

Ю.А. Кирютенко , В.А. Савельев

Объектно-ориентированное программирование
и язык Smalltalk

Интерфейс пользователя и среда
программирования

Ростов-на-Дону

1997

Объектно-ориентированное программирование и язык Smalltalk

Интерфейс пользователя и среда программирования

Аннотация

Методическая разработка посвящена современному направлению в программировании — объектно-ориентированной методологии программирования и языку Smalltalk и является продолжением методической разработки «Объектно-ориентированное программирование и язык Smalltalk. Общие концепции и синтаксис». 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 5 декабря 1995 года.

Настоящие методические указания набраны в системе L^AT_EX с использованием кириллических шрифтов семейства LH (дизайнеры О. Лапко и А. Ходулев).

1 Введение

Первоначальной целью построения **Smalltalk**'а было создание системы, которая позволяла бы человеку естественным образом общаться с компьютером, постоянно подстраивая его под себя и свои задачи, которая развивалась бы вместе с опытом и потребностями пользователя. Для реализации этого проекта были использованы первые рабочие станции, созданные в 70-х на фирме Xerox, представлявшие собой высокопроизводительные однопользовательские миниЭВМ, снабженные растровыми дисплеями с отображением на оперативную память, впервые использовавшие **Ethernet**-сеть и устройство указания, получившее название “мышь”.

Для такой системы был непригоден обычный в 70-х текстовый интерфейс пользователя, обеспечиваемый с помощью различных языков программирования и языков управления заданиями. Поэтому в действительности **Smalltalk** — не только язык программирования, но объектно-ориентированная система, включающая в себя кроме небольшого ядра языка, также множество классов, поставляемых с системой, которые обеспечивают:

- широкий выбор структур данных (списков, очередей, массивов и т.д.) необходимых для эффективного программирования;
- объектно-ориентированный интерфейс с файловой системой рабочей станции или сети, и другими периферийными устройствами компьютера;
- графический интерфейс пользователя.

В целом, в отличие от компактного и простого языка, система **Smalltalk**, является большой и сложной программной системой (около 40000 строк кода; что сравнимо с 435000 строк в ОС UNIX BSD 4.2). При изучении системы **Smalltalk**, основные трудности связаны именно с освоением обширных системных классов, которые по своим возможностям существенно превосходят библиотеки классов, стандартно поставляемые с гибридными языками.

Наибольшее влияние на другие системы программирования и компьютерную индустрию в целом оказал именно графический интерфейс **Smalltalk**'а. Именно в нем были впервые реализованы и представлены общественности идеи, которые позже легли в основу XWindow, MacOS, OSFMotif, Solaris и MS Windows. Этот интерфейс был одним из первых (историки спорят, был ли он первым

или вторым) оконных интерфейсов, одним из первых событийно-ориентированных интерфейсов, первым полностью графическим интерфейсом и первым интерфейсом, использовавшим мышь.

2 Введение в интерфейс пользователя

Цели создания системы Smalltalk определили основные требования к интерфейсу пользователя. Интерфейс должен обеспечить возможность просмотра, использования и изменения всех объектов системы, в том числе и объектов реализующих сам интерфейс, в естественной форме. Взаимодействие системы с пользователем осуществляется через графический растровый экран, клавиатуру и устройство указания (обычно мышь).

Основным элементом пользовательского интерфейса системы является *рамка* (*вид*, *окно*). Рамкой называется прямоугольная область экрана системы, внутри которой размещается некоторое изображение, визуальное представляющее объект (или совокупность объектов). Рамка, снабженная заголовком, называется окном. Внутри окно может содержать другие рамки (называемые подрамками). Окна не могут содержать другие окна. Хотя в системе при-

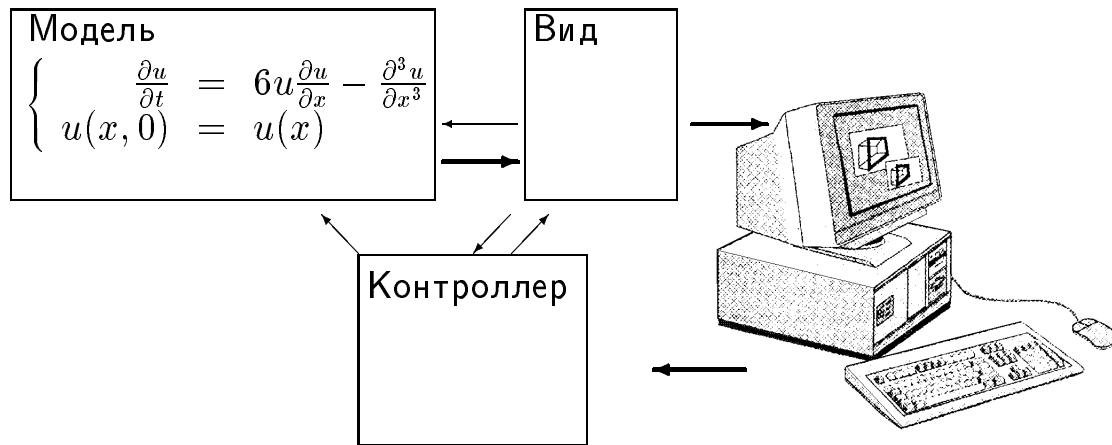


Рис. 1. Триада MVC

существует predetermined набор рамок, обеспечивающих единообразное представление объектов системы и среду программирования, для прикладных программ необходимо создавать свои разновидности рамок, которые бы обеспечивали визуальное представление специфических для данной программы объектов. Все программы в системе Smalltalk (как сама система, так и все коррект-

ные интерактивные программы) построены на основе триады MVC (Model-View-Controller/Модель-Рамка-Диспетчер, см. Рис 1), где единственным видимым элементом являются рамки (View), которые знают *как* отображать объекты (Model) на экране, и изображающие их по командам диспетчера (Controller). В качестве модели может выступать любой объект системы, поэтому нет каких-то специальных требований к модели. В качестве же рамки и диспетчера используются объекты, являющиеся экземплярами подклассов классов Рамка и Диспетчер соответственно. Задача модели — хранить и обрабатывать содержательные данные прикладной программы. Задача рамки — обращаясь за данными к модели, создать на экране ее визуальное представление. Задача диспетчера — поддерживать протокол взаимодействия пользователя с приложением, управляя вычислениями в модели и отображением данных рамкой.

За каждой рамкой (окном), используемым, в системе стоит триада MVC. Многие окна (см. Рис. 2) состоят из нескольких рамок, обладающих различными свойствами, которые позволяют манипулировать различными аспектами модели. В таких случаях каждая подрамка тоже представляет триаду MVC.

Более подробно вопросы связанные с триадой MVC будут освещены в методических указаниях, посвященных построению интерактивных приложений в системе Smalltalk.

