытия окна база будет закрыта. Возможна передача дескриптора уже открытой таблицы данных.

Задание шаблона на отображение полей таблицы

Если переменная `Template` содержит пустую строку, то отображаются все поля таблицы.

Иначе эта строка должна быть написана в следующем формате:

```
имя-поля1[n][h= "Заголовок"][r], имя-поля2 ..
```

где:

/n – номер поля,

/h = "Заголовок" – заголовок столбца,

/r – Установка режима ТОЛЬКО ЧТЕНИЕ для данного поля.

Поля разделяются запятыми. Все поля, кроме имени, являются необязательными.



VKeyState

Предназначается для контроля за состоянием отдельной клавиши на клавиатуре или мышке и использования ее для управления моделью.

В переменную `KeyCode` устанавливается код клавиши (см. приложение 2). Если клавиша нажата, `Pressed` устанавливается в 1, иначе 0.

Поскольку клавиатура (как и мышь) является в Windows разделяемым ресурсом, необходимо указывать окно, при активности которого будет контролироваться клавиатура.



MouseState

Предназначается для контроля за позицией и состоянием клавиш на мышке.

Имидж получает все сообщения от мыши, относящиеся к указанному графическому пространству. В переменные `xPos` и `yPos` системой устанавливаются координаты мыши. В переменную `fwKeys` – код нажатой кнопки (см. приложение 1).



ChangeModel

Используется для динамического изменения математической модели имиджа.

Имидж позволяет заменить конкретную строку указанного имиджа, или добавить в него новую.



FileString

Используется для чтения из текстового файла строки с указанным номером

Читает из указанного текстового файла строку с заданным номером.



Table

Используется для работы с таблицей базы данных. Открывает таблицу с указанным именем.

Применяется совместно с имиджами `BaseOpen`, `TableEdit` и другими.



BaseOpen

Открывает базу данных. Возвращает дескриптор открытой базы данных, который может использоваться в имиджах `Table` и других.



DrawFile

Отображает в графическом пространстве окна, указанное количество строк из текстового файла.

Библиотека Controls

Содержит имиджи, позволяющие осуществлять визуальный ввод/вывод и динамическое изменение переменных, отображать диапазон их изменения и другие аналогичные операции. Для всех из них необходимым является задание графического пространства через имя окна (`WindowName`) или дескриптор (`HSpace`). Если изображения управляющих элементов создаются в новом пространстве, необходимо указать их координаты (x и y).



WinButton

Имидж осуществляет управление кнопкой в стиле Windows, то есть содержащей надпись.

Текст кнопки задается переменной `text`.

Особенностью кнопки является возможность запретить нажатие (переменная `Enable`), при этом изображение кнопки соответственно будет закрашиваться серым цветом. При отпускании кнопки в переменную `pressed` устанавливается 1 имидж, обрабатывающий нажатие кнопки, должен установить переменную обратно в 0.



Button

Имидж осуществляет управление кнопкой с графическим изображением.

Изображение кнопки считывается из заранее подготовленного BMP файла (переменная `FileName`). Размер кнопки может быть любой, изображаются отжатое и нажатое состояния кнопки, одно над другим.

Особенностью кнопки является возможность работы в режиме переключателя (переменная `Type`). При нажатом состоянии кнопки в переменную `pressed` устанавливается 1, при отпускании в 0. Переменные `ButtonUp` и `ButtonDown` устанавливаются в 1 при соответственно отжатии и нажатии кнопки.



Lamp

Имидж служит цветовым индикатором.

Лампочка "загорается" при положительных значениях переменной `Light`. Переменная `Color` задает цвет, который по умолчанию красный.

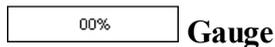


Slider

Имидж осуществляет быстрое изменение численной переменной в заданном диапазоне.

Текущее значение переменной устанавливается в переменную `Value`.

Шаг изменения устанавливается автоматически. Переменные `min` и `max` задают минимальную и максимальную границы изменения.



Имидж обеспечивает графическую индикацию текущего состояния численной переменной в заданном диапазоне в виде прямоугольника.

Текущее значение переменной устанавливается в переменную `Value`.

Шаг изменения устанавливается автоматически. Переменные `min` и `max` задают минимальную и максимальную границы изменения. Переменная `Color` устанавливает цвет заполнения.



