Министерство общего и профессионального образования
Российской Федерации
РОСТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Ю.А. Кирютенко В.А. Савельев

Объектно-ориентированное программирование и язык Smalltalk.

Окна системы Smalltalk/V for Windows

Ростов-на-Дону
2000

Ю.А. КирютенкоВ.А. Савельев

Объектно-ориентированное программирование и язык Smalltalk.

Окна системы Smalltalk/V for Windows

Аннотация

Методическая разработка посвящена современному направлению в программировании — объектно-ориентированной методологии программирования и языку Smalltalk и является продолжением вышедших ранее методических разработок по объектно-ориентированному программированию и языку Smalltalk, посвященных общим концепциям и синтаксису языка, интерфейсу пользователя и среде программирования, классам Collection, Magnitude, Stream и их подклассам, протоколам поддержки всех объектов и всех классов системы, классам ядра графики системы как в Smalltalk for Dos так и в Smalltalk/V for Windows. В данной методической разработке описываются окна, их устройство и применение в системе Smalltalk/V for Windowsфирмы ParkPlace/Digitalk (США).

Методическая разработка предназначена для студентов 3—5 курсов механико-математического факультета и слушателей ФПК.

Печатается в соответствии с решением кафедры математического анализа Ростовского государственного университета, протокол №4 от 14 октября 1997 года.

Настоящие методические указания набраны в системе LaTeX с использованием кириллических шрифтов семейства «Литературная» (AOParaGraph) и математических шрифтов семейств Computer Modern и Math Symbol (American Mathematical Society).

1 Основные операции с окнами в среде Smalltalk/V

Чтобы активно использовать среду Smalltalk/V, надо иметь ответы на следующие вопросы, связанные с окнами системы и интерфейсом пользователя:

- что такое окно, панель окна, меню окна,
- как использовать окна, различные типы панелей и меню,
- как вычислять выражения языка Smalltalk,
- какие стандартные окна для работы предоставляет среда,
- как работать со стандартными окнами.

Эти и некоторые, связанные с ними вопросы, мы и рассмотрим ниже.

1.1 Окна и панели

Под окном в среде Smalltalk/V понимается часть экрана, имеющая заголовок окна (title bar) и окруженое рамкой (border). Как объекты системы окна — это экземпляры класса Window. Область внутри рамки окна является объединением одной или более панелей, экземпляров класса SubPane. Окно с именем Transcript, которое открывается при запуске системы, — пример окна с одной панелью, а окно ClassBrowser имеет четыре панели.

По своему поведению на экране окно может быть активным (выбранным) или неактивным (невыбранным) окном. Если на экране открыто одно или несколько окон, то, чтобы выбрать окно — сделать его активным, поместите курсор внутрь окна и щёлкните (быстро нажмите и отпустите) левой кнопкой мыши. Если курсор поместить вне всех окон и щёлкнуть левой кнопкой мыши, то все окна станут неактивными. Активное окно — всегда в стеке окон первое (на экране — верхнее), а название окна имеет цвет, отличный от цвета названий неактивных окон. Окно остаётся активным независимо от того, находиться или нет курсор внутри него. Чтобы дезактивировать окно, поместите курсор поверх другого открытого окна или свёрнутой иконки окна, и нажмите любую кнопку мыши. Активным становится это окно, а прежнее активное окно становится неактивным.

Активизация окна левой кнопкой мыши только активизирует окно, в то время как щелчок правый кнопкой мыши одновременно активизирует и само окно, и меню панели только что активизированного окна. Нажатие на свёрнутую иконку окна левой кнопкой мыши активизирует окно, а нажатие на свёрнутую иконку окна правой кнопкой мыши вызывает на экран СистемноеМеню.

Когда на экране есть несколько открытых окон, нажимая функциональную клавишу F9, можно дезактивировать активное в настоящее время окно и переходить в следующее по порядку окно, активизируя его. Если продолжать нажимать F9, то активизируются по порядку все окна одно за другим. Такой механизм называется циклом по окнами, и полезен, когда есть много открытых окон, полностью или частично перекрывающие друг друга. Обратите внимание, что цикл применим только к стеку открытых окон среды Smalltalk. Окна других приложений при этом остаются незатронутым.

Итак, окно состоит из одной или более панелей. Наиболее распространённые виды панелей — текстовые, списковые, графические и кнопочные панели. Есть ещё анимационные панели и групповые панели (панели для группы панелей). Кроме того, можно создавать собственные подклассы класса SubPane, добавляя к системе Smalltalk новые в функциональном отношении панели.

Многие стандартные окна системы Smalltalk/V имеют панели, которые позволяют редактировать текст. Они называются текстовыми панелями. Окно Transcript или любое рабочее окно (Workspace) состоит из единственной текстовой панели. Эти и все другие текстовые панели используют один и тот же интерфейс текстового редактора.

Панели, которые позволяют производить выбор из отображаемого ими списка, называют списковыми панелями. Окна, которые имеют списковые панели, обычно называются окнами просмотра (браузерами). Окно просмотра Класса (Class Browser) — пример окна с двумя списковыми панелями и текстовой панелью.

Класс ControlPane (подкласс класса SubPane) содержит все управляющие структуры окна, реализуемые такими классами, как Button (Кнопка), EntryField (ПолеВвода), GraphBox (ПолеГрафики), ListBox (Список), ScrollBar (ПолосаПрокрутки) и StaticPane (НеизменяемаяПанель). Каждый из них имеет многочисленные подклассы, управляющие более тонкими настройками. Пользуясь окном просмотра Иерархии Классов (Class Hierarchy Browser), которое будет описано ниже, эти классы можно изучить более подробно.

Панель для группы (экземпляр класса GroupPane) используется прежде всего для того, чтобы несколько разных панелей, как единое целое, установить внутри окна, то есть создать своего рода «сложную панель» внутри сложного окна. В окне просмотра Иерархии Классов кнопки переключения instance/class, составляющие взаимно исключающее множество из двух «радио-кнопок» — это два экземпляра класса RadioButton, которые сгруппированы в экземпляре класса GroupPane.

Графические и анимационные панели позволяет отображать всевозможную графику. Так как эти типы панелей не используются в стандартных окнах среды Smalltalk/V, здесь они рассматриваться не будут.

1.1.1 Прокручивание

Есть много примеров, когда содержание, которое требуется отобразить в панели окна, требует для своего отображения большее пространства, чем может предоставить соответствующая панель. Можно рассматривать панель как область просмотра, которая позволяет сразу увидеть только часть из всего содержимого панели. Прокручивание позволяет перемещать область просмотра вдоль «виртуального пространства», включающего все содержимое. В текстовой панели можно производить прокручивание вверх, вниз, влево и вправо вдоль страницы, большей чем та область, что помешается в пространство на экране, выделенное для текстовой панели. Точно также и в графической панели, можно проводить прокручивание, чтобы в поле просмотра попадали различные части большого изображения.

Один из способ прокручивания панели состоит в использовании мыши и полосы прокрутки. Полоса прокрутки — прямоугольная область со двумя стрелками прокрутки на каждом из концов и прямоугольным бегунком, располагаемым гденибудь между стрелками прокрутки. Позиция бегунка, называемого ещё «слайдером», на полосе прокрутки указывает позицию отображаемого текста или графического образа относительно всего виртуального пространства информации, предназначенного для отображения в окне. Например, когда бегунок находиться посередине вертикальной полосы прокрутки текстовой панели, то, это означает, что видимый в панели текст расположен примерно по середине всего текста, доступного для отображения в данной панели.

Полосы прокрутки связываются с конкретной областью экрана. Если панель не должна прокручивать своё содержимое, то в такой панели полосы прокрутки могут опускаться. Вертикальная полоса прокрутки располагается по правому краю панели, а горизонтальная — по нижнему краю.

Щелчок левой кнопкой мыши в области полосы прокрутки произведёт перемещение вверх или вниз, влево или вправо вдоль текста или содержимого графического окна, в зависимости от того, где в полосе прокрутки относительно бегунка Вы его произвели. Например, щелчок выше бегунка в вертикальной полосе прокрутки вызовет на экран содержимое текстовой панели, расположенное ближе к началу страницы на один экран.

Нажатие и удерживание левой кнопки мыши на сером поле полосы прокрутки вызовет функцию листания, которое прекратиться только тогда, когда бегунок достигнет курсора мыши на полосе прокрутки. Если это проделать, содержимое, отображаемое в соответствующей панели, изменяется достаточно быстро.

Концевые стрелки на полосе прокрутки позволяют медленно производить перемещение вдоль содержимого панели в направлении стрелки. Если надо увидеть содержимое,например, чуть выше или чуть ниже видимого, надо нажать один раз

на вертикальной стрелке «вверх» или «вниз», соответственно. Каждый щелчок на стрелке прокрутки перемещает видимое изображение на один модуль в выбранном направление. Размер модуля зависит от содержимого панели: одна запись в списке; одна строка в тексте. Нажатие и удерживание левой кнопки мыши в то время, когда курсор находится на стрелке полосы прокрутки, вызывает непрерывное перемещение в соответствующем направлении.

Как альтернатива работе с мышью, можно проводить прокрутку, пользуясь на клавиатуре клавишами PgUp и PgDn, Home и End, а также клавишами управления курсором (клавишами со стрелками). Панель, на которую направляется действие, является панелью, «имеющей фокус», обычно такой панелью является та панель, в которой был сделан последний выбор.

1.1.2 Пользовательские меню

Меню — стандартный способ представления исполняемых операций как для самих окон и панелей системы Smalltalk/V, так и для их содержимого. Прежде всего, в строке меню каждого окна, расположенной между заголовком окна и панелями, представлены имена тех меню, которые доступны окну в целом. Каждое из них имеет список опций (элементов или строк меню) для выбора. Стандартные меню этого уровня — File, Edit, Smalltalk. В строке меню могут появляться специальные меню, например, Classes, Variables, Methods в окне просмотра Иерархии Классов, в зависимости от функциональных возможностей окна.

