

ЯЗЫК  
ПРОГРАММИРОВАНИЯ

# SMALLTALK-80

Адель Голдберг

Дейвид Робсон

КОНФИДЕНЦИАЛЬНО

## Предисловие

Успехи в конструировании и производстве компьютеров привели к тому, что множество людей, далеких от программирования, стали использовать их в своей работе. Чтобы эти возросшие контакты стали максимально плодотворными, необходимо еще и существенное продвижение в области конструирования и производства программного обеспечения. Система Smalltalk-80 является результатом десятилетних исследований по созданию программного обеспечения, предназначенного для высоко функционального и интерактивного контакта пользователя с персональной вычислительной системой, и пятилетних дополнительных усилий по ее разработке и модернизации.

Эта книга — переработанный вариант нашей первой книги, вышедшей в 1983 году и содержащей подробное описание системы Smalltalk-80. Книга разделена на три части:

Часть первая — краткий обзор концепций и синтаксиса языка программирования.

Часть вторая — снабженное примечаниями и примерами описание функциональных возможностей системы.

Часть третья — примеры проектирования и разработки приложений среднего размера.

Первая часть описывает подход системы Smalltalk-80 к представлению и обработке информации. Пять слов — *объект, сообщение, класс, экземпляр и метод* — основа словаря, в терминах которого рассматривается Smalltalk. В этой части книги определяются основные термины и вводится синтаксис языка программирования Smalltalk-80.

Вторая часть книги состоит из описания различных видов объектов, уже представленных в системе программирования Smalltalk-80. Программист может создавать и дополнять ее новыми видами объектов, которые необходимы для решения конкретных прикладных задач, однако в стандартной поставке системы уже имеется широкий набор общеупотребительных объектов. В этой части перечислены, прокомментированы и проиллюстрированы сообщения, которые можно посылать объектам представленных классов.

Третья часть книги — пример того, как можно добавлять в систему новые виды объектов, необходимые для разработки прикладной программы. Конкретные примеры касаются создания дискретных, управляемых событиями моделей, таких как станции по обслуживанию автомобилей, банки или информационные системы. Некоторые читатели могут приступить к чтению третьей части книги сразу после прочтения первой, обращаясь ко второй части всякий раз, когда необходимо получить справку по встретившимся выражениям Smalltalk-80.

Во всех трех частях исправлены ошибки, допущенные в первом издании, которые были замечены, в том числе и нашими читателями. Четвертая часть книги 1983 года, описывавшая механизм виртуальной машины Smalltalk-80, в этом издании опущена. Широкая доступность коммерческих реализаций, продаваемых и лицензируемых фирмой ParcPlace Systems, сделали ее включение излишней.

---

### О задачах, которые стояли перед авторами книги

Работа над первой книгой о языке Smalltalk-80 была сложной, частично из-за социологии создания системы, частично из-за разнообразия требований пользователей к такой системе. На наш взгляд основные сложности создания книги обусловлены следующими четырьмя причинами:

- Smalltalk — это новая парадигма программирования;
- Smalltalk основан на небольшом числе понятий, определяемых в необычных терминах;
- Smalltalk — графическая, интерактивная среда программирования;

- Smalltalk — это большая система.

### *Smalltalk — новая парадигма программирования*

В начале 1970-х годов в исследовательском центре фирмы Херох в г. Пало Альто, Калифорния (Palo Alto, California) была создана исследовательская группа по проблемам обучения (Learning Research Group), которая начала работу над такой концепцией программирования, в рамках которой люди с различной степенью подготовки могли бы эффективно и с удовольствием использовать вычислительную силу компьютера. В 1981 группа была переименована в группу по концепции программного обеспечения (Software Concept Group — SCG). Целью SCG стало создание мощной информационной системы, в которой пользователь мог бы хранить, получать и обрабатывать информацию так, чтобы система развивалась и расширялась вместе с развитием представлений и требований пользователя. Количество и разнообразие компонентов системы должны расти в соответствии с ростом понимания пользователем того, как эффективно работать с системой.

Стратегией SCG в реализации этого подхода стала концентрация на двух принципиальных областях исследования: языке описания (языке программирования), который служит интерфейсом между моделью в голове человека и моделью в компьютере; и языке взаимодействия (интерфейсе пользователя), который согласовывает коммуникативные системы человека и компьютера. Исследования по языку Smalltalk проходили по 2-4-х летнему циклу: создание системы, воплощающей текущее понимание требований к программному обеспечению; реализация прикладных программ для проверки способности системы поддерживать эти прикладные программы; и, наконец, основанное на полученном опыте переформулирование требований к системе и перепроектирование языка программирования и/или пользовательского интерфейса.