Имидж обеспечивает ввод числовых значений и индикацию текущего значения в виде числа на фоне прямоугольного поля.

Текущее значение переменной устанавливается в переменную `Value`.

Значение переменной можно также увеличивать/уменьшать на заданный шаг, используя встроенные кнопки вверх/вниз. Шаг изменения задается переменной `Step` (по умолчанию 1).



Имидж обеспечивает отображение текущего значения числовой переменной.

Текущее значение переменной устанавливается в переменную `Value`. Отображаемое значение переменной округляется, в `size` находится число знаков после запятой.



Имидж обеспечивает редактирование и отображение текущего значения строковой переменной.

Текущее значение переменной устанавливается в переменную `Text`.



Имидж обеспечивает выбор строки из списка, редактирование и отображение текущего значения текстовой переменной.

Список создается из указанного текстового файла по строкам. При выходе за границы файла в качестве строки указывается первая (последняя) строка файла. Текст должен быть в Windows кодировке.



Имидж осуществляет управление кнопкой в стиле Windows, т.е. содержащей надпись.

Текст, располагающийся справа от кнопки, задается переменной `text`.



Имидж осуществляет управление кнопкой-выбором из списка в стиле Windows.

Текст, располагающийся справа от кнопки, задается переменной `text`.

Особенностью кнопки является возможность работать в режиме переключателя, если на схеме установлено несколько таких кнопок.

Для создания несколько групп таких кнопок, необходимо дополнительно вставить на схему форму (Вставка | Формы | Радиокнопка) и объединить нужные кнопки в

графические группы. В имиджах указать соответствующие имена графических объектов.

Библиотека Graph2d

Содержит имиджи, позволяющие манипулировать различными графическими объектами, создавать двухмерные графические сцены, визуализировать данные.

1. Графические объекты

Создание и управление параметрами различных графических объектов. Для всех из них необходимым является задание графического пространства через имя окна (WindowName) или дескриптор (HSpace). Если необходимо управлять графическим объектом, созданным заранее, например из считанного VDR файла, то объект можно задать по имени (ObjectName) или дескриптору (HObject).



Line

Имидж создает в указанном графическом пространстве линию и управляет ее параметрами. Основные параметры линии – координаты ее начала и конца – x_1, y_1 и x_2, y_2 , соответственно.



Arrow

Имидж создает в указанном графическом пространстве изображение стрелки и управляет его параметрами. Стрелка может быть как одно-, так и двунаправленной. Основными параметрами являются координаты начала и конца – x_1, y_1 и x_2, y_2 .



Triangle

Имидж создает в указанном графическом пространстве треугольник и управляет его параметрами. Основные параметры – координаты трех его точек.



Rectangle

Имидж создает в указанном графическом пространстве прямоугольник и управляет его параметрами. Основные параметры – координаты левого верхнего угла прямоугольника (x, y) и размеры (Width, Height).



Polyline

Имидж создает в указанном графическом пространстве линию из произвольного числа точек. Основные параметры – координаты x, y очередной точки. Точки соединяются отрезками и устанавливаются, пока изменяются значения x и y . Установив переменную `regen` в 1, можно обновить параметры линии, то есть, например, отрисовать ее новым цветом или стилем.



Pie

Имидж создает в указанном графическом пространстве сектор эллипса и управляет его параметрами. Основные параметры – координаты x и y , радиусы r_1 (по горизонтали) и r_2 (по вертикали), начальный и конечный углы `angle1` и `angle2` в радианах.



Ellipse

Имидж создает в указанном графическом пространстве эллипс и управляет его параметрами. Основные параметры – координаты x и y , радиусы r_1 (по горизонтали) и r_2 (по вертикали).



Text

Имидж создает в указанном графическом пространстве одну или несколько текстовых строк. Основные параметры – отображаемая строка и координаты текста.



Picture

Имидж создает в указанном графическом пространстве изображение, прочитанное из графического файла. Основные параметры – имя файла и координаты отображения.



Video

Имидж создает в указанном графическом пространстве видефрагмент. Основные параметры – имя AVI файла и координаты отображения. Можно непосредственно управлять порядком воспроизведения кадров, задавая их порядковые номера (переменная `Pos`).