Чтобы выбрать меню, надо переместите курсор на заголовок меню в строке меню, нажать и задержать левую кнопку мыши. Меню появиться ниже своего заголовка. Если удерживать кнопку мыши и передвигать курсор по строкам меню, можно быстро просмотреть все меню и их элементы, так как они появляются сразу, как только курсор оказывается поверх заголовка. Чтобы выбрать элемент (опцию), достаточно передвинуть курсор на нужную опцию и отпустить кнопку мыши, а чтобы выйти из меню без выбора, надо переместить курсор вне меню в любом направлении, и отпустить кнопку мыши. Если элемент меню был выбран, выпадающее меню исчезнет, а связанное с выбранной строкой действие выполняется.

В качестве альтернативы, можно выбрать меню из строки, нажимая и отпуская (щёлкая) левую кнопку мыши. При этом выбранное меню появляется на экране и «замораживается» в заданной позиции экрана. Чтобы теперь выбрать элемент (опцию) меню, надо разместить курсор над нужным элементом, нажать и отпустить левую кнопку мыши, а чтобы закрыть меню без выполнения каких-либо операций, надо переместить курсор вне меню в любом направлении, и снова щёлкнуть левой кнопкой мыши.

Во всех окнах возможна работа с меню только с помощью клавиатуры. Чтобы активизировать меню с именем из строки меню окна, надо нажать клавишу Alt и,

удерживая её, нажать алфавитную клавишу, обозначающую подчёркнутую литеру в заголовке имени меню. Выбранное меню появиться на экране. Например, нажимая Alt + E, получим меню Edit. Кроме того, если одно из меню вызвано, и клавиша Alt не отпускалась, то используя клавиши управления курсором «влево» и «вправо», можно быстро просмотреть каждое из выпадающих меню. Используя клавиши управления курсором «вниз» и «вверх», в ранее «замороженном» меню можно выделить нужную опцию и выполнить связанное с ней действие, нажимая клавишу Enter (Ввод).

Некоторые пункты меню используются очень часто. Предусмотрено и использование клавиш быстрого вызова (горячих клавиш — Keyboard Accelerator Commands), указанных справа от элементов опускающихся меню, так что пользуясь клавиатурой, всегда можно быстро выбирать нужный пункты меню. Эти команды могут включать нажатие клавиши Alt или Ctrl в комбинации с другой клавишей, соответствующей пункту меню. Например, $Alt + S$ выдаст команду Save из меню File. Некоторые пункты меню не имеют связанной с ними клавиши быстрого вызова.

Кроме того, опции выпадающих меню, наиболее часто используемые в панели того или иного вида, объединены в меню панели и могут быть вызваны, если щёлкнуть правой кнопкой мыши, когда курсор расположен над данной панелью. Меню появиться на экране. Щёлкая левой кнопкой мыши над строкой, можно выбрать её, инициализируя соответствующие операции. Если же щёлкнуть левой кнопкой мыши вне выпавшего меню панели, меню исчезнет.

Во всех меню выбрать можно только те элементы, выбор которых возможен в данный момент времени. Элементы меню, недоступные для выбора, имеют светло-серое (блеклое) написание.

1.1.3 Управление окнами

Окна обычно открываются из меню. Но можно открыть окно и посредством вычисления Smalltalk-выражения, которое произведёт определение, создание и активацию окна.

Как пример откроем новое рабочее окно (окно Workspace). Для этого в системном окне Transcript откроем меню с именем File и выберем в нем строку Workspace. Откроется и станет активным новое рабочее окно. Это окно может быть закрыто, перемещено, оно может прокручиваться, минимизироваться, максимизироваться или изменяться, используя стандартные инструменты управления окнами среды Windows. Рабочее окно, как подсказывает само его имя, является "сверхоперативной" площадкой для работы с приложениями. Будучи открытым, рабочее окно сохранится в среде Smalltalk/V, если образ системы был сохранён в конце сеанса работы. Рабочие окна можно отдельно сохранить на диске для даль-

нейшего редактирования вне среды Smalltalk/V, используя привычный текстовый процессор или другой текстовый редактор.

Управление окном реализуется через инструменты управления, доступ к которым расположен в заголовке окна, где кроме имени окна, расположены иконка СистемногоМеню (слева от имени), минимизирующая и максимизирующая иконки (справа от имени). Кроме того, можно изменять размеры окна, перемещая любой угол или сторону рамки окна.

Чтобы выбирать одну из иконок управления окном, надо разместить курсор на нужной иконке и щёлкнуть левой кнопкой мыши.

Когда выбирается иконка СистемногоМеню, на экране возникает ещё одно меню, которое называется системным меню (или меню системы), с пунктами, которые позволяют воздействовать на состояние окна в целом (типа размера окна и его места расположения). Двойной щелчок на иконке СистемногоМеню выполняет опцию Close — закрывает окно. Если окно связано с файлом на диске или содержит текстовую панель, содержимое которой было изменено, перед тем как закрыть окно, система спросит Вас относительно того, сохранять или нет текущие изменения. Кроме того, к системному меню можно обратиться нажимая на клавиатуре клавиши $Alt+SpaceBar$. Системное меню выполняет следующие операции.

Restore (Восстановить) — восстанавливает окно в исходное состояние, если окно было сжато до иконки или сделано полноэкранным, то есть выбор этого элемента меню изменяет размеры активного окна и его позицию на экране до тех, которые окно имело прежде, чем оно было минимизировано или максимизировано.

Move (Переместить) — "захватывает" активное окно и перемещает его относительно экрана, если используются клавиши управления курсором на клавиатуре (до нажатия клавиши Enter) или мышь, перемещаемая с нажатой левой кнопкой (до момента отпускания кнопки).

Size (Размер) — позволяет изменять размеры активного окна, используя клавиши управления курсором на клавиатуре (до нажатия клавиши Enter), а не возможности перемещения рамки окна с помощью мыши.

Minimize (Минимизировать) — позволяет свернуть активное окно до иконки, которая доступна для быстрого поиска и восстановления окна.

Maximize (Максимизировать) — позволяет расширить рамки активного окна так, чтобы окно заполнило весь экран, предоставляя для редактирования максимально возможную область экрана.

ZoomText — расширяет активную текстовую панель окна так, чтобы она полностью заполнила окно, в котором, возможно, есть несколько панелей (доступно и через *Alt+Z*); если панель уже была изменена до такого состояния, вызывая эту команду вторично, возвращает панель в первоначальное состояние.

Fonts... (Шрифты...) — инициализирует диалоговое окно, позволяя выбрать шрифт и атрибуты шрифта для выбранной текстовой панели или окна.

Close (Закрыть) — закрывает активное окно (доступно и через Alt + F4). Обратите внимание, в окне Transcript этот элемент меню заменяется элементом Exit Smalltalk/V.... См. следующий пункт.)

Exit Smalltalk/V... (Выйти из Smalltalk/V...) — используется только в Системном меню окна для того, чтобы выйти из среды Smalltalk/V; выбор этого элемента заканчивает текущий сеанс работы системы Smalltalk/V, при этом Вас спросят о том, желаете ли Вы или нет сохранить текущее состояние системы (образ системы) прежде, чем сеанс завершится. (Доступно и через Alt + F4).

Последствия выбора Минимизирующей/Максимизирующей иконки аналогичны действиям системы при выборе соответствующих опций системного меню. Чтобы восстановить окно из свёрнутого состояния, щёлкните на иконке левой кнопкой мыши. После того, как была выбрана Максимизирующая иконка и окно заняло весь экран, эта иконка становится Восстанавливающей и её повторный выбор возвращает окно в исходное положение.

Границы окна и углы рамки окна перемещаются, когда в "горячей зоне", там, где курсор мыши превращается в двунаправленную стрелку-индикатор, позволяющую измененить размеры окна, нажимается левая кнопка мыши и при нажатой левой кнопке, мышь перемещается. При этом возникнет и будет передвигаться по экрану, в соответствии с перемещениями мыши, пунктирная линия, указывая новый размер окна, который окно займёт в том момент, когда будет отпущена левая кнопка мыши. Если выбирается сторону рамки окна, происходят изменения только горизонтального или вертикального размеров окна. Если выбирается угол, то изменяются оба размера. Независимо от того, какая сторона рамки или угол выбирается, противоположная сторона или угол остаются закреплённым в текущей позиции.

1.2 Текстовый редактор системы

Редактирование текста в Smalltalk/Vпроизводится в текстовых панелях окон, и соответствует основным соглашениям системы о редактировании текста.

1.2.1 Вставка текста

Чтобы вставить (ввести) новый текст, спозиционируйте курсор в то место, куда должен вставляться текст и нажмите левую кнопку мыши. Тем самым, определяется вертикальное место вставки. Если начать печатать символы, пользуясь клавиатурой, место вставки начнёт непрерывно перемещается вправо. Если сделали ошибку, можно нажать на клавишу Backspace, тем самым вернуть курсор на один символ влево и удалить ошибочную литеру.

Чтобы переместить место вставки, переместите курсор в новую позицию, и нажмите левую кнопку мыши. Место вставки переместится на новую позицию курсора. Кроме того, клавиша Home перемещает место вставки в начало текущей строки (той, в которой в настоящее время находится место вставки), а клавиша End перемещает место вставки в конец текущей строки.

1.2.2 Клавиши Enter, Tab, Backspace, Delete и клавиши со стрелками

Нажатие клавиши Enter перемещает место вставки в начало новой строки. Если место вставки спозиционировано на середину строки, нажатие клавиши Enter разбивает строку на две строки, при этом текст справа от места вставки перемещается на новую строку.

Если нажатm клавишу табуляции Tab, в место вставки вставляется символ табуляции, а место вставки перемещается в следующую, определяемую табуляцией позицию. По умолчанию табуляция состоит из четырёх пробелов.

Если нажать клавишу Backspace, удаляется литера слева от места вставки, а если нажать клавишу Delete, удаляется литера справа от места вставки. Хотя удалённая литера и не помещается в буфер обмена, можно всегда отменить сделанное удаление, немедленно выбирая функцию Undo из меню Edit.

Четыре клавиши со стрелками, иногда называемые клавишами управления курсором, обеспечивают дополнительный способ перемещения места вставки и прокручивания. Когда активная панель — текстовая панель, клавиши курсора перемещают место вставки на литеру вверх, вниз, влево или вправо, в соответствии с нажатой клавишей. Когда активная панель — списковая панель, клавиши курсора вверх и вниз, в соответствии с нажатой клавишей, перемещают выбор опции из списка.