Система Smalltalk-80 представляет результат пятикратного прохождения этого цикла. Исследования все еще продолжаются<sup>1</sup>. Мы надеемся, что представление детального описания результатов исследований в виде книги будет способствовать увеличению среди программистов числа сторонников идей группы SCG. Продолжающиеся исследования дают нам право сказать, что описанная в книге система программирования — лишь "движущаяся цель", а информация в книге представляет собой только одну из остановок на длинном пути к совершенному программному продукту. Необходимость задержаться на достаточно долгий срок, чтобы описать созданное, не делает проще задачу написания книги.

### *Smalltalk основывается на малом числе понятий*

Smalltalk основан на малом числе понятий, но сформулированы они в необычных терминах. Благодаря единообразию, с которым ориентация на объекты и сообщения реализована в системе, для понимания языка необходимо разобраться лишь в нескольких новых понятиях программирования. С одной стороны, это означает что все основные понятия можно быстро объяснить читателю и затем продемонстрировать различные способы их применения. Новые понятия вводятся путем определения ранее упомянутых пяти слов, которые составляют словарь языка Smalltalk: объект, сообщение, класс, экземпляр, метод. Но, с другой стороны, каждое из этих пяти слов определяется в терминах остальных, то есть получается, что читатель должен знать "все", прежде чем узнает что-либо.

### *Smalltalk — среда программирования*

Smalltalk — графическая, интерактивная среда программирования. В соответствии с концепцией персональных вычислений, Smalltalk разработан так, что каждый доступный пользователю компонент системы может быть представлен осмысленным образом для изучения и обработки. Интерфейс пользователя в языке Smalltalk поддерживает попытки создавать визуальный язык для представления каждого объекта. Предполагаемая аппаратура для работы с системой Smalltalk должна включать графический дисплей высокого разрешения и устройство указания места на дисплее, такое как световое перо или мышь. Эти устройства

<sup>1</sup> В настоящее время исследования продолжаются только в части интерфейса пользователя. Язык программирования уже не изменяется и проходит этап стандартизации. — Прим. ред.

позволяют выбирать информацию, отображаемую на экране, и посылать сообщения системе для взаимодействия с выбранной информацией.

Один из способов объяснить детали системы Smalltalk-80 состоит в том, чтобы начать с интерфейса пользователя и описывать по порядку каждое средство доступа к объектам системы. Такое объяснение может начинаться со сценариев, описывающих пути, которыми программист может взаимодействовать с системой. Каждый сценарий при этом будет моментальным снимком состояния динамической системы. В линейном статическом стиле книга попытается передать динамику многочисленных путей доступа к большому количеству разнообразной информации.

Эти стороны системы - важная часть того, что Smalltalk дает как среда разработки прикладных программ. Однако, чтобы можно было объяснять, как на самом деле работает графический интерфейс пользователя, читатель сначала должен понять язык программирования. Поэтому в книге описание системы начинается с языка, а информация об объектах системы, поддерживающих интерфейс пользователя, кроме классов ядра графики, оставляется вне книги. Детальному обсуждению интерфейса пользователя посвящена другая книга о системе Smalltalk-80. Это книга Adele Goldberg "*Smalltalk-80: The Interactive Programming Environment*" ("Smalltalk-80: интерактивная среда программирования").

### *Smalltalk — большая система*

Система Smalltalk-80 состоит из большого числа компонентов. Она включает объекты, обеспечивающие функции, обычно присущие операционной системе: автоматическое управление памятью, файловую систему, управление дисплеем, редактирование текстов и графических образов, ввод с клавиатуры и с мыши, отладчик, измерение производительности, планирование процессов, компиляцию и декомпиляцию. Есть много видов объектов, которые необходимо изучить.

Smalltalk создан в соответствии с моделью взаимодействующих объектов. Крупные приложения рассматриваются точно так же, как и фундаментальные объекты, из которых построена система. Взаимодействие между большинством примитивных объектов выглядит так же, как и взаимодействие на высоком уровне между компьютером и пользователем. Объекты системы поддерживают принцип модульности: функционирование любого объекта не зависит от внутренних деталей других объектов. Минимизация взаимной зависимости компонентов системы снижает ее сложность. Для дальнейшего снижения сложности системы используется объединение одинаковых объектов в группы. Это достигается через понятие класса. Классы являются и основным механизмом расширения системы Smalltalk. Определяемые пользователем классы становятся частью системы на тех же основаниях, что и классы ядра системы. Понятие подкласса позволяет разделить систему на составляющие для того, чтобы избежать повторений одних и тех же понятий в разных местах.