Стандартно в состав пользовательского интерфейса системы входят следующие рамки

- Рабочее окно
- Системная информация
- Суфлер
- Просмотр диска
- Просмотр иерархии классов
- Инспектор класса
- Инспектор
- Окно диагностики
- Отладчик

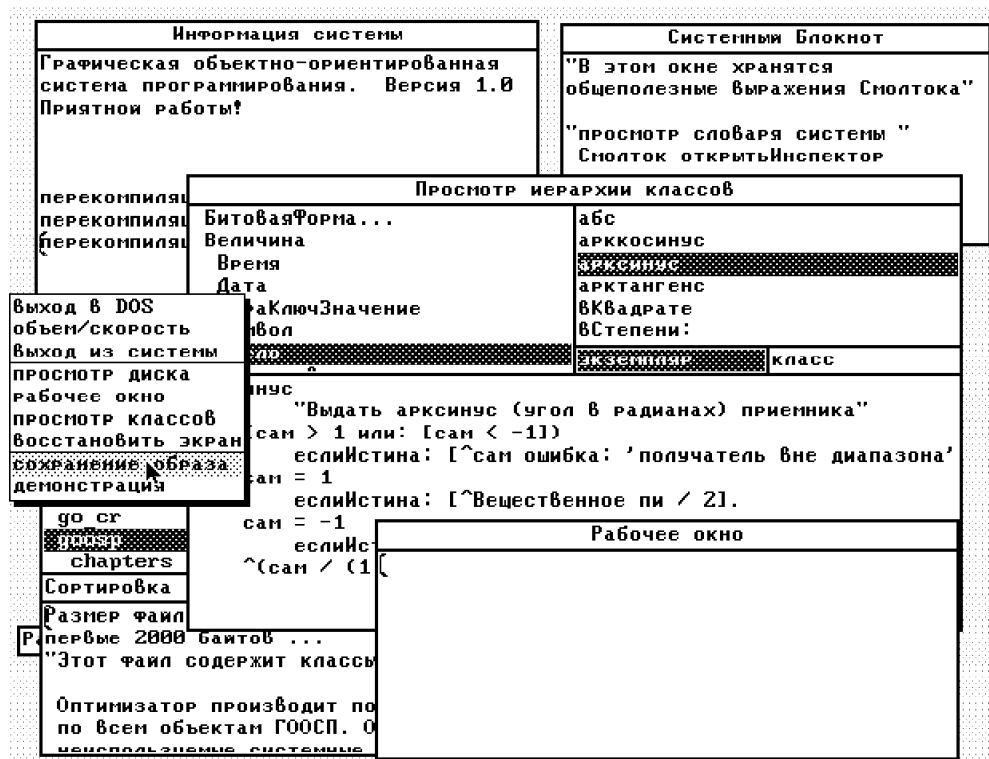


Рис. 2. Вид дисплея русифицированной системы Smalltalk

Некоторые из этих окон вы можете увидеть на Рис. 2, где изображен момент запуска демонстрационного примера путем вычисления выражения в рабочем окне в нижней правой части экрана.

Стрелка на экране — это курсор устройства указания. Обычно он управляется с помощью мыши, но в реализациях Smalltalk'а на IBM-совместимых компьютерах имеется возможность выполнять все операций без использования мыши (См. Рис. 3): для перемещения курсора устройства указания можно также использовать клавиши со стрелками, расположенные на клавиатуре.

Всюду далее мы не будем делать различия между действием с помощью мыши и действием с помощью клавиатуры, и после однократного описания альтернативного способа действий, будем говорить только о действиях мышью (заметим, что на системах не совместимых с IBM PC, все действия выполняются только с помощью мыши). Отметим также, что двойные щелчки в системе Smalltalk не используются.

Курсор позволяет выбирать различные объекты и их части на экране. Для этого необходимо щелкнуть левой клавишей мыши,

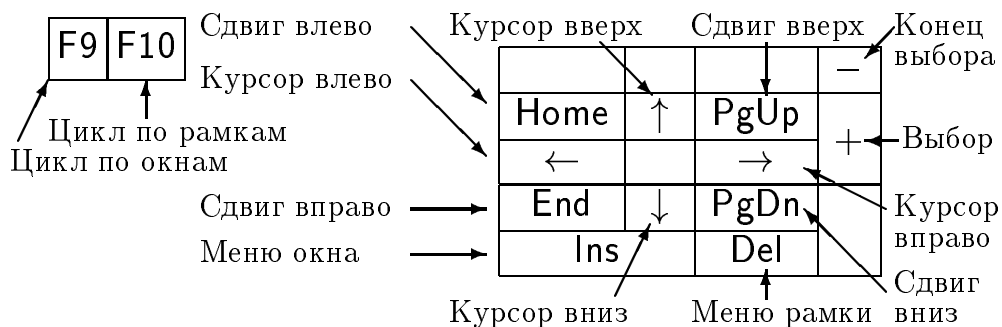


Рис. 3. Клавиши, используемые в реализациях Smalltalk для IBM PC

когда курсор указывает на этот объект. Так, для того чтобы какое-либо окно сделать активным, необходимо щелкнуть левой клавишей мыши когда курсор находится над этим окном. Активное окно изображается всегда на первом плане, его заголовок выделен цветом, и именно в него направляется ввод с клавиатуры. Альтернативой щелчку мыши является нажатие серой клавиши **+** на дополнительной клавиатуре. Кроме того, возможно циклически перебирать окна, нажимая клавишу **F9**.

Если щелчок мыши произошел вне какого-либо из окон все окна на экране становятся неактивными. Активной при этом становится рабочая поверхность экрана. Если после этого нажать на правую кнопку мыши или на клавиши **Ins** или **Del**, то на экране появится системное меню (См. Рис 2). Каждый выбор в этом меню занимает отдельную строку, текущий выбор выделяется цветом. Чтобы выбрать какой-либо пункт из меню необходимо переместить курсор внутри меню так, чтобы подсвеченной оказалась нужная строка после чего нажать правую кнопку мыши. Чтобы выйти из меню не делая выбора, надо просто вывести курсор за пределы меню; при этом меню исчезнет с экрана. Системное меню содержит следующие команды:

выход в DOS временно выйти из системы Smalltalk и запустить командный процессор DOS. Чтобы вернуться в Smalltalk, необходимо в командной строке DOS набрать команду **exit**.

память/скорость изменение режима работы пользовательского интерфейса. Текст в пункте меню отражает текущий режим

системы; по умолчанию это экономия памяти за счет некоторого снижения быстродействия.

выход из системы выйти из системы Smalltalk обратно в операционную систему. При этом система выводит подменю позволяющее отказаться от выхода в DOS, выйти без сохранения образа или выйти с сохранением образа (см. далее).

просмотр диска создать окно просмотра диска.

рабочее окно создать рабочее окно — простейший текстовый редактор с возможностью вычисления выражений Smalltalk'a.

просмотр классов создать окно просмотра иерархии классов, присутствующих в системе Smalltalk.

восстановить экран принудительно перерисовать экран. Может быть чрезвычайно полезно при отладке приложений.

сохранить образ сохранить бинарный образ системы, содержащий все объекты системы Smalltalk; при этом будут запомнены все установки, окна и их содержимое на момент сохранения образа. Если образ не сохранен, все изменения сделанные в системе с момента последнего сохранения образа, будут отменены при выходе из системы.

демонстрация запустить приложение, демонстрирующее возможности графических классов Smalltalk'a.

С каждым окном в Smalltalk'e связано меню окна, содержащее действия, общие для всех окон системы. Это выбор цвета, перемещение окна, изменение его размера, сокрытие и удаление окна, переход к следующему окну. Некоторые особые окна имеют не все из этих возможностей: например, окно системной информации не имеет в меню пункта "удалить". Для вызова этого меню необходимо щелкнуть правой кнопкой на заголовке активного окна или нажать клавишу Ins, когда курсор находится над окном. Работают с этим меню точно также как и с системным. При щелчке правой кнопкой мыши внутри окна или при нажатии клавиши Del, когда курсор находится над окном, вызывается меню рамки. Так как окно может содержать несколько различных рамок, то каждая из них имеет свое меню рамки или не имеет его вовсе. Меню рамки содержит функции, специфичные для данного типа рамки, а также

определяемые интерактивным приложением, создавшим эту рамку. Примеры меню рамки будут приведены позднее.