1.2.3 Функции выбора, замены, удаления, копирования и вставки

Чтобы начать процесс по выделению (выбору) текста в активном окне, надо сначала переместить курсор в позицию, с которой следует начать или закончить выделение, нажать левую кнопку мыши. Переместить курсор в то место текста, в котором следует завершить выбор, и, удерживая клавишу SHIFT, вновь нажать

левую кнопку мыши. Выделенный текст теперь будет простираться от первоначально отмеченной точки до точки текущего расположения курсора. Расширение при выделении текста может выполняться при удерживаемой клавише SHIFT и с помощью четырёх клавиш курсора, клавиш Home или End. Выбранный текст отображается в виде негативного изображения — белый текст на чёрном поле.

Можно выделить текст, используя метод протягивания курсора вдоль текста. Для этого надо поместить курсор в один их концов выделяемого текста, в так называемую точку привязки. Затем нажать левую кнопку мыши, и, удерживая её нажатой, переместить курсор в другой конец выделяемого текста. Обратите внимание, что по мере перемещения курсора, происходит выделение текста (его отображение в негативном виде), которое включает текущее положение курсора. Работа метода протягивания курсора вдоль текста завершается, когда отпускается кнопка мыши.

Чтобы выбирать единственную литеру, надо спозиционировать курсор на нужной литере, нажать левую кнопку мыши и использовать метод протягивания курсора вдоль текста, пока литера не примет негативное изображение.

Чтобы выбрать целое слово, достаточно спозиционировать курсор в любом месте слова и дважды щёлкнуть левой кнопкой мыши. Слово будет выбрано, что отобразится в его негативном изображении.

Чтобы выбирать строку, достаточно разместить курсор внутри границы окна слева от нужной строки и дважды щёлкнуть левой кнопкой мыши.

Чтобы быстро выбирать весь текст из активного в данный момент окна, достаточно выбрать опцию Select All (Выбрать Все) из меню Edit.

Чтобы заменить текст, надо выбрать тот текст, который будет заменён, и просто начать печатать новый текст, в этот момент выбранный текст будет заменён на новый.

Чтобы удалить литеру, надо воспользоваться клавишами Backspace или Delete (см. выше). Чтобы удалить большой кусок текста, надо выбрать нужную часть текста и либо использовать опцию Cut(Вырезать) или опцию Clear (Очистить) из меню Edit, либо просто нажать клавишу Backspace или клавишу Delete. Операции Cut и Backspace удаляют текст и помещают его в буфер обмена, с тем, чтобы потом его можно было использовать. Операции Clear и Delete только удаляют текст, но не помещают его в буфер обмена. Поэтому, если требуется восстановить текст после любой из этих операций, надо немедленно выбирать опцию Undo(Отмена) из меню Edit.

Несколько слов надо сказать о буфере обмена, который имеет текстовый редактор системы, он называется Clipboard (БуферОбмена) и может использоваться для того, чтобы перемещать текст с одного места в другое. Как сказано выше, текст помещается в буфер обмена при выполнении операции вырезки (Cut), но тоже самое происходит и при выполнении операции копирования (Copy).

Когда текст выделен, то после выбора опции Cut, он заменяет содержимое буфера обмена, а сам удаляется из панели. Если же выбирается опция Copy, копия выбранного текста заменяет содержимое буфера обмена, а он сам не удаляется из панели. После того, как текст помещён в буфер обмена, его можно вставить в любое место либо той же текстовой панели либо текстовой панели другого окна или заменить им некоторый существующий текст. Чтобы вставлять содержимое буфера обмена, надо определить место вставки и выбрать опцию Paste (Вставить) из меню Edit. Чтобы заменить некоторый текст на содержимое буфера обмена, надо сначала выбрать этот текст, а затем выбрать опцию Paste из меню Edit.

Функция Paste оставляет содержимое буфера обмена без изменений. Это означает, что один и тот же текст можно, например, вставлять несколько раз в различных местах. Обратите внимание, что операция Backspace заменяет содержимое буфера обмена на удалённую литеру, даже если литера — пробел. Так что, стоит взять за правило использовать только клавишу Delete, чтобы стереть символ, если знаете, что в буфере обмена находиться текст, который впоследствии предполагается использовать.

Так как тот же самый буфер обмена разделяется всеми приложениями, с помощью тех же операций можно легко перемещать текст не только между окнами системы Smalltalk, но и между окнами других приложений.

1.2.4 Выполнение (Do It) и показ (Show It)

Язык Smalltalk включает выражения, которые подобны выражениям в других языках программирования. В среде Smalltalk/V, текст выражения можно вводить в любую текстовую панель, в ней вычислять его и отображать результат, поскольку все текстовые панели системы поддерживают непосредственное вычисление выражений через меню окна с именем Smalltalk. Чтобы вычислить выражение, необходимо сначала выбрать его, а затем из меню Smalltalk (или из меню панели, которое содержит все опции этого меню) выбирать или опцию Show It (ПоказыватьЭто) или опцию Do It (СделатьЭто). Если выбирается опция Show It, выражение вычисляется, и символьное представление возвращаемого значения (возвращаемого объекта) выводится в панель после вычисленного текста. Если выбирается опция Do It, выражение вычисляется, но возвращаемое значение отбрасывается. Обратите внимание, что вычисляется только выбранный текст; остальной текст панели игнорируется. Дополнительные пробелы или в начале или в конце выбранного текста также игнорируются.

В текстовой панели можно выбрать и вычислить любое допустимое выражение или ряд выражений языка Smalltalk. Если необходимо, могут объявляться временные переменные.

Когда выбираются и вычисляются выражения языка, Smalltalk/V их сначала

компилирует и затем вычисляет. Если при трансляции обнаруживаются синтаксические ошибки, Smalltalk/V вставляет в исходный текст в месте ошибки сообщение об ошибке. Ради удобства пользователя, сообщение об ошибке сразу же выбирается (отображается в негативном виде), так что простое нажатие клавиши Backspace удаляет его, и тем самым восстанавливает исходное состояние панели. Устранив ошибку простым редактированием текста, его можно попытаться вычислить снова.

Совсем не обязательно заново набирать выражение, которое надо вычислить. Можно отредактировать некоторый существующий в панели текст, а затем его выбрать и вычислить. Поэтому имеет смысл построить окно (а ещё лучше файл) полезных выражений (шаблонов), которые можно редактировать, создавая новые выражения, нужные для вычислений. Файл с именем sample.sml содержит в качестве примеров несколько таких полезных выражений. Его, разумеется можно расширять или создать собственный файл шаблонных выражений.

1.2.5 Сохранение (Save) и Восстановление (Restore)

Текстовый редактор системы Smalltalk/V всегда работает с текстовой копией объекта системы, это может быть файл, строка, или некоторый Smalltalk-код. Например, в окне Transcript или любом рабочем окне как правило редактируют строки символов, а в текстовой панели окна просмотра Иерархии Классов редактируют код языка Smalltalk. Каждая панель, которая позволяет редактировать текст, имеет полный набор функций редактирования из меню Edit, и функции Save (Сохранить), Save As... (Сохранить Как...) and Restore (Восстановить) из меню File.

Поскольку редактируется копия, когда редактирование завершается, об этом необходимо сообщить системе, выбирая функцию Save (или Save As...). После чего отредактированный текст сохраняется в первоначальном объекте. Например, если редактируется файл, то файл перезаписывается с текстом, содержащимся в панели. Если необходимо, всегда можно восстановить отредактированный текст в первоначальное состояние.

Опция Save As... из меню File сохраняет копию содержимого активной панели на устройстве в файле, когда путь и имя указываются через открывающееся диалоговое окно. Таким образом, Save As... сохраняет сделанные в процессе редактирования изменения, в новом файле, оставляя оригинал без изменения. После этого новый файл становится открытым файлом. Если не было сделано изменений в тексте первоначального файла и была выбрана опция Save As..., то будут существовать два идентичных файла, каждый с собственным именем.

Функция Restore отбрасывает все изменения, сделанные в активной текстовой панели, начиная с того момента, когда информация сохранялась в последний раз. Если редактируется строка, то текст строки панели заменится на текст исходной

строки. Если редактируется файл, текст первоначально загруженного файла повторно прочитается в панель. Если редактируется Smalltalk-код, первоначальный текст кода вновь скопируется в панель.

1.2.6 Подсказчик (Prompter)

Prompter (Подсказчик) — специальный вид окна редактирования текста, которое полезно тогда, когда нужно, прежде чем продолжить процесс, запросить у пользователя дополнительную информацию. Подсказчики в Smalltalk/V реализованы как диалоговые окна, а строка подсказки — как сообщение внутри окна. Подсказка должна быть сформулирована так, чтобы пользователь знал, что ему отвечать. Пользователь вводит свой ответ в единственную текстовую панель окна. Если необходимо, в текстовую панель всегда можно вывести ответ по умолчанию.

Текстовый редактор, используемый подсказчиком — такой же как текстовый редактор, как и описанный выше, за исключением следующих различий:

- Диалоговые окна gподсказчика запрашивают информацию от пользователя и он или обязан ответить на вопрос перед тем, как продолжить взаимодействие с окном, которое было активным перед появлением подсказчика, или отказаться от взаимодействия с диалоговым окном, в противном случае система подаст звуковой сигнал. (Часто окно с такими свойствами называют модальными окнами.)

- Клавиши управления курсором “левая стрелка” и “правая стрелка” перемещают место вставки внутри поля редактирования подсказчика, а клавиши “стрелка вверх” и “стрелка вниз” циклически повторяют подсветку, тем самым по порядку выделяя поле редактирования и кнопки окна, позволяя с помощью клавиатуры сделать выбор любой нужный выбор.

- По умолчанию средствами управления в подсказчике являются кнопки OK и Cancel. Когда выбирается кнопка OK, подсказчик посылает текст из текстовой панели объекту, запросившему информацию. Когда выбирается кнопка Cancel, подсказчик посылает объекту, запросившему информацию, nil. Объект, запросивший информацию, должен знать, как обрабатывать полученную от промптера информацию. Нажатие клавиши Enter равносильно выбору кнопки OK. После того, как запрос удовлетворён, окно подсказчика закрывается.