ColorBar

Имидж создает в указанном графическом пространстве прямоугольник закрашенный переходом указанных цветов. Основные параметры – Цвет первый и второй – координаты левого верхнего угла прямоугольника (x, y) и размеры (`Width, Height`).

2. Вспомогательные имиджи

Осуществляют воздействие на графические объекты с целью изменения их параметров. Кроме задания пространства (`WindowName` или `HSpace`) необходимо задать графический объект по имени (`ObjectName`) или указать его дескриптор (`HObject`).



Object2d

Имидж управляет параметрами любых графических объектов и групп. Основные параметры – это координаты, размер и угол поворота и видимость.



ShowObject

Используется для динамического переключения между различными графическими объектами. При этом активный объект становится видимым и может быть автоматически перемещен в указанные координаты.

Имидж управляет видимостью определенного графического объекта, указанного переменной `ObjectName` или `HObject`. Если установлен флаг `Motion`, объект перемещается в координаты x, y , иначе показывается на своем месте.



ScanObject

Имидж позволяет получить дескриптор некоторого объекта в графическом пространстве. Основные параметры это координаты (x, y). Если в указанной точке объектов нет, переменная `HObject` устанавливается в 0. Управлять работой имиджа

можно с помощью переменной `_enable`. В случае если в точке пространства есть несколько объектов, то возвращается дескриптор верхнего (с большим *Z*-порядком).



DragObjects

Имидж позволяет перемещать любые графические объекты в указанном графическом пространстве. Управлять работой имиджа можно с помощью переменной `_enable`.



Color

Имидж позволяет пользователю управлять яркостью и цветом переменной типа `COLORREF`.

Имидж преобразует компоненты *R,G,B* в цвет. Яркость каждой компоненты может изменяться от 0 до 1.

Библиотека *Graph3d*

1. Имиджи перемещения, вращения и масштабирования

Во всех имиджах имеется переменная `hSpace3d`, задающая дескриптор трехмерного пространства. Эту переменную можно устанавливать либо по связи, либо определить ее в имидже уровнем выше.

Три имиджа управляют линейным движением, размерами и вращением трехмерного объекта.

Имиджи во многом схожи и имеют одинаковые переменные. Задание объектов может осуществляться как по именам, так и по их дескрипторам. Причем имя используется только тогда, когда дескриптор равен нулю.



Move3d

Перемещение трехмерного объекта



Resize3d

Масштабирование трехмерного объекта



Rotate3d

Вращение трехмерного объекта



ObjectColor3d

Изменение цвета и других атрибутов трехмерного объекта

2. Трехмерные примитивы

Данные имиджи создают в указанном трехмерном пространстве `hSpace3d` соответствующий трехмерный примитив. Дескриптор созданного примитива записывается в переменную `hObject3d`. После создания трехмерного объекта имидж засыпает.



Bar3D

Имидж создает трехмерный параллелепипед

Sphere3D

Имидж создает трехмерную сферу заданного радиуса, с требуемым количеством сегментов.

Cylinder3D

Имидж создает трехмерный цилиндр с заданным радиусом и высотой.

Cone3D

Имидж создает: конус (если либо R_1 либо R_2 равны 0); усеченный (если радиусы не равны друг другу); цилиндр (если радиусы равны).

Tube3D

Создание трубы с наружным радиусом R_2 и внутренним R_1 и заданной длиной L .

Tore3D

Создание тора с требуемыми радиусами R_1, R_2 и с заданным количеством сегментов.

Pyramid3D

Создание трехмерной пирамиды с требуемой высотой и размером основания.

Custom3D

Имидж создает трехмерный объект в заданном трехмерном пространстве, считывая его из файла с расширением `v3d`.

Teapot3D

Создание трехмерного чайника. Можно включить или выключить наличие: носика, крышки, ручки, самого тела чайника.

Grid3D

Создание плоской сетки с заданным количеством ячеек

3. Трехмерная визуализация

Имиджи трехмерной визуализации предназначены для отображения данных в трехмерном пространстве.

Material3d

Имидж нанесения трехмерной текстуры

OSC3D

Имидж трехмерного осциллографа.

MakeFace3D

Имидж строит трехмерную поверхность.

Histogramm3D

Имидж строит трехмерную гистограмму.