В большинстве Smalltalk-систем применяется мышь с тремя кнопками и вызов меню окна и меню рамки закреплен за правой и средней ее кнопками. При реализации Smalltalk на IBM PC/XT/AT разработчики учли, что на этих машинах подавляющее большинство составляют двухкнопочные мыши, в связи с чем и был выбран вариант с пространственным разделением зон выбора меню окна и меню рамки.

Большинство рамок могут служить для представления большего количества информации, чем можно показать сразу на экране. Для отображения информации многие типы рамок поддерживают прокрутку информации внутри рамки; т.е. если визуальное представление объекта не может быть полностью показано, показывается только его часть и рамка предоставляет возможность перемещать визуальное представление объекта под рамкой выполняя прокрутку. Для выполнения прокрутки с помощью мыши надо нажать правую кнопку мыши (при этом курсор изменится, приняв вид четырех стрелок), и по правой границе окна появится *полоса прокрутки* (или, в терминологии Smalltalk'а, *лифт*), являющаяся прямоугольником, представляющим весь объект, на котором контрастным цветом выделена видимая часть объекта. Если удерживая правую клавишу нажатой, вывести курсор за пределы активной рамки, он примет вид стрелки (см. рис. 10), указывающей направление прокрутки объекта и начнется прокрутка объекта. Чтобы остановить прокрутку достаточно отпустить правую кнопку или вернуть курсор в рамку. Для использования полосы прокрутки необходимо ввести курсор на ее территорию (при этом он примет вид перекрестия, см. рис. 8) и отпустить правую кнопку над частью прямоугольника, соответствующей желаемой позиции над рассматриваемым объектом. Использование прокрутки требует определенной тренировки и может сразу не получиться.

В IBM-совместимых машинах для прокрутки можно использовать и клавиатуру: клавиша **Home** обеспечивает прокрутку влево, **End** — прокрутку вправо, **PgUp** — вверх, и **PgDn** — вниз.

Рамки позволяющие использовать прокрутку имеют в качестве контроллера (диспетчера) экземпляр одного из подклассов класса ДиспетчерСдвига. В частности, самые часто используемые рамки, текстовые и списковые поддерживают прокрутку.

2.1 Рабочее окно

Начнем с простого окна редактора текста. Обычно такое окно имеет заголовок “Рабочее окно”. Его можно получить любым из следующих двух способов:

- Выбрать в системном меню “рабочее окно”. Вам будет предложено задать размер этого окна указав сначала верхний левый, а затем нижний правый углы с помощью курсора и клавиши выбора (напомним, что это левая клавиша мыши или серый плюс). Вы получите редактор по умолчанию без какого-либо текста в окне.
- Послать любому экземпляру класса **Строка** сообщение редактировать. При этом после уже описанной процедуры запроса положения и размера окна на экране появится окно, содержащее ту строку, которой было послано сообщение редактировать.

Окно текстового редактора состоит из единственной рамки, являющейся экземпляром класса **ТекстоваяРамка**, управляемой диспетчером (контроллером) из класса **РедакторТекста**. Моделью здесь, очевидно, является экземпляр класса **Строка**. Кроме отдельных окон-редакторов текстовая рамка активно используется как часть многих других окон, поэтому опишем ее подробнее.

В каждой текстовой рамке присутствует текстовый курсор, он может иметь вид вертикальной черты с усиками на концах (каретки), либо инвертированной области в текстовой рамке. Каретка показывает место в редактируемой строке, в которое будут вставляться набираемые на клавиатуре символы. Если нажать клавишу **Backspace** будет удален символ перед кареткой. Нажатие на клавишу **Enter** приведет к вставке новой строки, курсор переместится в начало новой строки вместе со всеми текстом, который находился правее каретки. Клавиша пробел вставляет в строку четыре пробела. Если текстовый курсор имеет вид инвертированной области в текстовой рамке (называемой также выбором) то вводимый текст будет замещать выделенный фрагмент, нажатие клавиши **Backspace** удалит весь выделенный фрагмент, и над ним выполняются все операции из меню рамки. Строго говоря, каретка

— это тоже выбор, правда пустой, нулевой длины, который имеет позицию в строке, но не имеет содержимого.

Перемещение каретки осуществляется щелчком мышью над соответствующим местом текстовой рамки. Для создания непустого выбора необходимо нажать левую клавишу мыши над позицией начала выбора и удерживая эту клавишу нажатой протянуть курсор мыши до позиции конца выбора. Если выбор не уместается в рамке можно либо удерживая левую клавишу вывести курсор мыши за пределы рамки так, чтобы началась прокрутка (при этом прокручиваемый текст будет сразу попадать в выбор), либо воспользоваться клавиатурой: клавишей поставить каретку в начало выбора и любым способом переместить курсор мыши в конец выбора и нажать клавишу расширенного выбора .

Меню текстовой рамки содержит следующие операции:

восстановить — вернуться к предыдущему запомненному состоянию редактора, см. **сохранить**.

скопировать — перенести текущий выбор в общий буфер всех текстовых рамок в системе, в активной текстовой рамке все оставить как было.

вырезать — перенести текущий выбор в общий буфер текстовых рамок, и удалить его в активной текстовой рамке.

вставить — вставить в позицию каретки содержимое общего буфера текстовых рамок, если текущий выбор был непуст, содержимое буфера замещает его.

показать — считая текущий выбор выражением, вычислить его. Результат вычисления преобразовать в строку и вставить в текущее окно сразу после текущего выбора. Сделать вычисленный результат новым текущим выбором. Если выражение неправильно, выдать окно диагностики.

выполнить — считая текущий выбор выражением, вычислить его. Результат игнорировать, в текущей рамке редактор ничего не меняет. Если выражение содержит ошибку, показать окно диагностики.

сохранить — сохранить содержимое текстовой рамки, т.е. ее текст, для восстановления при необходимости. кроме того,

если вы открывали текстовую рамку для редактирования существующей строки, то эта операция сохранит в ней изменения.

следующее меню — перейти к дополнительному меню текстовой рамки. Оно содержит команды, приведенный далее.

печать — печатает на печатном устройстве по умолчанию текст (экз Строки) редактируемый в данной рамке.

Кроме того, включены операции контекстного поиска и замены, они сами запрашивают необходимую для работы информацию:

поиск

поиск назад

заменить все

повторить

Таким образом рабочее окно является очень важным элементом среды Smalltalk'a, позволяющим не только редактировать текст, но и производить вычисления.

Так как окна сохраняют свое содержимое между сеансами работы (если был сохранен образ), то можно использовать их и как записные книжки и сборники полезных выражений. Одно из них делают глобальным и называют Системный Блокнот (System Workspace). В него принято заносить общепользные выражения, необходимые для запуска различных приложений Smalltalk'a, имеющих в системе, а также общую справочную информацию.