2 Стандартные окна системы Smalltalk

Опишем те специальные окна, которые система Smalltalk/V предоставляет пользователю для разработки приложений, их сопровождения и отладки. От реализации к реализации окна видоизменяются, меняется число элементов в их меню, одинаково работающие опции меню могут размещаться в разных меню окна, но основные принципы организации работ окна и операции, выполняемая по одноимённым опциям меню, во всех реализациях практически одинаковы. Поэтому, как образец, мы рассмотрим окна системы SmalltalkExpress.

В Smalltalk-системах обязательно присутствуют следующие окна:

Class Hierarchy Browser — окно просмотра Иерархии Классов. Показывает взаимосвязи классов внутри системы Smalltalk, позволяет редактировать любой доступный пользователю код из любого класса.

ClassBrowser — окно просмотра Класса. Подобно окну просмотра Иерархии Классов, предоставляет возможность редактировать методы, сконцентрированные в конкретном классе иерархии.

Inspector — окно инспектора. Позволяет исследовать структуру объектов и редактировать её. Эти окна полезны при отладке.

Walkback и Debugger — окно ОстановаСистемы и окно Отладчика. Предоставляют возможность просмотреть состояние программы в момент ошибки и являются средствами отладки на уровне системы Smalltalk. Здесь рассматриваться не будут, работу с ними предполагается изложить в другом месте.

MethodBrowsers — окно просмотра Методов. Предоставляет возможность просматривать и редактировать список методов.

MessageBrowsers — окно просмотра Сообщений. Удобный инструмент исследования селекторов, которые составляют метод, и определения перекрёстных ссылок для каждого селектора, что позволяет получать списки всех отправителей и разработчиков конкретного сообщения.

Все окна просмотра состоят по крайней мере из двух панелей: панели списка и текстовой панели. Выбор из списка отображает связанную с выбранным элементом списка информацию в текстовой панели. Этот текст может изменять и в конечном счёте сохранять, тем самым изменяя систему.

2.1 Окно просмотра Иерархии Классов

Окно просмотра Иерархии Классов позволяет исследовать взаимосвязи классов внутри системы Smalltalk/V и редактировать их содержимое.

2.1.1 Как открыть окно просмотра Иерархии Классов

Чтобы открыть окно просмотра Иерархии Классов, надо из меню File любого открытого окна системы (всегда, после старта системы, доступно окно системной информации Transcript) выбрать опцию Browse Classes (ПросмотрКлассов) или нажать Alt + B.

Окно просмотра Иерархии Классов разделено на пять панелей.

Список, отражающий текущее состояние иерархии классов, появляется в левой верхней панели. В этой панели, имена всех классов системы приводятся в иерархическом порядке. Класс Object (Объект) является первым в этом списке, поскольку он — базовый, самый верхний класс в иерархии. Все другие классы — подклассы класса Object и, следовательно, их имена отображаются в иерархии смещёнными вправо. Имена подклассов классов первого уровня тоже смещены вправо, и так далее.

Список переменных — средняя панель в верхней половине окна просмотра. Она расположена ниже множества “радио-кнопок” instance/class (экземпляр/класс). Панель списка переменных показывает наследование переменных для выбранного класса по отношению к его суперклассам. Наследование переменных показывается в обратном порядке: сначала, на самом верху списка, переменные, определённые в выбранном классе. Суперкласс выбранного класса отображается его именем с приставкой и суффиксом в виде чёрточки, то есть в виде “-SuperclassName -”. Затем перечисляются наследуемые переменные указанного суперкласса, заканчивается такой список переменными, наследуемыми из класса Object. Переменные внутри каждого класса перечисляются в алфавитном порядке. Содержимое панели списка переменных зависит от выбранной “радио-кнопки”: если выбрана кнопка instance, отображаются переменные экземпляра выбранного класса, а если выбрана кнопка class — переменные класса.

Панель списка методов — панель справа от панели списка переменных. В ней отображается в соответствии с выбранной “радио-кнопкой” или список методов экземпляра или список методов класса. Если в панели списка переменных выбирается некоторая переменная, то список методов уменьшается до подмножества тех методов данного класса, которые ссылаются на выбранную переменную.

Непосредственно Smalltalk-код отображается в панели содержания — текстовой панели, которая занимает нижнюю половину окна. Когда в панели иерархии классов выбирается класс, в панели содержания отображается сообщение, которое определяет выбранный класс. Когда выбирается метод, в этой панели отображается код выбранного метода. Панель содержания обеспечивает возможности по редактированию и определению как самого класса, так и исходного текста его методов.

2.1.2 Просмотр иерархии

Чтобы выбрать класс для просмотра, надо переместить курсор в списковую панель иерархии классов, верхнюю левую панель окна, и щёлкнуть левой кнопкой мыши на имени класса. Smalltalk/V отобразит определение класса в панели содержания, а список переменных и методов, реализуемых классом, в панели переменных и панели списка методов, соответственно. Используя полосу прокрутки, можно просмотреть весь список классов в иерархии.

Меню Classes содержит функции, которые позволяют поддерживать иерархию, производя добавление и удаление подклассов, модифицировать панели иерархии после сделанных в иерархии изменений, открывать окно просмотра класса на выбранном классе, записывать определения класса и методов класса в файл. Это меню дублируется, как меню панели, и возникает при нажатии в панели просмотра иерархии классов правой кнопки мыши. Опции (элементы) меню Classes следующие:

Add Subclass... (ДобавитьПодкласс...) — позволяет добавить подкласс к выбранному классу.

File Out... (ВывестиВФайл...) — записывает определение выбранного класса наряду со всеми его методами класса и экземпляра в файл. Файл создаётся в специальном формате файла регистрации изменений (см. раздел “Автоматическая регистрация изменений”). Сначала появляется диалоговое окно, с предлагаемым системой по умолчанию именем файла, которое строиться по правилу <укороченное имя класса>. cls. Класс будет сохранён в файле в текущем каталоге, если он не будет изменён в процессе диалога. Подклассы выбранного класса в таком файле автоматически не сохраняются. Функция File Out... не воздействует на класс в самой системе.

Update (Модифицировать) — требует, чтобы окно просмотра иерархии классов повторно вычислило список классов в иерархии и отобразило его. Это надо делать всегда, если, пользуясь другими окнами просмотра, выполнялись операции удаления или добавления классов, иначе список в панели не будет отражать текущее состояние иерархии.

Browse (Просмотреть) — открывает отдельное окно просмотра класса на выбранном классе. Это может быть удобно, когда необходимо при просмотре одного класса обратиться к другому.

Hide/Show (Скрыть/Показать) — переключатель, который позволяет скрывать или показывать подклассы выбранного класса. Это позволяет укорачивать список классов, скрывая те их них, которые в настоящее время не

представляют интереса. Если подклассы класса скрыты, после имени класса в панели списка иерархии класса появляется многоточие (...). В такой ситуации в меню Classes прочитается опция Show Subclasses (ПоказатьПодклассы). Чтобы показать подклассы класса, можно выбрать эту функцию. То же самое последует, если просто дважды щёлкнуть на нужном классе. Если выбран класс, подклассы которого отображаются в панели, то в меню Classes прочитается опция Hide Subclasses (СкрытьПодклассы). Выберите её, или просто снова дважды щёлкните на классе, чтобы свернуть иерархию. Если класс не имеет подклассов, отображаемая блеклым шрифтом опция Hide/Show меню Classes, указывает, что эта функция для текущего выбора не доступна.

FindClass... (НайтиКласс...) — вызывает простое диалоговое окно, позволяя ввести имя того класса, который надо отыскать в иерархии. Эта опция чрезвычайно полезна при поиске подклассов, которые содержаться в иерархии на уровнях, которые в настоящее время в панели не показываются.

Remove Class (УдалитьКласс) — удаляет выбранный класс из системы.

2.1.3 Добавление нового класса

Чтобы добавить подкласс в класс, сначала в списковой панели иерархии классов, надо выбрать класс, который будет суперклассом нового класса. Затем из меню Classes, надо выбрать опцию Add Subclass.... (ДобавитьПодкласс...), которая выводит на экран диалоговое окно с именем “Add a SubClass”, отображающее в поле с меткой Superclass: (Суперкласс:) имя выбранного класса, и предоставляющее поле редактирования New Class: (Новый Класс:), куда надо ввести имя нового класса.

Кроме того, в диалоговом окне находятся радио-кнопки, которые позволяют выбрать тип создаваемого подкласса. Тип подкласса зависит от того, должны ли объекты, принадлежащие классу, содержать именованные переменные экземпляра, индексированные переменные экземпляра или массивы байт. После того, как выбор сделан, сформируется новый подкласс, а список иерархии класса автоматически модифицируется.

Чтобы для класса определить переменные экземпляра, переменные класса и словари пула, сначала надо выбрать этот класс в панели списка иерархии классов. На это Smalltalk/V отреагирует отображением текущего определения класса в панели содержания. Определение включает в себя суперкласс класса, переменные экземпляра, переменные класса и словари пула. Определение класса изменяется, когда редактируется текст в панели содержания. Затем в меню File необходимо выбрать опцию Save (Сохранить) или нажать Alt + S. Определение

класса будет модифицировано согласно сделанных изменений. Система автоматически перетранслирует все методы в классе и все его подклассы. Кроме того, в системный файл change.log, будет записано сообщение, определяющее новый класс.

Напомним, что панель содержания, как всякая текстовая панель, имеет достаточно большое меню, которое вызывается при нажатии в панели правой кнопки мыши. Меню текстовой панели содержит в себе все элементы меню Edit и Smalltalk, а также опцию Save. Так что большинство операций здесь можно делать не обращаясь к меню окна.

Изменение определения класса ведёт к немедленным последствиям, так что все будущие новые экземпляры класса будут иметь новую структуру. Надо быть очень осторожным при изменении тех классов, которые используются средой Smalltalk/V. Пока нет уверенности в том, что, все сделано корректно, следует определять подклассы основных классов, а не изменять структуру существующих классов.