4. Трехмерные функции – имиджи

 _SetObjectBase3d	Перемещение трехмерного объекта
 _ResizeObject3d	Масштабирование трехмерного объекта
 _RotateObject3d	Вращение трехмерного объекта
 _RotateObjectPoints3d	Вращение вершин трехмерного объекта
 _MoveObjectPoints3d	Масштабирование вершин трехмерного объекта

Библиотека Network

Работа с сетью построена на базе динамической библиотеки `scnet32.dll` (`scnet16.dll`). Данная библиотека реализует обмен данными по сети с использованием протокола TCP/IP. В последующих версиях возможна поддержка других протоколов.

Функционально библиотека поддерживает синхронизацию областей памяти на разных машинах по сети.

Stratum использует эту библиотеку для синхронизации переменных имиджей на разных машинах.

Функционально реализация работы в сети осуществляется с помощью функций: `RegisterNetObject` и `UnRegisterNetObject`. Основная, это - `RegisterNetObject`, которая осуществляет связывание переменных на разных машинах.

Для осуществление связи следует указать имидж, переменные которого следует передавать или получать из сети. Имиджи должны содержать одинаковое количество переменных совпадающих типов.

Следует также указать, будет ли этот имидж сервером или клиентом. Опуская технические подробности, можно заметить, что разница между клиентом и сервером следующая.

1. Сервер должен быть запущен раньше клиента. Сервер может сколь угодно долго ждать подключения клиента(ов), клиент же предполагает о активности сервера и не начинает работать без него. Это основное отличие.
2. К серверу может быть подключено множество клиентов, только в этом случае разумно рассылать данные из сервера.
3. При отключении клиента сервер продолжает работать и может обеспечить подключение этого клиента снова. При отключении сервера все клиенты перестают работать.

Кроме того, при рассылке значений переменных можно использовать две стратегии.

1. Посылать их все время, через промежутки времени.
2. Посылать только при изменении.

Для соединения следует указать адрес (буквенный или цифровой), порт и название протокола. Адрес нужен только клиенту, чтобы соединиться с сервером. Порты должны быть одинаковыми. Пока другие протоколы не поддерживаются, но для совместимости следует указывать TCP/IP.

Для работы по сети можно воспользоваться имиджами:



Network_Client_Object

Имидж осуществляет связь переменных любого имиджа с переменными такого же имиджа, но расположенного на другой машине. Это имидж выступает в качестве клиента и должен быть запущен после сервера на другой машине.



Network_Server_Object

Имидж осуществляет связь переменных любого имиджа с переменными такого же имиджа, но расположенного на другой машине. Это имидж выступает в качестве сервера и должен быть запущен до любого клиента.

Эти имиджи осуществляют вызов функции RegisterNetObject, и по сути дела являются еще более высокоуровневым интерфейсом к ней.

Гиперссылки

Stratum поддерживает такую современную технологию, как гиперссылки. Пример гиперссылки можно увидеть на web-страничках. Гиперссылка – это фрагмент текста, картинка или часть картинки, при щелчке кнопкой мыши на которую производятся какие-то действия. В Internet Explorer это, как правило, открытие нового окна или переход на другую страничку.

По аналогии с web-страницами в среде Stratum можно создавать гиперссылки, которые:

1. открывают новые окна,
2. запускают приложения Windows,
3. запускают существующие проекты среды Stratum,
4. выполняют системные команды из заданного набора,
5. ничего не делают (используется при тестировании).

Ниже эти возможности будут рассмотрены подробнее.

Вся совокупность гиперссылок в проекте Stratum образует гипербазу.

Под гипербазой понимается совокупность графических и текстовых страниц, связанных друг с другом посредством логических связей (ссылок).

В качестве страниц могут выступать:

- графические файлы в vdr формате;
- схемы имиджей;
- сгенерированные моделями в процессе работы изображения.

Функционирование гипербазы подразумевает автоматизированный переход между различными страницами. Механизм поддержки гипербазы активен только во время работы модели.

Создание гиперссылки

Создайте новый проект. Как это делается, уже было описано выше.

Вставьте в схему системный имидж “LGSpace” . Благодаря ему при запуске проекта будет открываться новое окно. Теперь на главной схеме проекта создайте имидж с

названием “hyper”. Затем создайте схему этого имиджа. Данная схема в нашем случае и будет открываться при нажатии на гиперссылку. Поскольку схема ничего не содержит, то, для того, чтобы впоследствии наглядно убедиться в том, что открывается именно то окно, которое требовалось, желательно вставить в схему один или несколько графических объектов. Пусть это будет залитый прямоугольник с текстом: “Correct hyperlink”.