2.2 Системная информация

При работе в Smalltalk'е всегда где-то на экране присутствует окно системной информации. Это тоже текстовый редактор, правда с той особенностью, что его нельзя удалить из системы (точнее, нельзя просто удалить из системы, но удалять его и не нужно). По соглашению, это окно используется для вывода необходимых сообщений о происходящем в системе (т.е. как системную консоль). Его лучше не использовать в качестве обычного текстового редактора. При старте системы в нем появляется ее приветствие. В процессе работы многие системные классы выводят в него свои сообщения

и предупреждения. Вы тоже можете воспользоваться этим окном. Оно доступно всем как глобальный объект с именем **СистемнаяИнформация**. Например, сообщение о завершении записи файла данных, может быть оформлено следующим образом:

СистемнаяИнформация символ**ВК**;
показать: 'Вывод файла ', имя**Файла**, ' завершен.'.

сообщение **символВК** обеспечивает продолжение вывода с новой строки, а сообщение **показать**: выводит свой аргумент в окно системной информации.

Окно системной информации необходимо использовать, когда необходимо вывести сообщение, не являющееся сообщением об ошибке (об этом случае смотри окно диагностики), и нет необходимости (возможности, желания) организовывать для этого отдельное окно.

2.3 Суфлер

Суфлер — окно очень специального вида. Это одностроковый текстовый редактор с рядом специальных свойств. Так, когда он активен вы не можете выбрать другое окно. Сначала необходимо завершить работу с суфлером. Работа с суфлером завершается либо нажатием на клавишу **Enter**, что подтверждает ввод текста, набранного в суфлере, или выбором из меню рамки команды “отменить”, что по стандартному соглашению означает отказ от продолжения действий, для которых запрашивал информацию суфлер.

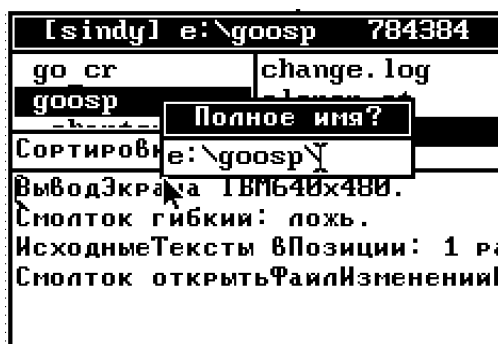


Рис. 4. Суфлер над окном просмотра диска запрашивает имя нового файла

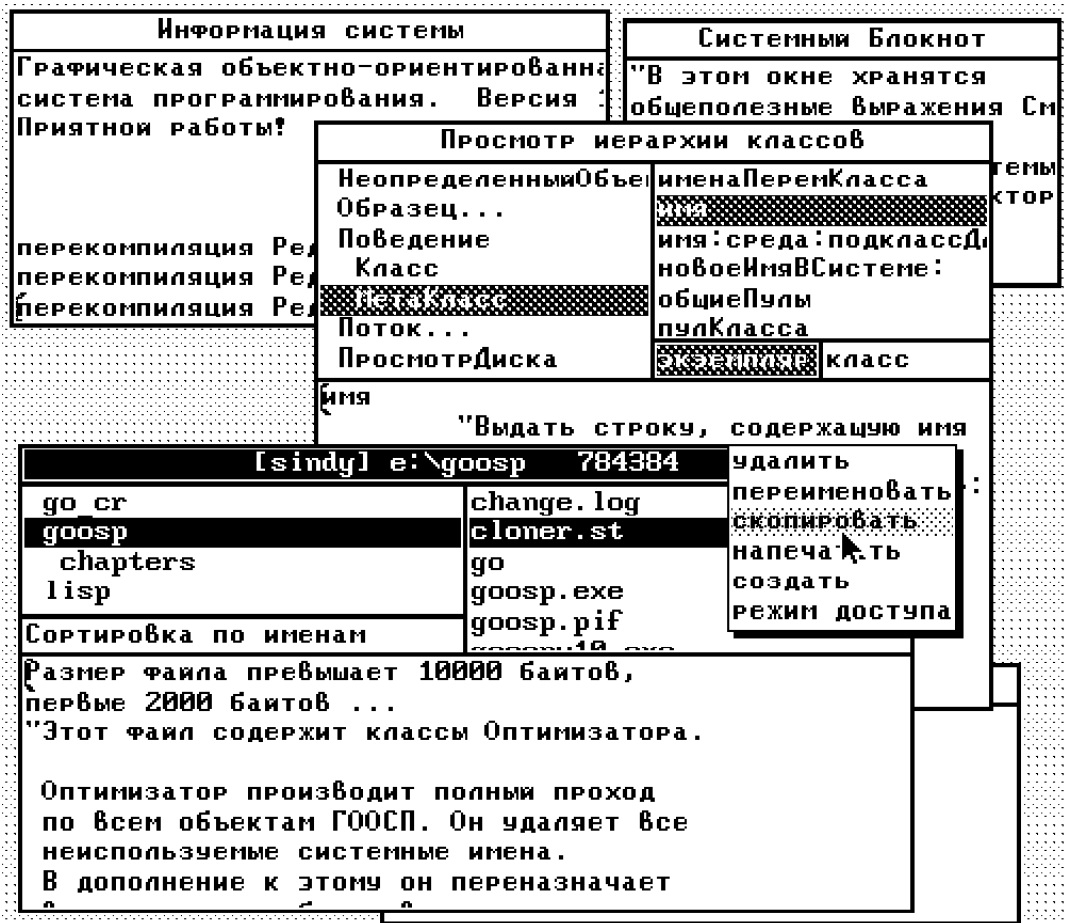


Рис. 5. Окно просмотра диска

Чтобы применить суфлер надо в программе набрать что-то вроде

ответ := Суфлер

подсказка: 'введите имя' по умолчанию: ''.

Это выражение приведет к появлению на экране суфлера, который будет ждать пока вы не завершите ввод строки нажатием на клавишу **Enter**. Возвратит он строку, видимую в нем, или нуль, если ввод был отменен командой меню рамки.

2.4 Просмотр диска

Просмотр диска — это сложное окно, состоящее из нескольких рамок. Его задача — предоставить удобное средство как для взаимодействия с файловой системой машины, так и для редактирова-

ния текстовых файлов. Окно просмотра диска (см. Рис. 5) разбито на четыре части:

- дерева каталогов,
- списка файлов,
- сортировки,
- редактора файлов.

Заголовок окна просмотра файлов зависит от операционной системы в которой работает система Smalltalk. В UNIX-системах заголовок просто сообщает, что это окно просмотра диска (Disk Browser). Системы на основе IBM PC обычно содержат в заголовке букву диска, имя тома и некоторую дополнительную информацию, зависящую от реализации, например, количество свободного места на диске.

Рамки дерева каталогов и списка файлов являются типичными примерами еще одного стандартного вида рамок — списковых. Они, как ясно из названия, служат для отображения списков и работы с ними. Списки могут быть больше чем видно в рамке, так как списковые рамки поддерживают прокрутку. Соответствующим образом организованные списки позволяют представлять практически любую информацию. В отличие от текстовых рамок, списковые рамки не имеют стандартных команд меню рамки, которые полностью определяются приложением.

Рамка каталогов служит для отображения дерева каталогов на диске. Вложенность каталогов обозначается размером отступа. Пока не один каталог не выбран, рамка списка файлов и рамка редактора пусты. После выбора какого-либо каталога в рамке списка файлов отображаются имена файлов (но не подкаталогов) из этого каталога. Эта же информация в более подробной форме (размер файла, время изменения, атрибуты доступа) показывается первоначально и в рамке редактора.