2.1.4 Удаление экземпляров класса

Когда меняется определение класса или класс удаляется, а в системе все ещё остаются экземпляры этого класса, возникает окно Walkback (ОстановаСистемы). Перед тем, как система позволит сделать и сохранить изменения, надо удалить все экземпляры модифицируемого класса!

Для классов, связанных с окном, удостоверитесь, что в открытых окнах системы сохранена вся необходимая информация, и выполните выражение Notifier reinitialize. При этом закроются все открытые окна и повторно инициализируется окно системы Transcript.

Если экземпляры не связанны с окном, или если некоторые из них все ещё остаются в системе, пробуйте вычислить следующий код, но сначала сохраните образ системы:

```
<MyClass> allInstances do: [ :each | each become: String new]
```

2.1.5 Удаление класса

Чтобы удалить класс, надо сначала в панели списка иерархии выбрать тот класс, который будет удаляться. Затем из меню Classes выбрать опция Remove Class (УдалитьКласс). Система попросит подтверждения на выполнение такой операции.

Когда класс удаляется, автоматически удаляются все методы данного класса. Smalltalk/V не допустит удаления класса, если он имеет подклассы или если в

среде есть экземпляры класса. Если существует подкласс или экземпляр, при попытке удаления класса возникнет окно Walkback, объясняющее причину останова. Не забудьте, что Smalltalk автоматически удаляет (собирает как мусор) любой объект, на который нет ссылок со стороны других объектов системы. Чтобы удалить экземпляр класса, ссылка на экземпляр должны быть изменены так, чтобы она относилась к объекту другого класса.

Если в системе все же остаётся ссылка на удалённый класс, эта ссылка теперь укажет на этот класс как на deletedClass (удаленныйКласс) и не будет удаляться сборщиком мусора. Это сделано в целях безопасности, хотя такие фиктивные ссылки и занимают место в образе.

Чтобы выявить все объекты, которые обращаются к объекту-приёмнику, можно послать объекту сообщение allReferences. Как ухе отмечалось, переназначение указателя, который ссылается на экземпляр удаляемого класса, например, на объект String new, решит проблему с указателями.

2.1.6 Просмотр методов

Панель списка методов — верхняя правая панель окна. В классе есть два вида методов: методы экземпляра и методы класса. Список, который появляется в панели списка методов, определяется тем, какая из двух кнопок (экземпляр или класс) выбрана в панели, расположенной выше панели списка переменных.

На выделение селектора сообщения в панели списка методов, Smalltalk/V отреагирует тем, что разместит исходный текст метода, реализующего выбранное сообщение, в панели содержания. Всегда можно переместить курсор или прокрутить панель списка методов и выбрать другой метод.

Меню Methods содержит восемь функций:

New Method (Новый Метод) — используется для того, чтобы добавить новый метод экземпляра или класса к выбранному классу. Процедура добавления новых методов описана ниже.

Senders (Отправители) — заставляет Smalltalk/V искать все те методы среды, которые отправляют (вызывают) метод с выбранным селектором сообщения. Появляется окно отправителей, которое является своего рода окном просмотра Методов, и отображает каждый метод (и класс), который посылает сообщение с выбранным селектором сообщения.

Implementors (Реализаторы) — заставляет Smalltalk/V искать все те классы среды, которые реализуют (определяют) методы с выбранным селектором сообщения. Появляется окно реализаторов — окно просмотра Методов, оно отображает имена всех классов, которые реализуют методы с выбранным селектором сообщения.

Local Senders (ЛокальныеОтправители) — во всем подобна опции Senders (Отправители), за исключением того, что среда поиска отправителей ограничена текущим классом и его подклассами.

Local Implementors (ЛокальныеРеализаторы) — во всем подобна опции Implementors (Реализаторы), за исключением того, что среда поиска реализаторов ограничена текущим классом и его подклассами.

Messages (Сообщения) — заставляет Smalltalk/V собрать в список все селекторы методов, вызываемых в выбранном методе. Открывается окно SelectorBrowser, отображающее этот список сообщений.

File Out... (ВывестиВФайл...) — заставляет Smalltalk/V послать выбранный метод в файл на диске, который создаётся в формате файла регистрации изменений.

Remove (Удалить) — удаляет из класса выбранный метод. Список методов немедленно модифицируется.

Меню Methods дублируется, как меню панели, и возникает при нажатии в панели методов правой кнопки мыши.

Меню Variables работает вместе с панелью списка переменных и помогает сузить круг методов, отображаемых в панели списка методов. В классах с большим числом методов, сужение списка может быть очень полезно при быстром поиске нужного метода или методов. Меню Variables дублируется, как меню панели, и возникает при нажатии в панели переменных правой кнопки мыши.

Меню Variable содержит три элемента:

Assigned (Назначение) — вызывает панель списка методов и отображает в ней только те методы, в которых назначается переменная, выбранная в списковой панели переменных.

Used (Использование) — вызывает панель списка методов и отображает в ней только те методы, в которых вызывается переменная, выбранная в списковой панели переменных.

Both (И то и другое) — выбор по умолчанию, вызывает панель списка методов и отображает в ней только те методы, в которых или назначается или используется переменная, выбранная в списковой панели переменных.

Способ сужения круга выбираемых методов обозначается в меню Variables «птичкой» (checkmark — маркером выбора), которая появляется рядом с одним из элементов меню. Это взаимно исключающий выбор, в любой момент может быть выбранным только один элемент.

Выбор элемента из списка переменных можно отменить, если нажать или на радио-кнопку или на имя класса в панели иерархии классов.

2.1.7 Добавление, изменение и удаление метода

Чтобы увидеть исходный текст метода, надо переместить курсор в панель списка методов и выбрать соответствующий селектор сообщения. Панель содержания — текстовая панель, используется для добавления и изменения методов. Панель списка методов используется для удаления методов из выбранного класса. Все функции из меню панели содержания ведут себя точно так же, как и во всех текстовых панелях.

Чтобы добавить новый метод, сначала надо переключить радио-кнопку, определяя какой метод — экземпляра или класса — будет создаваться, а затем выбрать опцию New Method (НовыйМетод) из меню Methods. Сделанный выбор выводит в панель содержания доступный для редактирования шаблон. В соответствии с шаблоном следует ввести правильно сформированный текст нового метода и использовать функцию Save из меню File (или комбинацию клавиш Alt+S), вызывая компилятор и устанавливая новый метод в систему.

Чтобы изменить существующий метод, сначала надо выбрать его в панели списка методов, отредактировать исходный текст в текстовой панели, а затем выбрать функцию Save.

Если с процессе трансляции добавляемого или изменяемого метода будет обнаружена синтаксическая ошибка, в соответствующем месте текста метода появляется сообщение, объясняющее причину ошибки. Сообщение об ошибке сразу выбирается (отображается в негативе). Чтобы удалить сообщение об ошибке, достаточно нажать клавишу Backspace, а затем заново отредактировать текст и снова использовать функцию Save.

Когда метод без синтаксических ошибок успешно откомпилируется, система Smalltalk/V автоматически установит его в класс и все будущие вызовы этого метода будут использовать его новую версию. Исходный текст нового или изменяемого метода будет записан в файл change.log.

Можно отредактировать текст любого существующего в классе метода. Имя нового метода принимается из первой строки текста. Но, если Вы изменяли имя метода и провели операцию сохранения, то будет создан новый метод с новым именем, а первоначальный метод останется без изменений. Это чрезвычайно полезно при создании внутри класса разновидностей метода.

Чтобы удалить существующий метод из некоторого класса, надо выбрать селектор метода в панели списка методов, и выбрать опцию Remove (Удалить) из меню Methods. Список методов немедленно модифицируется, показывая, что выбранный метод удалён из класса.

2.2 Окно просмотра Класса

Окно просмотра Класса может быть открыто одним из двух способов. Можно или посылать сообщение edit любому классу, или выбрать опцию Browse (Просмотреть) из меню Classes окна просмотра Иерархии Классов.

Переместите курсор в любую текстовую панель. Напечатайте имя класса, который хотите просмотреть, сопровождая его селектором edit, и выберите это выражение. Выберите из меню Smalltalk опцию Do it. Откроется окно просмотра класса на указанном в выражении классе.

Заголовок окна просмотра Класса идентифицирует просматриваемый класс. В этом окне можно просматривать, добавлять и изменять только методы выбранного класса.

Списковая панель, определяющая просматриваемый словарь — верхняя левая панель, содержит список, позволяющий сделать всего два выбора: class или instance (класс или экземпляр). Если выбирается class, то в панели списка методов, расположенной ниже первой панели, немедленно отображается список методов класса. Если выбирается instance, то отображается список методов экземпляра. Выбирая в этой панели метод, его исходный текст можно увидеть в панели содержания — самой большой панели, расположенной справа от списковых панелей, в которой можно редактировать, перетранслировать и добавлять новые методы.

Все функции всех меню этого окна аналогичны соответствующим элементам окна просмотра Иерархии Классов. Обратите внимание, что меню Methods окна Class Browser — подмножество меню Methods окна просмотра Иерархии Классов.

2.3 Инспекторы

Окно Inspector (Инспектор) используется как инструмент для исследования и изменения структуры любого объекта системы. Чтобы открыть инспектор на объекте, надо послать сообщение inspect этому объекту. Например, чтобы открыть инспектор на объекте Display pen (пере, связанном с экраном), надо вычислите выражение Display pen inspect.

Можно поступить и проще: выбрать в любой текстовой панели имя объекта, который надо просмотреть (например, Display pen) и выбрать опцию InspectIt из меню Smalltalk. Это приведёт к тем же последствиям, что и приведённое выше вычисление выражения. Так можно выбирать и просматривать объекты почти из любого контекста.

Окно Инспектора имеет две панели. Левая списковая панель — панель списка переменных экземпляра. Правая текстовая панель — панель содержания, ото-

бражающая значение выбранной переменной экземпляра.

Списковая панель переменных экземпляра показывает все переменные экземпляра осматриваемого объекта (учитывая механизм наследования). Первый элемент в списке — всегда имя self, которое ссылается на инспектируемый объект. Затем перечисляются именованные переменные экземпляра, если они есть. Если объект имеет индексированные переменные экземпляра, то они перечисляются последними, указанием числовых индексов.