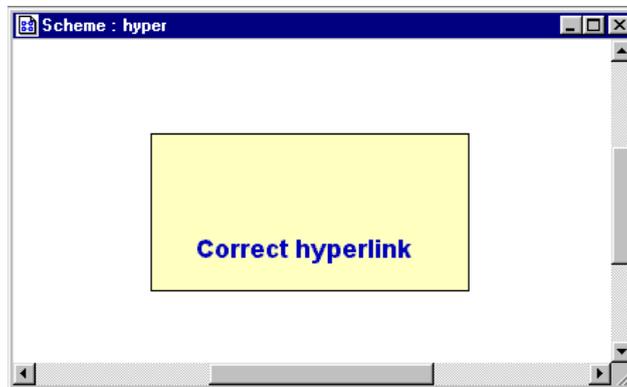


fig. 16 Окно схемы имиджа hyper

Теперь потребуется установить гиперссылку на данную схему. Для этого вернитесь в корневую схему проекта и откройте свойства имиджа “hyper” на закладке «Hyperbase»:

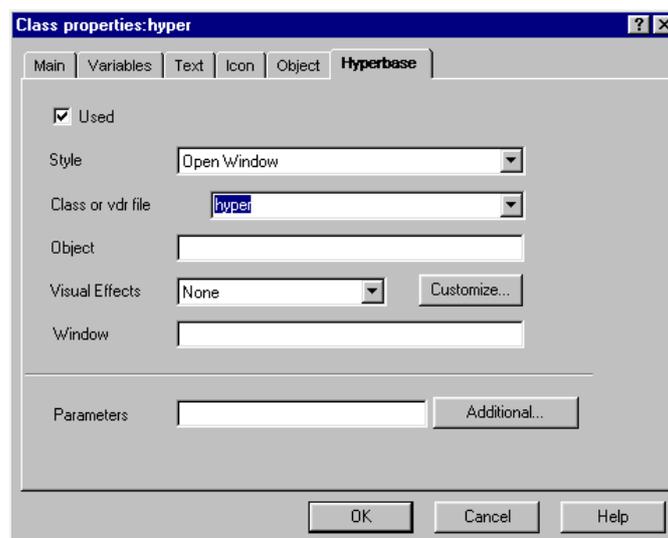


fig. 17 Диалоговое окно свойства имиджа. Hyperbase

Поставьте галочку «Used». Это означает, что данный объект будет иметь гиперссылку. Что именно будет происходить при выборе этой гиперссылки, зависит от настроек, которые будут рассмотрены по отдельности.

Открытие окна

В “Style” выберите пункт «Open Window». Далее потребуется указать в следующем поле имя схемы или файла с расширением “.vdr”. В нашем случае вставляется имя схемы “hyper”. Вместо того, чтобы набирать его с клавиатуры, можно просто выбрать его в выпадающем списке.

В поле «Object» указывается имя объекта, которому посылается сообщение WM_HYPERJUMP в момент открытия окна. Это может использоваться, например,

когда требуется отслеживать, куда перешел по гиперссылкам пользователь. В данном случае его заполнять не нужно.

В поле «Window» указывается имя открываемого окна. Если поле остается пустым, то имя окна будет “MainWindow” для MDI-окон и “PopupWindow” для popup-окон. Впишите в качестве названия «MyWindow».

Поле “Parameters” в данном случае не используется и будет рассмотрено позже.

Теперь можно нажать кнопку «ОК», после чего гиперссылка сохранится.