Меню рамки содержит основные операции с каталогами:

создать новый каталог, как подкаталог текущего; имя нового каталога запрашивает суфлер.

обновить дерево каталогов в текущем окне просмотра; полезно при работе в многопрограммной/многопользовательской среде (сеть, UNIX, etc.)

удалить текущий каталог; каталог должен быть пуст.

В нижней части рамки каталогов расположена еще одна рамка для отображения способа сортировки имен файлов в рамке списка файлов. Изменение способа сортировки осуществляется путем вызова меню этой рамки.

Операции с файлами доступны через рамку списка файлов. В этой рамке можно выбрать файл, который будет показан в окне редактора (файлы, размером более 10К, по умолчанию отображаются не полностью). Кроме этого поддерживаются следующие операции над файлами:

удалить

переименовать — Smalltalk запросит с помощью суфлера новое имя файла.

скопировать — Smalltalk запросит с помощью суфлера куда копировать файл.

напечатать

создать пустой файл

режим доступа — изменение атрибутов доступа файла.

Рамка редактора файлов, в сущности, является обычной текстовой рамкой, и, если не выбран файл и/или каталог, она пуста или содержит дополнительную информацию к рамке списка файлов, а ее меню отличается от меню обычной текстовой рамки только командой "сохранить в", которая позволяет сохранить содержимое рамки в файле, имя которого запрашивает суфлер. Если же выбран файл, то в рамке появляется его содержимое, а меню этой рамки содержит, кроме стандартных команд текстовой рамки, следующие дополнительные команды:

установить — взять текущий выбор в рамке редактора и после компиляции включить в систему.

прочитать — если файл имеет длину более 10К, система не читает его полностью, а выводит в рамке редактора соответствующее сообщение за которым следуют первые 2К этого файла. Данная команда позволяет прочесть и редактировать весь длинный файл.

Если Вы отредактировали файл, то в меню появятся еще команды: **сохранить** в изменения в том же файле, переписав старое содержимое.

сохранить в новом файле, имя которого будет запрошено с помощью суфлера.

При выборе другого файла/каталога в соответствующих рамках или при попытке закрыть окно без сохранения отредактированного файла появится специальное меню позволяющее выбрать отмену сделанных изменений, или продолжение редактирования.

Такая организация сложных окон, когда содержимое и функциональность рамки зависит от того, сделан ли какой-либо выбор в предыдущей рамке, типична для системы **Smalltalk**, и используется в описываемых далее инструментах программиста.

3 Программирование в системе **Smalltalk**

Программирование в системе **Smalltalk** опирается на широкое использование представления классов и их иерархии с помощью окна просмотра иерархии классов и окна инспектора классов, а работа с произвольными объектами осуществляется с помощью инспектора объектов или специализированных редакторов. Для работы с исполняющимся кодом методов **Smalltalk**'а используются окна диагностики и отладчики.

3.1 Окно просмотра иерархии классов

Для того, чтобы получить доступ к окну просмотра иерархии классов, необходимо выбрать в системном меню команду "просмотр классов". Система создаст окно просмотра иерархии классов (см. Рис. 6), состоящее из списка классов, списка селекторов методов, переключателя "методы класса–методы экземпляра", и редактора определений. Окно списка классов содержит имена классов, иерархически упорядоченных с помощью отступов. В большинстве систем, из-за большого числа классов, первоначально отображают только имена классов верхнего уровня (класс **Объект** и его непосредственные подклассы) и обеспечивают средства для локального углубления видимого дерева классов. В **Smalltalk**'е для рабочих

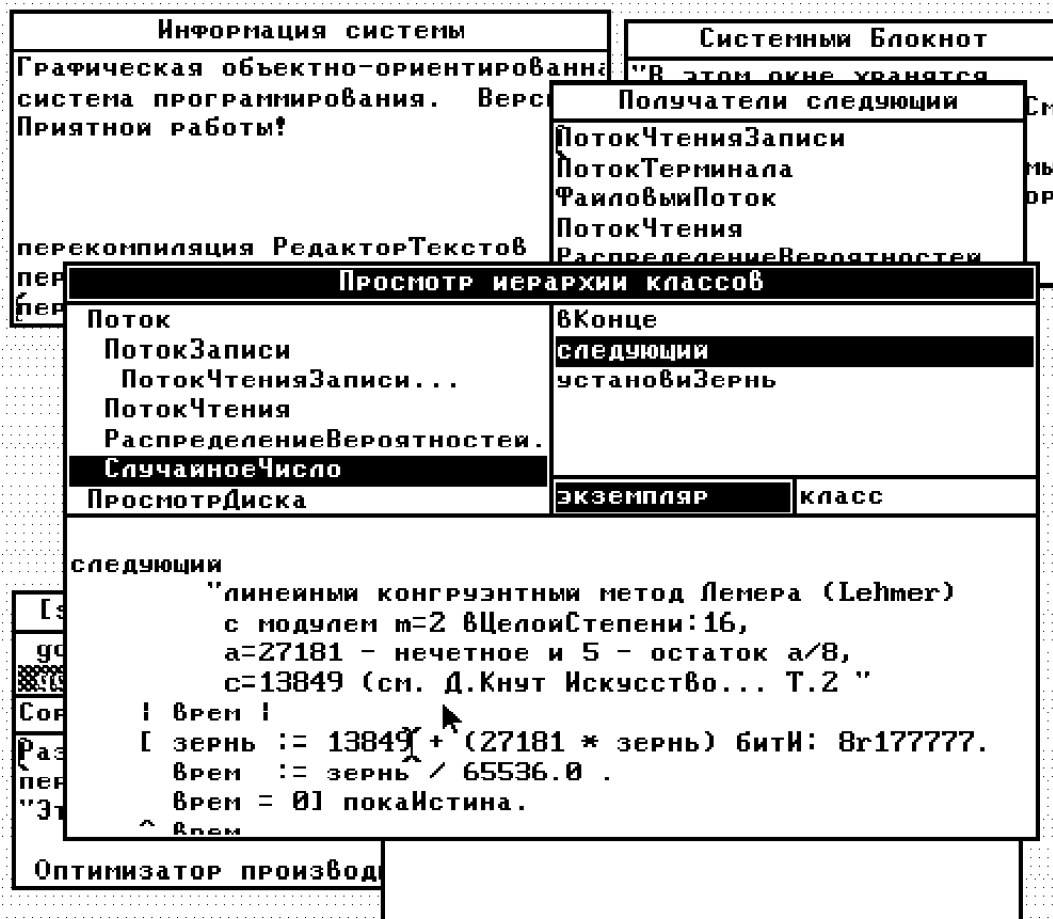


Рис. 6. Окно просмотра иерархии классов

станций, дополнительно применяются категории классов/методов, что позволяет просматривать только классы/методы имеющие отношение к определенной теме (например, классы графики, или методы инициализации), но в системах для IBM PC/AT, где классов существенно меньше, механизм категорий как правило не поддерживается.