Когда из списка выбирается одна из переменных, её текущее значение отображается в панели содержания переменной экземпляра. Если выбрать self, то отобразится текущее значение инспектируемого объекта. Если выбрать в меню Inspect опцию Inspect (или сделать двойной щелчок на переменной экземпляра), открывается новое окно Инспектора на выбранной переменной экземпляра.

Панель содержания переменной экземпляра — текстовая панель. Чтобы вызвать нужные функции, можно использовать соответствующие опции из меню File, Edit, **Smalltalk** или воспользоваться опциями из меню текстовой панели. Панель можно использовать и для вычисления любого желаемого выражения. Есть две очень важные особенности этой панели:

- Любое выражение, которое вычисляется, компилируется в окружении, определяемом осматриваемым объектом. Это означает, что в выражениях можно использовать имена всех переменных экземпляра.

- Если выбирается опция Save, все содержимое текстовой панели компилируется и вычисляется, а результат заменяет текущее значение выбранной переменной экземпляра. Если выбирается опция Restore (Восстановить) из меню File, то **Smalltalk/V** отобразит в текстовой панели текущее значение выбранной переменной экземпляра.

При инспектировании словарей, возникают некоторые особенности. Напомним, что словарь — объект, который содержит в качестве своих элементов ассоциативные пары, связывающие ключи со значениями. Точно так же как и обычные инспекторы, инспекторы словарей имеет те же две панели, однако, в панели списка переменных экземпляра перечисляются ключи словаря, а не имена переменных экземпляра и индексы. Когда в этой панели выбирается ключ, связанное с ним значение отображается в панели содержания. Чтобы увидеть все это, откройте инспектор на словаре ColorConstants, выполняя выражение ColorConstants inspect.

Обратите также внимание на то, что в списковой панели инспекторов словарей отсутствует self, а к строке меню окна добавляется меню Dictionary (Словарь). Меню Dictionary имеет три опции:

Add (Добавить) — позволяет добавить в словарь новый элемент. Система вызывает диалоговое окно (Подсказчик), спрашивая о новом ключе. Пока новый ключ не выбран, связанное с ним значение равно nil, но его можно изменить в панели содержания, а затем сохранить.

Remove (Удалить) — позволяет удалить из словаря ассоциативную пару с выбранным ключом.

Inspect (Осмотреть) — создаёт окно Инспектора на выбранном ключе.

2.4 Окно просмотра Методов

Окно просмотра Методов (Method Browser) позволяет просматривать и редактировать список методов. Есть четыре основанных на меню способа открыть окно просмотра Методов. Выбор опции Senders в меню любого окна, которое предлагает эту опцию (например, меню Methods окна просмотра Иерархии Классов), открывает окно просмотра Методов на списке всех методов системы, которые посылают выбранный селектор сообщения. Выбор опции Implementors в меню любого окна, которое предлагает эту опцию, откроет окно просмотра Методов на списке всех методов, которые реализуют выбранный селектор сообщения. Кроме того, опции Local Senders и Local Implementors в меню Methods окна просмотра Иерархии Классов открывают окна просмотра Методов на соответствующем списке методов, выбор которых ограничен классом и его подклассами.

Окно просмотра Методов часто упоминается как окно отправителей или как окно разработчиков в зависимости от отображаемой окном информации.

Окно просмотра Методов имеет два панели: панель списка методов в верхней части и текстовую панель в нижней части. Панель списка методов отображает список методов, идентифицированных селектором сообщения и классом. Когда в списке выбирается метод, его исходный текст отображается в текстовой панели.

Меню Methods в строке меню окно отправителей или разработчиков имеет шесть элементов:

Remove from List (Удалить из Списка) — обеспечивает удобный способ отбросить записи из панели списка методов, которые в данный момент не нужны; удаление записи не удаляет метод из класса.

Senders (Отправители) — заставляет Smalltalk/V искать методы, которые посылают выбранный селектор сообщения; на найденных методах открывается новое окно просмотра Методов.

Implementors (Реализаторы или Разработчики) — заставляет Smalltalk/V искать методы, которые реализуют (определяют) выбранный селектор сообщения; на найденных методах открывается новое окно просмотра Методов.

Local Senders (ЛокальныеОтправители) — открывает окно просмотра Методов, подобно опции Senders, за исключением того, что окружение поиска отправителей ограничивается текущим классом и его подклассами.

Local Implementors (ЛокальныеРеализаторы) — открывает окно просмотра Методов, подобно опции Implementors, за исключением того, что окружение поиска разработчиков ограничивается текущим классом и его подклассами.

Messages (Сообщения) — открывает окно просмотра сообщений, перечисляющее селекторы методов, вызываемые в выбранном методе.

Текстовая панель позволяет просматривать и редактировать исходный текст выбранного метода. При редактировании методов в этой панели, можно, как и в других окна, использовать соответствующие опции из меню File, Edit и Smalltalk. Когда отредактированный текст сохраняется, метод перетранслируется.

2.5 Окно просмотра Сообщений (Селекторов)

Окно просмотра Сообщений (Message (Selector) Browser) имеет три панели, которые позволяют полностью идентифицировать сообщение для каждого селектора в методе, на котором было открыто окно просмотра.

Левая верхняя списковая панель отображает селекторы всех сообщений, представленных в методе, отображаемом в нижней текстовой панели окна просмотра. Кроме того, окно имеет ещё одну списковую панель (правую верхнюю), благодаря которой можно просмотреть список методов (отправителей или реализаторов) для селектора, выбранного в панели селекторов.

Окно просмотра Сообщений часто упоминается просто как окно Сообщений, так как первичная списковая панель отображает найденные в методе селекторы, которые часто называют сообщениями.

Если в панели списка селекторов выбрать селектор, то в тексте метода, отображаемом в нижней текстовой панели, выберется (выделится) полное сообщение, связанное с этим селектором. Это может оказаться чрезвычайно полезно при попытке идентифицировать сложные сообщения, в котором параметры размещаются вместе с компонентами селектора.

Меню Selector имеет два элемента:

Senders — заставляет Smalltalk искать методы, которые посылают выбранный селектор сообщения; список методов (по классам) отображается в правой панели списка методов.

Implementors — заставляет Smalltalk искать методы, которые реализуют (определяет) выбранный селектор сообщения; список методов, (по классам) отображается в правой панели списка методов.

Выбор любой строки в правой панели, приводит к отображению в нижней текстовой панели текста выбранного метода.

Меню Methods в строке меню окна Сообщений имеет те же самые четыре элемента, что и в меню окна просмотра Методов (класс MethodBrowser является суперклассом класса SelectorBrowser) и выполняет все описанные ранее операции.

3 Управление системой в целом

3.1 Выход из системы и сохранение образа

Образ (image) — хранилище всех Smalltalk-объектов, кода и данных, которые вместе составляют среду Smalltalk/V. Когда система запускается, образ считывается из файла V.EXE, который может изменяться пользователем, и неизменяемых DLL-файлов. Все объекты загружаются в память. Среди них — окна, которые появляются на экране.

Так как Smalltalk/V - интерактивная модифицируемая среда, образ в процессе работы пользователя над приложениями постоянно изменяется. Изменения не записываются на диск, пока об этом не попросят. Это можно сделать в любое время, выбирая опцию Save Image из меню File. Можно также сохранить образ и при выходе из системы, что происходит, когда выбирается опция Exit Smalltalk/V... (Выйти из Smalltalk/V...) из системного меню окна Transcript.

В последнем случае возникнет диалоговое окно, в котором спрашивается: надо ли сохранить в файле V.EXE произведённые в сеансе работы изменения?

Если в ответ на вопрос Save Image? (Сохранить Образ?) выбрать “Да”, инициализируется процесс сохранения полного состояния системы, включая расположение и содержимое всех окон (то же самое происходит, когда выбирается опция Save Image из меню File). В следующий раз, когда система будет запускаться, она возникает в том виде, в каком была в момент сохранения образа.

Если выбрать “No” — все изменения, сделанные в Smalltalk/V в течении текущего сеанса работы, будут отметены (забыты), но, несмотря на это, файл change.log их все же запомнит. Когда в следующий раз Smalltalk/V будет запущен, он возникнет в предыдущем сохранённом состоянии, игнорируя все изменения, сделанные в не сохранявшем образ сеансе работы.

Если же выбрать “Cancel” (Отменить) — произойдёт возврат в среду Smalltalk/V. Это то же самое, что и отказ от какого-либо выбора из меню.

Файл V.EXE представляет последнюю сохранённую версию; следовательно, он становится отправной точкой в случае аварийного отказа системы или серьёзной ошибки. По этим причинам, всегда следует хранить резервную копию файла V.EXE и тесно связанных с ним файлов sources.sml и change.log. В последующих разделах более подробно объясняются связи между `v.exe` и DLL-файлами, приводятся команды для их поддержки, а так же даются советы о том, как восстановить системы после её аварийного отказа или в случае возникновения других проблем.

3.2 Файл автоматической регистрации изменений

Когда происходит определение и изменение пользователем классов или методов, Smalltalk/Vрегистрирует все эти изменения в файле журнала изменений системы `change.log`. Формат файла `change.log` очень похож на тот, который описан Гленом Краснером (Glenn Krasner, Smalltalk-80: Bits of History and Words of Advice, Addison-Wesley, 1983). Краткое описание этого формата, использующего нотацию расширенной формы Бэкуса-Наура, следующее:

<rule> changeLog = textChunk.
<rule> textChunk = sourceCode | classDefinition | expression | imageComment.
<rule> sourceCode= '!' className 'methods ! 'method '!' '!'.
<rule> classDefinition = '"define class" 'classDefinitionMessage '!'.
<rule> expression = '"evaluate" 'evaluatedText '!'.
<rule> imageComment = '" *** saved image on:' dateAndTime '*** " !'.

Где

className — имя класса.

method — допустимый метод.

classDefinitionMessage — допустимое сообщение определения класса, посылаемое суперклассу класса.

evaluatedText — текст, который вычисляется посредством Do It или Show It.

dateAndtime — символьное представление даты и времени сохранения образа.