Управление сложностью — один из основных вкладов языка Smalltalk в современное программное обеспечение. Первые примеры языка очень просты и взяты из упражнений по программированию, типичных для большинства книг по языкам программирования. Такие примеры коротки и иллюстрируют, как правило, одну или две черты языка. Эти примеры не позволяют продемонстрировать силу и особенности языка и могут быть легко выполнены на других языках программирования. Достоинства языка Smalltalk проявляются при проектировании и реализации больших прикладных программ, или при попытке осуществить модернизацию системы. Например, рассмотрим словарь, часто используемую структуру данных в системе Smalltalk-80. Вы можете разработать, запрограммировать, отладить и установить в систему новую реализацию словаря без разрушения работающей системы. Это возможно до тех пор, пока не будет разрушен интерфейс сообщений, от которого зависит функционирование других объектов системы.

Smalltalk-80 поддерживает ряд интересных инструментальных средств, а именно: классы и экземпляры, как модули для организации и совместного использования информации; подклассы, как средство наследования и совершенствования существующих возможностей. Вместе с интерактивным способом разработки программы это делает систему Smalltalk-80 мощной средой для создания прототипов новых приложений и усовершенствования существующих.

При написании книги о такой сложной и богатой возможностями системе очевидно, что некоторые вещи приходится опускать. Еще раз отметим, что мы опустили в этой книге подробное описание интерфейса пользователя и того, как могут создаваться интерактивные графические приложения. Мы сосредоточили свое внимание на языке и классах ядра системы.

---

## Кому адресована эта книга

Книга предполагает некоторую компьютерную грамотность со стороны читателя. Мы предполагаем, что читатель

- знает для чего необходимы системы программного обеспечения;
- является программистом или проектировщиком языков программирования и знает хорошо хотя бы один язык программирования;
- знаком с понятием синтаксиса выражений и выполнением выражений интерпретатором;
- знаком с последовательным выполнением инструкций в компьютере, с такими управляющими структурами, как итерация и рекурсия, с ролью структур данных;
- понимает необходимость улучшения представления и обработки информации в вычислительной системе;
- ищет новые идеи для создания программного обеспечения (прикладной программы), которые поддерживали бы способность выражать программное решение поставленной задачи способом, тесно связанным с естественным представлением решения.

Система Smalltalk-80 основывается как на идеях алгоритмического языка Simula (Симула), так и на идеях Элана Кея (Alan Kay), который первым предложил нам попробовать создать однородную объектно-ориентированную систему. Нынешнее воплощение этих идей является результатом трех факторов: исследования, выполненного в Xerox Palo Alto Research Center; сотрудничества с группой по рецензированию результатов исследования; настойчивых усилий группы служащих фирмы Xerox, с успехом создавших в 1988 году коммерческую организацию для системы Smalltalk — ParcPlace Systems.

В августе 1980 года несколько производителей вычислительной техники были приглашены на обсуждение нашей второй попытки написать книгу о системе Smalltalk и о ее последней реализации. Первая попытка, описывавшая систему Smalltalk-76, была оставлена в надежде создать более переносимую систему для распространения ее вне исследовательского центра Xerox. Вторая попытка была книгой частично исторической, частично посвященной изложению нашей точки зрения на основные концепции программирования и, частично, описанием функциональных особенностей новой системы Smalltalk. Мы думали назвать ее "*Smalltalk Dreams and Schemes*" ("Мечты и схемы Smalltalk"), как отражение двойственной цели книги. Производители, которые терпеливо рецензировали наши материалы работали в Apple Computer, Digital Equipment Corporation, Hewlett-Packard и Tektronix. Эти компании были выбраны потому, что они разрабатывали компьютерную технику. Мы надеялись, что рассматривая наши материалы, они познакомятся с необычным программным продуктом и, возможно, посветят некоторое время проблеме создания аппаратных средств специально для подобных систем. Мы знали, что аппаратные системы, имевшиеся на рынке в то время и планируемые к выпуску в будущем, имели недостаточную мощность, чтобы обеспечить поддержку наших идей. Вместо того, чтобы разрабатывать программное обеспечение под уже готовые вычислительные системы, мы решили попробовать получить аппаратные средства, разработанные под наше программное обеспечение. И мы преуспели в этом!

Производители выделили сотрудников своих исследовательских лабораторий для чтения второй версии книги. Книга сильно выиграла от многочисленных обсуждений и интенсивной работы рецензентов. Часть книги была полностью переписана в результате их конструктивной критики. Рецензенты ответственны за то, что мы продолжали интенсивно работать над завершением книги, но они не несут никакой ответственности за ошибки и неточности в ее окончательном тексте. Каждая из групп рецензентов по крайней мере один раз реализовала систему, чтобы проверить наши спецификации виртуальной машины Smalltalk-80. И то, чего мы добились на сегодняшний день, отражает их тщательную работу.