Запустите проект. При этом откроется новое окно «MainWindow»:

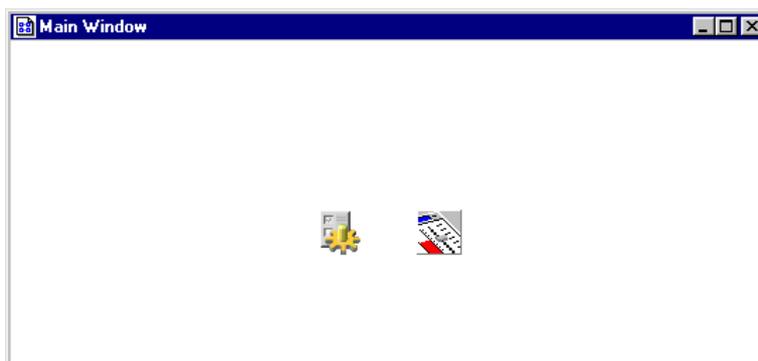


fig. 18 Окно Main window – графическое пространство

В момент нахождения над изображением имиджа «hyper» (изображенного на рисунке в виде шестеренки) курсора мыши он принимает вид руки. Это означает, что у данного объекта имеется гиперссылка. Щелкните левой кнопкой мышки на него, и откроется окно схемы “hyper”. Единственным отличием этого окна от самой схемы, представленной на рисунке 1, будет заголовок “MyWindow” и отсутствие линеек прокрутки (скроллеров).

Настройка параметров страницы

Наличие/отсутствие скроллеров, тип открываемого окна, его начальные размер и привязка задаются в параметрах листа, который служит шаблоном для этого окна. В нашем случае это задается в параметрах листа на схеме “hyper”.

Остановите проект и вернитесь в схему “hyper”. Зайдите в свойства листа этой схемы откройте закладку “Window”:

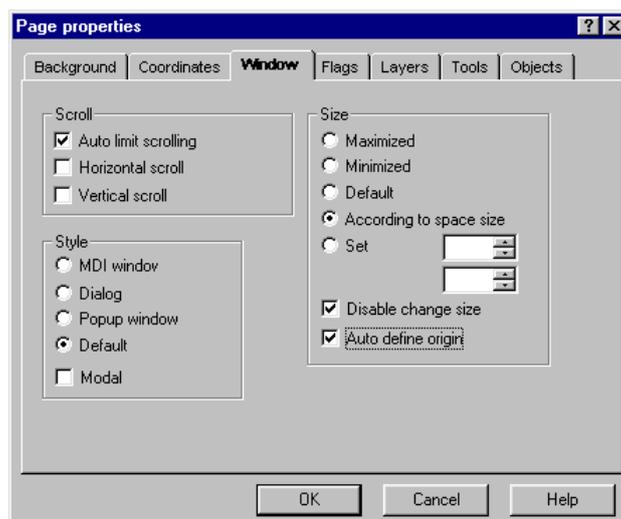


fig. 19 Диалоговое окно Page properties

Группа параметров «Scroll»

Auto limit scrolling позволяет установить прокрутку в зависимости от размера используемого графического пространства.

Horizontal scroll устанавливает горизонтальную линейку прокрутки.

Vertical scroll устанавливает вертикальную линейку прокрутки.

Группа параметров «Style»

MDI window. Выбор данного типа окна означает, что будет открываться MDI – окно (обычное окно с заголовком наверху, изменяемыми размерами, кнопками минимизации и закрытия).

Dialog. Означает, что будет открываться диалоговое окно.

Popup window – всплывающее окно.

Default – открытие окна по умолчанию (сейчас это MDI окно).

Modal – модальное окно (в данный момент не реализовано).

Группа параметров «Size».

Maximized – при открытии максимизировать размер окна.

Minimized – при открытии минимизировать размер окна.

Default – размер и положение окна выбираются операционной системой.

According to space size – размер окна будет таким, чтобы в него точно входили все объекты, имеющиеся в графическом пространстве.

Set – произвольное задание размеров окна с помощью полей ввода.

Disable change size – запрещает изменение размеров окна с помощью мыши или клавиатуры.

Auto define origin – сопоставление левой верхней границы графического пространства соответствующему углу окна. В нашем примере это приведет к тому, что при открытии окна его левый верхний угол в точности совпадет с левым верхним углом прямоугольника.



fig. 20 Пример окна с установленным пунктом Auto define origin

Настройка параметров гиперссылки

Выполнение приложений Windows

Откройте снова свойста имиджа “Hyper” на закладке “Hyperbase”. В поле “Style” выберите пункт меню “Execute Windows application”. После этого название второго поля изменится на “File”. В этом поле необходимо задать имя выполняемого файла приложения с указанием пути к нему.

Остальные поля в данном случае не используются.