Система Smalltalk/V поддерживает указатели из компилируемых методов среды на последнюю версию исходного текста. Если исходный текст никогда не меняется, указатель направлен на файлы *.DLL. Иначе, указатель направлен на файлы change.log и sources.sml. Как результат, при просмотре в классе методов, происходят обращения к диску. Кроме того, так как файл V.EXE меняется сам и содержит указатели и на меняющиеся файлы change.log и sources.sml, всегда надо поддерживать эти три файла как единое целое.

Итак, система Smalltalk/V фактически состоит из динамических библиотек, которые содержат виртуальную машину Smalltalk и примитивные методы, плюс следующие четыре файла:

`v.exe` — файл, который содержит комбинацию кода выполняемой программы (.EXE) и образа (набора объектов). Этот файл изменяется всякий раз, когда образ сохраняется.

`vw.exe` — очень маленький файл, который содержит выполняемый код, используемый системой разработки программ.

`change.log` — постоянно модифицируемый файл регистрации изменений исходного кода и вычисляемых выражений.

`sources.sml` — файл, содержащий исходный текст приложений и представляющий основное содержание или версию вашей системы.

По этим причинам, нельзя редактировать и изменять файл регистрации изменений! Его можно просматривать, пользуясь функцией Open... из меню File, можно, пользуясь им, повторно устанавливать в систему методы и определения классов. Именно эта возможность используется при восстановлении системы, если она разрушилась.

Все важные для системы события автоматически регистрируются Smalltalk/V. Каждый раз, когда сохраняется образ, соответствующее сообщение с указанием даты и времени записывается в файл `change.log`. Каждое выражение, которое вычисляется с помощью Do It или Show It, также регистрируется. Кроме того, каждый раз, когда из класса удаляется метод, регистрируется соответствующее сообщение. И если по каким-либо причинам образ не был сохранён, можно все ещё восстановить нужные изменения — записи, которые производились в файле регистрации изменения `change.log`, все ещё доступны. Не сохранять образ иногда полезно, например, когда разрабатывается что-то новое, и в ходе разработки были допущены ошибки, которые хотелось бы отбросить.

Поскольку в процессе работы, размер файла change.log постоянно растёт, его периодически необходимо сжимать — уменьшать до последней копии каждого нового или изменённого метода. Для корректного проведения операции сжатия, на диске должно быть достаточно места и для нового и для старого файла change.log.

Чтобы сжать файл регистрации изменений, в любой текстовой панели (например, в окне Transcript) надо вычислить выражение Smalltalk compressChanges.

Если на диске на хватает места, надо удалить некоторые файлы, а затем снова повторить вычисление.

Как было сказано, все изменения регистрируются в файле change.log и эти изменения могут быть сжаты. Но когда все новые классы и методы отлажены, можно их интегрировать в систему, вычисляя выражение Smalltalk compressSources (производя тем самым сжатие файлов-источников change.log и sources.sml). При этом будут сделаны две вещи:

- Копия самой последней версии исходного текста каждого метода вне основной системы, содержащаяся в файлах `change.log` или `sources.sml`, будет сохранена в файле `sources.sml`. Изменения сохранятся в сжатом формате.

- Будет создан пустой новый файл `change.log`.

Периодические сжатия источников сохраняет дисковое пространство и очищает файлы от ненужных дублей исходных текстов из файла change.log. Дополнительно, исходный текст, содержащийся теперь в файле sources.sml находится в форме, которая доступна для совместного использования группой разработчиков.

Как результат взаимосвязи файлов `v.exe`, `sources.sml` и `change.log`, чрезвычайно важно поддерживать текущие резервные копии трёх этих файлов как единого целого, хранящего ядро и исходные тексты системы. Настоятельно рекомендуется делать такое резервное копирование перед каждым сжатием источников, чтобы можно было восстановить систему, если что-нибудь пройдёт неправильно.

Отметим здесь ещё одну полезную операцию. Поскольку экземпляры класса Symbol как мусор автоматически не собираются, неиспользуемые имена имеют тенденцию медленно накапливаться и засорять систему. При вычислении выражения Symbol purgeUnusedSymbols, имена, которые больше не вызываются, удаляются из системы. Число удалённых имен и количество освобождённого пространства обычно невелико.

3.3 Выживание в случае аварийного отказа системы

Smalltalk/V очень эластичен. Но, к сожалению, могут случаться аварии, особенно, если производятся значительные изменения в системе Smalltalk/V. Характерная черта Smalltalk/V — автоматическая регистрация изменений, в большинстве случаев даёт возможность восстановить работоспособность системы. Беды происходят не потому, что Smalltalk/V содержит в себе ошибки, а потому что это — модифицируемая среда разработки программы. Smalltalk/V позволяет изменить почти все, даже если это наносит вред среде.

Например, если допущена ошибка при замене метода для окна Walkback, или если произошли сбои в аппаратуре, появиться сообщение об ошибке: “doesNotUnderstand: is missing”. Это означает, что Smalltalk/V не может найти код, который отображает соответствующее walkback-сообщение и потому завершает работу.

Все это не представляет никакой проблемы, так как можно очень быстро восстановить большую часть, если не все, из того, что было сделано. Фактически, экспериментирование с системой — хороший способ научиться с ней работать во всех ситуациях, но прежде обязательно надо удостовериться, что выполнены следующие меры предосторожности:

- Есть надёжная резервная копия файлов `v.exe`, `change.log` и `sources.sml`. Помните, файлы `v.exe` и `change.log` работают вместе. Смешивание старого `change.log` с более новым файлом `v.exe` может привести к невозможности обращения к исходному тексту некоторых методов. Всегда копируйте файлы `v.exe`, `change.log` и `sources.sml` вместе!

- Перед тем, как пытаться сделать то, что может разрушить систему, пользуясь опцией Save Image из меню File, сохраните образ системы. Выполнение этого правила сохранит большую часть проделанной работы. Если произошёл сбой системы, всегда можно перезапустить Smalltalk/V, используя только что сохранённый образ. Если окажется, что использовать опцию Save Image невозможно, и произошёл сбой, можно использовать функцию Open... из меню File, чтобы исследовать файл change.log.

- Если проводились эксперименты с системой Smalltalk/V, и есть причины считать, что система повреждена, или есть изменения, от которых стоит избавиться, не сохраняйте образ. Выберите опцию Exit Smalltalk/V... из системного меню окна Transcript, и нажмите кнопку No, чтобы выйти из системы без сохранения образа. Помните, большинство сделанных изменений находятся в файле change.log, и то, что нужно, можно установить снова.

- Если система Smalltalk/V все же терпит крах, не паникуйте. Сделайте копии файлов `v.exe`, `change.log` и `sources.sml`. Информация относительно того, как восстановить систему содержится в следующем разделе.

3.4 Восстановление после аварийного отказа системы

Если система потерпела крах, перед тем как что-либо предпринять, обязательно надо сделать копии файлов `v.exe`, `change.log` и `sources.sml`. Чтобы восстановить ранее сделанную работу, надо действовать по следующей схеме:

- Проверить запускается ли система Smalltalk/V. Если система не запускается, надо взять самую последнюю резервную копию файлов `v.exe`, `change.log` и `sources.sml` и запустить систему.

- Теперь, когда есть работающая система, используя опцию Open... из меню File, надо просмотреть тот файл change.log, который использовался в момент краха системы и содержит все происходившие изменения и все вычислявшиеся выражения.

- В файле change.log надо найти запись о том, когда сохранялся тот образ, который выполняется в настоящее время. Ведь каждый раз, когда образ сохраняется, Smalltalk/V записывает в файл change.log комментарий, содержащий время и дату сохранения файла V.EXE.

- Теперь, когда запись найдена, известно что потерянная работа — от этой записи и до конца файла change.log. Не забудьте, что файл change.log форматируется как ряд порций. Точный формат приведён выше. Чтобы восстановить работу, надо выбрать одну или более последовательных порций текста (ограниченных восклицательными знаками '!'), а затем, выбрать опцию File It In из меню Smalltalk. Каждая порция — выражение, которое будет вычислено. Если порция — определение метода, он будет перетранслирован и установлен в систему. Выбирая порции, будьте очень внимательны, одна из них содержит ошибку, которая и привела систему к краху!

Когда восстановление завершено, используйте опцию Save Image из меню File и сохраните образ.

3.5 Словарь системы

В Smalltalk/V есть класс, известный как SystemDictionary. Он имеет только один экземпляр — словарь системы. Словарь системы определяет методы для операций, ориентированных на саму систему, типа сжатия файла регистрации изменений и определения доступной системе памяти. Кроме того, он содержит все глобальные имена, известные системе, то есть включает в себя имена классов, имена глобальных переменных и имена словарей пула. Следовательно, когда пользователем определяются новые классы, глобальные переменные и словари пулов, их имена автоматически добавляются в словарь системы. Напомним, что имена глобальных переменных должны начаться с заглавной буквы.

На словарь системы в Smalltalk-коде можно ссылаться по имени Smalltalk, которое является глобальной переменной. Именно к словарю системы мы обращались выше в выражениях, производящих сжатие файлов `change.log` и `sources.sml`

Ещё одним примером сообщения к словарю системы служит выражение **Smalltalk unusedMemory,** которое вычисляет число байтов памяти, доступное системе в настоящий момент.

Опишем кратко словарь системы и некоторые из системных команд (сообщений), которые могут ему посылаться. Для классов, представленных в словаре, ключ — имя класса, значение — сам класс. Для глобальных переменных, ключ — имя переменной, значение — объект. Если переменная была определена, но не инициализирована, её значение — **nil.** Для словарей пула, ключ — имя словаря, значение — сам словарь.

Вот некоторые из наиболее важных глобальных переменных, представленных в системе, наряду с их значениями и кратким описанием:

CharacterConstants — словарь пула, который связывает имена со специальными символьными значениями (ASCII-значениями) типа: carriage return (возврат каретки), line feed (перевод строки), escape (отказ), form feed (перевод страницы) и т.д.; ключи этого словаря — описательные строки, а значения — связанные со строкой ASCII-символы.

Clipboard — буфер обмена, позволяющий передавать данные между окнами и приложениями (обычно через копии), используя экземпляр класса ClipboardManager (АдминистраторБуфераОбмена), который и проводит операции вырезки, копирования и вставки.