После появления окна просмотра иерархии классов в нем отображаются только имена классов верхнего уровня, а рамки селекторов и редактор пусты. Просмотр подклассов текущего класса вызывается командой "скрыть/показать" из меню рамки; при этом иерархия классов обычно отображается с помощью отступов, хотя встречаются системы, рисующие графическое дерево классов. Выбор класса вызывает появление в рамке методов списка селекторов методов экземпляра/класса для выбранного класса (выбор отображения методов либо экземпляра, либо класса, выбирается с помо-

щью переключателя экземпляр/класс, расположенного в нижней части рамки селекторов методов), а в рамке текстового редактора появляется определяющее выражение класса, которое вводит в систему имя класса, задает вид класса, и определяет имена переменных экземпляра, переменных класса и разделяемые пулы. Операции с классами выполняются с помощью меню рамки, содержащего следующие операции:

вывод в файл с расширением .cls и именем, образованным из имени текущего класса, последовательности выражений, определяющих класс и его методы, в формате файла изменений системы.

обновить — обратиться к системе для показа изменений, произведенных в классах вне данного окна просмотра.

скрыть/показать дерево подклассов для текущего класса.

просмотр открыть окно редактирования для текущего класса

подкласс — определяет новый класс как подкласс текущего (выбранного) класса, запрашивает имя нового класса и его тип (см. [8], Разд. 2.2):

- подкласс
- подкласс с индексированными переменными,
- подкласс с байтами,
- подкласс со словами.

После ответа на эти вопросы в редакторе появится определяющее выражение нового класса. Его можно редактировать, добавляя переменные экземпляра, переменные класса и разделяемые пулы в определение класса.

Рамка селекторов отображает список селекторов методов экземпляров текущего класса или методов самого класса в зависимости от состояния переключателя экземпляр/класс в нижней части рамки селекторов. Пока селектор метода не выбран (это бывает если еще не выбирался селектор или после переключения экземпляр/класс), редактор, как указано выше, показывает определяющее выражение класса. Если выбран селектор, то в редакторе появляется текст соответствующего метода, предваряемых шаблоном сообщения, вызывающего метод. Меню рамки селекторов выполняет позволяет вызывать следующие функции:

удалить метод с выбранным селектором

новый метод — вывести в редактор шаблон метода.

имяСообщения

”комментарий”

| временные переменные |

операторы

Если после его редактирования выбрать в меню редактора пункт сохранить, то селектор нового метода появится в списке методов, а текст метода будет откомпилирован и помещен в исходные тексты системы, и откомпилированный метод будет включен в словарь класса.

отправители — какие методы в каких классах посылают сообщение с выбранным селектором. Появляется текстовое окно содержащее список методов в формате:

ИмяКласса

селекторМетода.

получатели — те классы в системе которые понимают сообщение с этим селектором. Появляется окно со списком классов, имеющих метод с тем же селектором, что и текущий. На рис. 6 это окно находится выше и правее окна просмотра иерархии классов.

Редактор окна просмотра иерархии классов обычен для Small-talk’a, за тем исключением, что операция сохранения отредактированной информации вызывает компилятор. Поэтому, если вносились изменения в классы системы, они не будут действовать, пока не сохранены. Если внесенные изменения затрагивают многие методы (например, добавление новых переменных в класс, имеющий уже много методов и/или подклассы) система сама перекомпилирует все затронутые изменением методы, выводя сообщение об этом в окно системной информации.

ИндексированныйНабор: Просмотр класса	
класс	экзНабор
экземпляр	"Видать значение истина, если элемент равен элементу, содержащемуся в
,	
=	! индекс !
ВОбратном	сам == экзНабор
Включает:	еслиИстина: [^истина].
ВоВсеПозиции	(сам класс == экзНабор класс)
ВоВсеПозиции	еслиЛожь: [^ложь].
Выполнить:	индекс := сам размер.
ВыполнитьЕ	индекс ~ экзНабор размер
доступКПус	еслиИстина: [^ложь].
заменитьОт	[индекс <= 0]
заменитьОт	покаЛожь: [
заменитьОт	(сам ВПозиции: индекс) = (экз
индексОбъек	еслиЛожь: [^ложь].
индексОбъек	индекс := индекс - 1].
найтиПервый	^истина

Рис. 7. Инспектор класса

3.2 Инспектор класса

Другой способ редактировать определения методов системы — использовать инспектор класса. Для его вызова надо послать сообщение редактировать классу, например:

Время редактировать.

На экране появится окно, состоящее из переключателя экземпляр/класс, списка селекторов методов и редактора. Его заголовок состоит из имени класса, за которым, после вертикальной черты, указано 'Просмотр класса'. Это окно не позволяет, в отличие от окна просмотра иерархии классов, редактировать определяющее выражение класса, т.е. структуру класса и его экземпляров, но его возможности редактирования методов указанного класса полностью аналогичны окну просмотра иерархии методов.

3.3 Редакторы и инспекторы

Изучение и изменение объектов, существующих в системе, осуществляется с помощью инспекторов и специальных редакторов, использующих человеко-ориентированные представления объектов.

Специальные редакторы вызываются посылкой сообщения ре-

дактировать объекту; ответом на это сообщение является создание окна редактора для редактирования объекта, которому послано сообщение. Специальные редакторы определены не для всех объектов системы. На сообщение **редактировать** могут отвечать экземпляры классов **Строка** и **Класс**. Редактором для строки является обычное рабочее окно, а для класса — окно просмотра класса. Оба эти окна описаны выше.

Для других объектов специальных редакторов по умолчанию нет и посланное им сообщение **редактировать** вызывает ошибку. Если в систему добавлено приложение, способное редактировать объекты некоторого класса, то его нетрудно превратить в специальный редактор: надо просто в этом классе определить метод с селектором редактировать. Например, если в системе установлен графический редактор, в класс **Форма** можно добавить следующий метод:

редактировать

"Открывает графический редактор для приемника"

ГрафическийРедактор открытьНа: сам

В отличие от специальных редакторов, инспекторы определены для любого объекта системы, но они предоставляют доступ ко внутренней организации объекта, что обеспечивает неограниченные возможности изучения и изменения системы, хотя и способны обескуражить начинающего, показав при попытке просмотреть картинку служебную информацию и ссылку на большой массив, вместо изображения.

Вызываются инспекторы посылкой объекту сообщения **открытьИнспектор**. При этом на экране появляется окно из двух рамок: списковой, отображающей названия частей объекта, и текстовой рамки, содержащей текстовое представление этой части. Если представление выбранного в списке объекта непонятно, для него можно вызвать отдельный инспектор с помощью команды меню списковой рамки. В текстовой рамке можно набрать любое выражение **Smalltalk'a** и по команде меню рамки сохранить оно будет вычислено и его значение заменит текущее значение. Такое представление легко изучать только в случае словарей, когда список содержит ключи доступа к словарю, а текстовая рамка — представление ассоциированных с ключом значений.