Запуск другого проекта

Выберите в поле “Style” пункт “Run project”. Название второго поля поменяется на “Project file name”. В нем необходимо, как и в случае с запуском приложения Windows, указать имя файла вместе с указанием пути. Но вместо выполняемого файла там будет проект Stratum с расширением “*.spj” или “*.prj”

В момент нажатия на объект, имеющего гиперссылку на другой проект, при отсутствии дополнительных параметров в закладке “Hyperbase” происходит следующее.

- 1) Выполнение текущего проекта останавливается.
- 2) Запускается указанный в гиперссылке проект.
- 3) На панели инструментов активизируется левая кнопка гиперперехода “Previous Page “ (стрелка окрашивается в синий цвет).

Последнее, кстати, происходит при любом типе гиперпереходов, а не только при запуске независимого проекта.

Если в момент выполнения второго проекта нажать эту кнопку, то среда вернется к выполнению первого проекта, прервав вычисление второго. В случае же нажатия кнопки “Stop(clear)” возвращения к исходному проекту не произойдет, текущим проектом останется второй. И при повторном нажатии кнопки “Start” он снова начнет выполняться. То есть в данном случае возвращение к исходному проекту возможно лишь с помощью кнопки “Previous Page”. Если она была нажата, то вторая кнопка гиперперехода на панели инструментов активизируется. И теперь, если необходимо вернуться обратно во второй проект, можно вместо гиперссылки нажать на эту кнопку.

Очень часто пользователь при большом количестве гиперпереходов пугается в них и не помнит, какой проект сейчас активен – главный или второстепенный. В этом случае признаком активизации главного проекта может служить неактивная кнопка “Previous Page”. Обратное не всегда верно, так как она, как уже было сказано выше, активизируется при любом гиперпереходе, а не только при запуске нового проекта.

Можно настроить запуск другого проекта по время гиперперехода таким образом, что будут, например, открываться окна редактирования схем, скрыты все окна родительского проекта, при остановке проекта будет осуществлен переход обратно к родительскому. Все это задается в поле “Parameters” закладки “Hyperbase”. Параметры задаются следующим образом:

```
/parameter1[option11[,option21[,...]]] [ /parameter2[option21[,option22[,...]]] [...]]
```

Параметры могут быть следующими.

/open: – позволяет задать, как именно будет запускаться другой проект. Опции этого параметра:

run – запуск подпроекта независимо от текущих установок;

stop – открытие без запуска. Для запуска необходимо будет нажать кнопку “Start”;

edit – открытие окна редактирования схемы. Какой именно схемы – задается параметром /scheme.

/stop: – указывает, что произойдет при остановке проекта. Опции:

edit – открытие окна редактирования главной схемы родительского проекта;

close или **return** – закрытие подпроекта и возврат в предыдущий. При указании этой опции для возвращения в родительский проект уже не придется после нажатия кнопки “Stop(clear)” нажимать “Previous page”, возвращение произойдет автоматически.

/hidewindows – при указании этого параметра в момент гиперперехода все окна родительского проекта будут закрыты.

/readonly – этот параметр позволяет запускать проект в режиме “readonly”. То есть подпроект в этом случае редактировать нельзя.

/save: – устанавливает режим записи изменений в проекте. Опции следующие:

prompt – выводить обычный диалог для записи изменений;

no – не записывать изменения;

/state=filename – позволяет задать имя файла состояний, отличающегося от “_preload.stt”, загружаемого средой по умолчанию. Если путь содержит пробелы, то его необходимо заключить в двойные кавычки.

/scheme=schemename – имя схемы, которая будет активна после гиперперехода. Если этот параметр не указан, то открывается главная схема.

Пример параметров запуска гиперпроекта при гиперпереходе:

```
/OPEN:RUN,EDIT /STOP:CLOSE,EDIT /SCHEME=PLACE /HIDEWINDOWS
```

В данном случае будет происходить следующее:

- при гиперпереходе подпроект будет запущен автоматически, будет открыта его схема “place”;

- при остановке подпроекта все его окна будут закрыты, будет осуществлен переход к родительскому проекту и открыта его главная схема;
- все окна родительского проекта при активизации гиперссылки будут закрыты.

Ничего не делать

Данный режим (пункт “Nothing”) используется лишь в отладочных целях. В этом случае курсор мыши над графическим объектом так же будет принимать вид руки, но при нажатии на него никаких действий производиться не будет.