CurrentProcess — процесс, который в настоящее время выполняется.

Cursor — курсор. Экземпляр класса CursorManager, который используется для того, чтобы хранить местоположение курсора.

Disk — экземпляр класса Directory (Каталог), представляющий текущий каталог, из которого была запущена система.

Display — единственный экземпляр класса **Screen** (**Экран**), позволяющий работать непосредственно с доступным системе экраном монитора, не обращаясь к окнам, используя связанное с экраном перо для вывода информации.

KeyboardSemaphore — экземпляр класса **Semaphore** (**Семафор**), который сообщает о появлении прерывания от мыши или клавиатуры.

Notifier — окно уведомлений, единственный экземпляр класса NotificationManager, который обрабатывает события, происходящие в окнах.

KeyboardLibrary — единственный экземпляр класса **KeyboardDLL,** используемый для того, чтобы создавать API-вызовы связанные с клавиатурой.

FloatLibrary — экземпляр **FloatEmulatorDLL** или **FloatCoprocessorDLL,** используемый для того, чтобы создавать API-вызовы для чисел с плавающей точкой.

WinEvents — массив, который обеспечивает отображение пронумерованных Windows-сообщений на селекторы системы **Smalltalk**.

GDILibrary — единственный экземпляр класса **GdiDLL,** используемый для создания API-вызовов к GDI.

KernelLibrary — единственный экземпляр класса **KernelDLL,** используемый для создания API-вызовов к ядру Windows.

UserLibrary — единственный экземпляр класса **UserDLL,** используемый для создания API-вызовов к информационным и вспомогательным функциям Windows.

Processor — процессор, единственный экземпляр класса **ProcessScheduler.**

TextFont — шрифт, используемый как шрифт по умолчанию для текстовых панелей.

ListFont — шрифт, используемый как шрифт по умолчанию для списковых панелей.

Smalltalk — единственный экземпляр класса SystemDictionary, словарь системы.

Sources — массив, содержащий два файловых потока для доступа к исходным текстам системы **Smalltalk/V**: файлам `sources.sml` и `change.log`.

SysFont — шрифт по умолчанию, используемый для отображения текстовой информации при текущем графическом разрешении.

Terminal — экземпляр класса **Pen** (**Перо**), связанный с объектом **Display**; используется для того, чтобы произвести звуковой сигнал (звонок), записывать текст или вывести графику на экран.

Transcript — экземпляр класса TextWindow, представляющий окно, используемое прежде всего для вывода сообщений системы; может использоваться пользователем для отображения нужной ему информацию (например, об отладке).

Существует много полезных методов, которые функционируют на словаре системы. Вот некоторые, наиболее часто используемые.

at: key put: anObject — создаёт новую глобальную переменную, с именем **key** и значением **anObject,** если глобальная переменная с таким именем уже существует, изменяет её значение на **anObject.**

compressChanges — сжимает файл `change.log`, удаляя из него все, кроме последних версий методов.

compressSources — сжимает файл `sources.sml` после его изменения в соответствии с последними версиями методов из файла `change.log`.

implementorsOf: aSymbol — возвращает окно **MethodBrowser** (окно просмотра Методов), содержащее все классы, которые определяют метод с именем aSymbol, что эквивалентно выбору функции Implementors из меню Methods окна просмотра Иерархии Классов.

keys — возвращает все ключи, определённые в настоящее время в словаре системы.

removeKey: key — удаляет ключ **key** из словаря системы.

sendersOf: aSymbol — возвращает окно **MethodBrowser** (окно просмотра Методов), содержащее все методы, и соответствующие им имена классов, которые посылают сообщение с именем **aSymbol,** что эквивалентно выбору функции **Senders** из меню **Methods** окна просмотра Иерархии Классов.

unusedMemory — возвращает число свободных байтов памяти, доступных системе.

Словарь, содержащийся в глобальной переменной может использоваться как словарь пула. Можно создавать новые словари пула, вычисляя выражения типа:

```
Smalltalk at: \#MyPool put: Dictionary new
```

После чего можно добавлять в словарь новые элементы (переменные пула), вычисляя выражения типа **MyPool at: 'Age' put: 29.**

В базовой среде **Smalltalk/V** есть несколько словарей пула. Имя одного из них (**CharacterConstants**) указано выше среди имен глобальных переменных. Приведём описание ещё нескольких, часто используемых словарей, с некоторыми из них мы уже встречались.

WinConstants — словарь пула, связывающий имена констант операционной системы Microsoft Windows с их значениями, которые доступны в среде **Smalltalk**.

ColorConstants — словарь пула, связывающий имя доступного цвета с его индексом в системной палитре или его RGB-представлением.

CursorConstants — словарь пула, связывающий имена курсоров с соответствующими экземплярами класса CursorManager.

DrawingModeConstants — словарь пула, связывающий имена графических режимов с их значениями.

FileConstants — словарь пула, связывающий имена атрибутов файла с их значениями.

VirtualKeyConstants — словарь пула, связывающий имена виртуальных ключей с их значениями.

Чтобы исследовать содержимое словаря пула, надо в какой-либо текстовой панели выбрать его имя, а затем выбрать опцию Inspect It (Проинспектировать) из меню Smalltalk. Можно поступить и по другому: в текстовой панели выбрать выражение, в котором словарю посылается сообщение inspect, а затем выбрать опцию Do It из меню Smalltalk. Например, вычисляя выражение CharacterConstants inspect, мы откроем окно инспектора словаря, из которого узнаем, что словарь пула CharacterConstants среди прочих содержит и такие элементы:

Ключ	ASCII-значение
Bell	Звонок (ASCII значение 7)
Bs	Забой (Backspace — ASCII значение 8)
Cr	Возврат каретки (ASCII значение 13)
Del	Удаление (ASCII значение 127)
Esc	Символ ESC (ASCII значение 27)
Ff	Перевод формы (Form Feed — ASCII значение 12)
FunctionPrefix	Первый символ функциональной клавиши из двух символьной последовательности (ASCII значение 0)
Lf	Перевод строки (Line Feed — ASCII значение 10)
MouseButton	Символ, посылаемый тогда, когда изменяется состояние любой кнопки мыши (ASCII значение 254)
SetLoc	Символ, посылаемый тогда, когда мышь перемещается (ASCII значение 255)
Space	Символ пробела (ASCII значение 32)
Tab	Символ горизонтальный табуляции (ASCII значение 9)
UpperToLower	Число из класса SmallInteger, равное 32, выражающее числовое различие между символами верхнего и нижнего регистров для английского алфавита.

Список литературы

[1] Goldberg A., Robson D., Smalltalk-80. *The language.* — Addison-Wesley Publishing Company, 1988.

[2] Буч Г. *Объектно-ориентированное проектирование с примерами применения.* — М.: Конкорд, 1992.

[3] *Смолток. Объектно-ориентированная система программирования. Руководство пользователя, часть 1, 2, 3* — М.: Ин-т проблем информатики РАН, 1995

[4] Иванов А., Кремер Ю., *Язык* Smalltalk: *концепция объектно-ориентированного программирования* // КомпьютерПресс — 1992, № 4 — С. 21–31

[5] Фути К., Судзуки Н., *Языки программирования и схемотехника СБИС*: пер. с япон. — М.: «Мир», 1988

[6] Кирютенко Ю.А., Савельев В.А., *Объектно-ориентированное программирование и язык* Smalltalk. *Общие концепции и синтаксис.* — Ростов-на-Дону: УПЛ РГУ, 1995

[7] Кирютенко Ю.А., Савельев В.А., *Объектно-ориентированное программирование и язык* Smalltalk. *Интерфейс пользователя и среда программирования.* — Ростов-на-Дону: УПЛ РГУ, 1995

[8] Кирютенко Ю.А., Савельев В.А., *Объектно-ориентированное программирование и язык* Smalltalk. *Протокол поддержки всех объектов системы.* — Ростов-на-Дону: УПЛ РГУ, 1997

[9] Кирютенко Ю.А., Савельев В.А., *Объектно-ориентированное программирование и язык* Smalltalk. *Протокол поддержки классов.* — Ростов-на-Дону: УПЛ РГУ, 1997

[10] Кирютенко Ю.А., Савельев В.А., *Объектно-ориентированное программирование и язык* Smalltalk. *Ядро графики: классы* Point, Rectangle, Form. — Ростов-на-Дону: УПЛ РГУ, 1997

[11] Кирютенко Ю.А., Савельев В.А., *Объектно-ориентированное программирование и язык* Smalltalk. *Ядро графики: класс* BitBlt. — Ростов-на-Дону: УПЛ РГУ, 1997

Содержание

1 Основные операции с окнами в среде Smalltalk/V **3**
1.1 Окна и панели 3
1.1.1 Прокручивание 5
1.1.2 Пользовательские меню 6
1.1.3 Управление окнами 7
1.2 Текстовый редактор системы 9
1.2.1 Вставка текста 10
1.2.2 Клавиши Enter, Tab, Backspace, Delete и клавиши со стрелками 10
1.2.3 Функции выбора, замены, удаления, копирования и вставки 10
1.2.4 Выполнение (Do It) и показ (Show It) 12
1.2.5 Сохранение (Save) и Восстановление (Restore) 13
1.2.6 Подсказчик (Prompter) 14

2 Стандартные окна системы Smalltalk **15**
2.1 Окно просмотра Иерархии Классов 15
2.1.1 Как открыть окно просмотра Иерархии Классов 16
2.1.2 Просмотр иерархии 17
2.1.3 Добавление нового класса 18
2.1.4 Удаление экземпляров класса 19
2.1.5 Удаление класса 19
2.1.6 Просмотр методов 20
2.1.7 Добавление, изменение и удаление метода 22
2.2 Окно просмотра Класса 23
2.3 Инспекторы 23
2.4 Окно просмотра Методов 25
2.5 Окно просмотра Сообщений (Селекторов) 26

3 Управление системой в целом **27**
3.1 Выход из системы и сохранение образа 27
3.2 Файл автоматической регистрации изменений 28
3.3 Выживание в случае аварийного отказа системы 30
3.4 Восстановление после аварийного отказа системы 31
3.5 Словарь системы 32