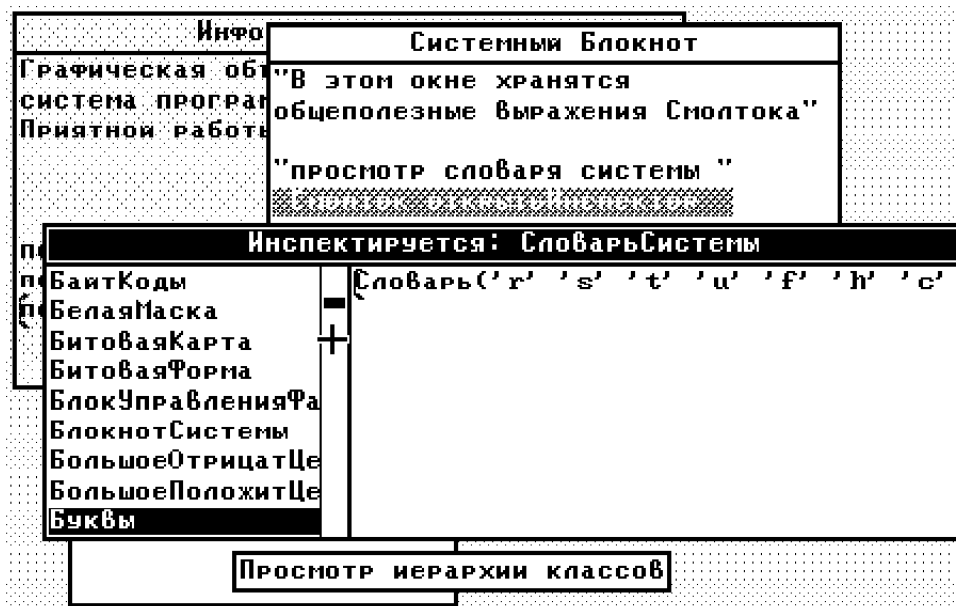


Рис. 8. Инспектор словаря

Инспекторы использовать нелегко, но это очень мощное средство, которое требует осторожности. Если Вы не представляете, к чему приведет изменение значения в текстовой рамке, его лучше не производить.

3.4 Средства отладки

В системе Smalltalk отладка осуществляется с помощью двух специальных рамок: окна диагностики и отладчика. Эти окна дают представление о состоянии программы в момент ошибки (останова, прерывания).

Отладчик является чрезвычайно мощным средством исследования и редактирования системы. Он вызывается в случае необходимости из окна диагностики. Окно диагностики предоставляет пользователю информацию о происшедшей ошибке или другом событии, приводящем к активации окна диагностики, и информацию о последовательности вычислений, выполнявшихся в текущем процессе на момент появления окна диагностики.

Заголовок окна диагностики содержит описание причины появления окна диагностики. Текстовая рамка окна диагностики содержит трассу выполнения прерванного метода, т.е. последовательность вложенных контекстов. Они отображаются как список вло-

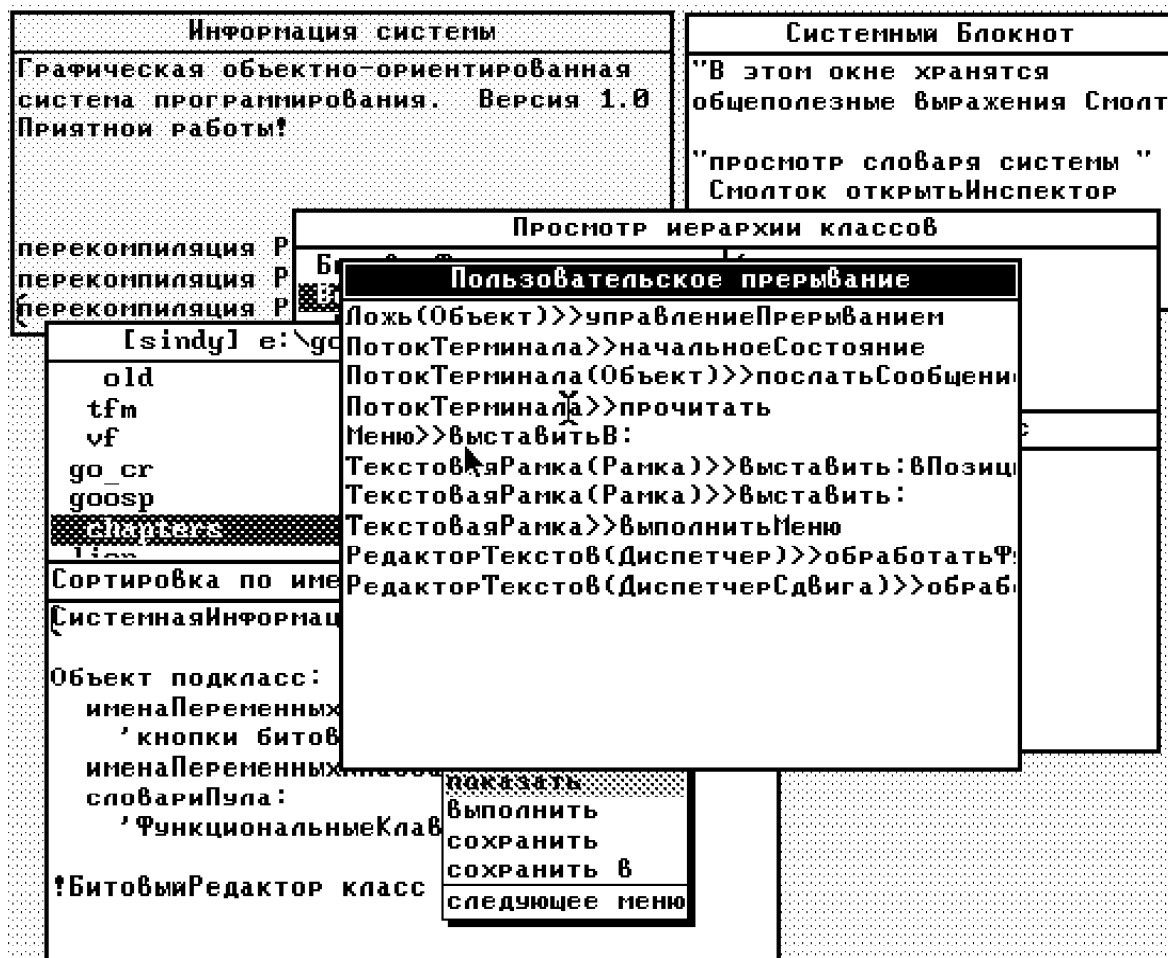


Рис. 9. Окно диагностики

женных вызовов методов и вычислений блоков, описанных в формате:

ИмяКласса>> селекторМетода

если метод с указанным селектором был найден в классе, экземпляру которого послано сообщение, или в формате:

ИмяКласса(ИмяКлассаПолучателя)>>селекторМетода

если метод был найден в классе-получателе, а не в классе, экземпляру которого послано сообщение, или в формате:

[] в *ИмяКласса*>>*селекторМетода*

если внутри метода, указанного в предыдущей строке, вычислялся блок.

Последний контекст находится вверху списка, а контекст, с которого началось вычисление рассматриваемого процесса, находится внизу списка.

Окно диагностики появляется на экране в одном из следующих случаев:

- любому объекту послано сообщение **ошибка:**. Аргументом этого сообщения является строка описывающая ошибку (она появляется в заголовке окна диагностики).
- произошло переполнение стека интерпретатора. Заголовок окна описывает эту ситуацию.
- любому объекту послано сообщение **остановка**. В заголовке окна указано 'Встретился останов'
- пользователь нажал Ctrl+Break. В заголовке окна указывается 'Пользовательское прерывание'.

Первые две причины приводят к невозобновляемому прерыванию активного процесса (т.е. невозможно вернуться к выполнению процесса после появления окна диагностики). Вторые две причины позволяют возобновить выполнение прерванного процесса. Для этого достаточно выбрать в меню рамки пункт “возобновить”. Указанная в окне диагностики информация достаточна для локализации многих проблем и дальнейшего их устранения. В любом случае у пользователя остается возможность получить более подробную информацию о состоянии процесса запросив вместо окна диагностики окно отладчика. Для этого надо выбрать второй пункт из меню рамки окна диагностики — “отладить”.