Системная команда

Выберите пункт “System command” в поле “Style” закладки “Hyperbase” в свойствах имиджа. Название второго поля изменится на “Name of command”. В нем задаются системные команды, которые будут выполняться при выборе данной гиперссылки. Команды могут быть следующими (их можно выбрать в выпадающем списке):

CM_PREVPAGE – переход на предыдущую страницу. В этом случае нажатие на гиперссылку будет эквивалентно нажатию на кнопку “Previous page” в панели управления.

CM_NEXTPAGE – переход на следующую страницу. В этом случае нажатие на гиперссылку будет эквивалентно нажатию на кнопку “Next page” в панели управления.

CM_FILEOPEN – открытие файла.

CM_STOP – приостанавливает выполнение текущего проекта. После нажатия кнопки “Start” на панели инструментов проект продолжит выполнение дальше. Команда эквивалентна нажатию кнопки “Pause”.

CM_CLEARALL – останавливает выполнение текущего проекта. В отличие от предыдущей полностью останавливает вычисление проекта, сбрасывая его состояние в исходное положение. Команда эквивалентна нажатию кнопки “Stop(clear)”.

Использование этих команд позволяет реализовать кнопки управления самостоятельно, что очень часто полезно для дизайна проекта.

Использование функций гиперперехода

В языке Stratum имеется функция, позволяющая симитировать нажатие кнопки мыши на графический объект, являющийся гиперссылкой:

```
StdHyperJump (HANDLE hSpace, FLOAT X, FLOAT Y,
              HANDLE hObject, FLOAT flags)
```

Параметры:

hSpace дескриптор двумерного графического пространства, в котором расположен объект, являющийся гиперссылкой;

X координата указателя мышки относительно объекта, на который имитируется click. Не используется и равно 0;

Y координата указателя мышки относительно объекта, на который имитируется click. Не используется и равно 0;

`HObject` дескриптор двумерного графического объекта, являющегося гиперссылкой;

`flags` дополнительные флаги. Не используется и равно 0.

Пример использования:

```
StdHyperJump (~HSpace, 0, 0, ~HObject, 0)
```

Следующая функция позволяет изменить настройки гиперссылки:

```
FLOAT SetHyperJump2d(HANDLE hSpace, HANDLE hObject, FLOAT mode  
[, STRING target, STRING object, STRING effect,  
STRING window, STRING parameters])
```

Параметры:

`HSpace` дескриптор двумерного графического пространства, в котором расположен объект, являющийся гиперссылкой или который предполагается сделать гиперссылкой;

`HObject` дескриптор двумерного графического объекта, являющегося гиперссылкой или который предполагается сделать гиперссылкой;

`mode` стиль гиперссылки (см. поле “Style” закладки “Hyperbase”), возможны следующие значения:

-1 удалить существующую гиперссылку (эквивалентно ликвидации отметки “Used”),

0 открытие нового окна,

1 запуск приложения Windows,

2 загрузка нового проекта,

3 ничего не делать,

4 выполнение системной команды.

Следующая группа параметров является необязательной:

`target` имя класса или `vdg` файла, имя файла, системная команда (в зависимости от предыдущего параметра);

`object` объект, получающий сообщение `WM_HYPERJUMP` в момент активизации гиперссылки;

`effect` эффект замещения старого изображения новым;

`window` имя открываемого окна;

`parameters` дополнительные параметры, влияющие на открытие проекта.

В примерах мы использовали в качестве гиперссылки имидж. На самом деле в качестве нее может выступать любые объекты, за исключением связей. У всех графических объектов в окне свойств так же имеется закладка “Hyperbase”. Поэтому, в принципе, возможно создание проекта с большой гипербазой и одним единственным имиджем – корневым.

Пример использования функции:

ОТКРЫТИЕ ОКНА

```
ret := SetHyperJump2d(~HSpace, ~HObject, 0, "ClassName")
```

ВЫЗОВ ПОДПРОЕКТА

```
ret := SetHyperJump2d(~HSpace, ~HObject, 2,  
"..\project1\project.spj", "", "", "",  
"/open:run /stop:close /hidewindows")
```

ВЫПОЛНЕНИЕ СИСТЕМНОЙ КОМАНДЫ

```
ret := SetHyperJump2d(~HSpace, ~HObject, 4, "CM_CLERALL")
```