Окно отладчика (см. рис. 10), появляющееся вместо окна диагностики, состоит из четырех рамок:

- список контекстов — содержит ту же информацию, что и окно диагностики. Если выбран один из контекстов, остальные рамки будут содержать относящуюся к нему информацию.
- список объектов, доступных в контексте — приемник метода и локальные переменные.

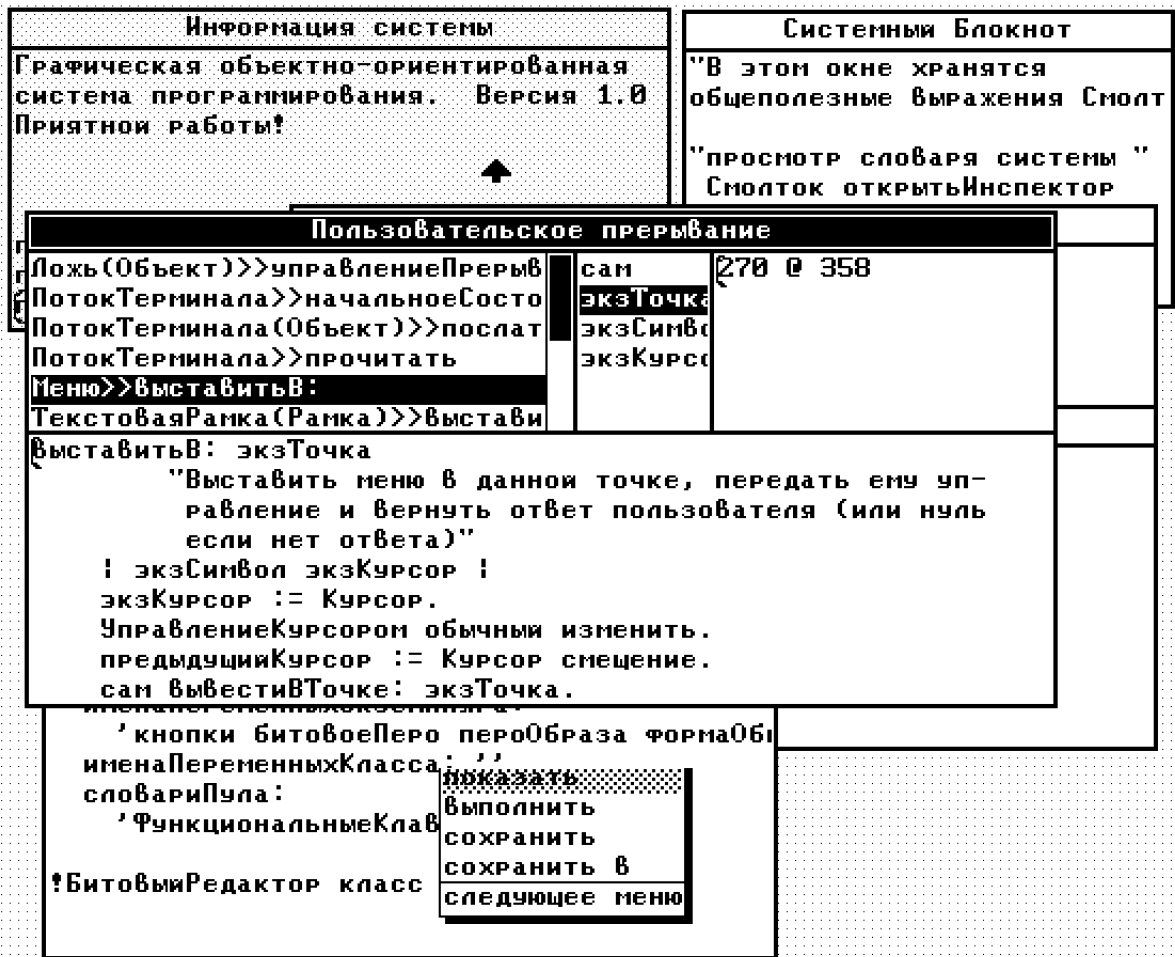


Рис. 10. Отладчик

- рамка редактирования доступных объектов — вместе с предыдущей рамкой образует аналог окна просмотра словаря, позволяя просматривать и редактировать доступные в контексте объекты
- текст метода, выполнение которого представляет контекст — текстовая рамка со всеми средствами редактирования. При выборе контекста появляется текст соответствующего метода, в котором выбрано выражение, вычисление которого привело к остановке вычислений. Если текст какого-либо метода редактировался в отладчике, в любом случае становится невозможным возобновить выполнение процесса, а можно только запустить его заново.

4 Заключение

На этом мы завершим первое знакомство с системой **Smalltalk**. После всего сказанного, можно включать компьютер и, если на нем есть любая **Smalltalk**-система, запускать ее и, прежде чем создавать что-то свое, изучать ее "внутреннее строение", то есть изучать многочисленные встроенные классы системы **Smalltalk**. Надеемся следующую методическую разработку по системе **Smalltalk** посвятить основным классам системы.

А Приложение

Приведем здесь в дополнение к ранее введенным в форме Бэкуса-Наура синтаксическим правилам языка **Smalltalk** их представление в графической форме. Возможно кому-то это другое представление покажется более простым и поможет лучше разобраться в синтаксических констукциях **Smalltalk**'а. При этом двойной рамкой выделяются определяемые конструкции языка.

Литература

- [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] *Технологический модуль объектно-ориентированного программирования (ТМООП). Руководство пользователя*. — М.: Ин-т проблем информатики АН СССР, 1990
- [4] *Технологический модуль объектно-ориентированного программирования (ТМООП). Описание языка*. — М.: Ин-т проблем информатики АН СССР, 1990
- [5] *Технологический модуль объектно-ориентированного программирования (ТМООП). Описание применения*. — М.: Ин-т проблем информатики АН СССР, 1990
- [6] Иванов А., Кремер Ю., *Язык Smalltalk: концепция объектно-ориентированного программирования* // КомпьютерПресс — 1992, № 4 — С. 21–31
- [7] Фути К., Судзуки Н., *Языки программирования и схемотехника СБИС*: пер. с япон. — М.: «Мир», 1988
- [8] Кирютенко Ю.А., Савельев В.А., *Объектно-ориентированное программирование и язык Smalltalk. Общие концепции и синтаксис*. — Ростов-на-Дону: УПЛ РГУ, 1995

Содержание

1	Введение	3
2	Введение в интерфейс пользователя	4
2.1	Рабочее окно	10
2.2	Системная информация	12
2.3	Суфлер	13
2.4	Просмотр диска	14
3	Программирование в системе Smalltalk	17
3.1	Окно просмотра иерархии классов	17
3.2	Инспектор класса	21
3.3	Редакторы и инспекторы	21
3.4	Средства отладки	23
4	Заключение	27
A	Приложение	28
	Литература	35

Список рисунков

1	Триада MVC	4
2	Вид дисплея русифицированной системы Smalltalk . . .	6
3	Клавиши, используемые в реализациях Smalltalk для IBM PC	7
4	Суфлер над окном просмотра диска запрашивает имя нового файла	13
5	Окно просмотра диска	14
6	Окно просмотра иерархии классов	18
7	Инспектор класса	21
8	Инспектор словаря	23
9	Окно диагностики	24
10	Отладчик	26