Как авторы этой книги, мы несем ответственность за создание описания системы Smalltalk-80. Но честь создания системы принадлежит всем членам SCG. Всем этим людям мы выражаем нашу признательность, благодарность и любовь. Дэн Ингалс (Dan Ingalls) руководил проектированием и разработкой системы. Питер Дейч (Peter Deutsch) на Dorado, Глен Креснер (Glenn Krasner) на Alto и Ким Маккол (Kim McCall) на Dolphin (также известном как Xerox 1100 Scientific Information Processor) приобретали опыт реализации виртуальной машины на компьютерах фирмы Xerox. Идеи интерфейса пользователя и их реализации, и управление процессом выпуска версий выполнили: Джеймс Олтофф (James Althoff) — разработка интерфейса пользователя, Роберт Флигал (Robert Flegal) — проектирование графического редактора, Тед Кэлер (Ted Kaehler) — работа над проблемами виртуальной памяти, Диана Мери (Diana Merry) — наш гуру по обработке текста, и Стив Патц (Steve Putz) — управление версиями. Пеги Эспри (Peggy Asprey), Марк Мейер (Marc Meyer), Билл Финзер (Bill Finzer) и Лора Гоулд (Laura Gould), стараясь сохранить прикладные программы в постоянно развивающейся системе, отладили многочисленные изменения. Плодотворное чтение рукописи в разные периоды создания книги выполнили Майкл Рутенберг (Michael Rutenberg), Майкл Медсен (Michael Madsen), Сузанна Бодкер (Susanne Bodker) и Джей Трой (Jay Trow). Помощь в редактировании оказали Рейчел Ратерфорд (Rachel Rutherford) и Джанет Морленд (Janet Moreland).

Глава 18 о ядре графики Smalltalk-80 взята из рукописи, подготовленной Дэном Ингалсом для журнала Byte. Графические изображения в Главах 18, 19 и 20, создали Роберт Флигал (особенно рисунки 18.1 и 20.1), Дэн Ингалс и Адель Голдберг (особенно рисунки 20.2 и 20.3). Стив Патц предложил ценную помощь в создании изображений для Главы 17. Изображения перед первой и второй частями и все изображения, располагаемые в начале Глав с 1 по 20, созданы Адель Голдберг. Изображения для третьей части и все изображения, располагаемые в начале Глав с 21 по 26, созданы Робертом Флигалом. Эти изображения создавались при помощи редактора графики системы Smalltalk-80 в комбинации со сканером изображения с низкой разрешающей способностью, который разработал Джозеф Мейлсон (Joseph Maleson).

Мы также выражаем наши благодарности всем участникам рецензирования. С ними мы установили постоянные научные связи, которые, надеемся, будут развиваться и расти. Мы ощущали постоянную поддержку от администратора лаборатории Берта Сазерленда (Bert Sutherland). Рецензентами и реализаторами были: из APPLE — Рик Мейер (Rick Meyers) и Дейвид Кессерес (David Casseres); из Digital Equipment Corporation — Стоуни Болард (Stoney Ballard), Эрик Осман (Eric Osman), Стив Широн (Steve Shirron); из Hewlett-Packard — Алек Дара-Абрамс (Alec Dara-Abrams), Джо Фальконе (Joe Falcone), Джим Стингер (Jim Stinger), Берт Спилпенинг (Bert Speelpenning) и Джефи Истмен (Jefi Eastman); из Tektronix — Пол Маккалах (Paul McCullough), Элан Вирфс-Брок (Allen Wirfs-Brock), Д. Джейсон Пенни (D. Jason Penney), Ларри Катц (Larry Katz), Роберт Рид (Robert Reed) и Рик Самко (Rick Samco). Мы благодарим их компании и администраторов за терпение и готовность уйти от промышленных стандартов, по крайней мере на один краткий миг: в APPLE — Стива Джобса (Steve Jobs) и Брюса Даниэльса (Bruce Daniels); в Digital Equipment Corporation — Ларри Самбурга (Larry Samburg); в Hewlett-Packard — Пола Стофта (Paul Stoft), Джима Дули (Jim Duley) и Теда Лалиотиса (Ted Laliotis) и в Tektronix — Джека Граймза (Jack Grimes) и Джорджа Райна (George Rhine). Люди из Tektronix подготовили детальные обзоры на аудиоленте, так что мы могли не только видеть наши ошибки, но и слышать их!

Мы пользуемся этой благоприятной возможностью, чтобы выразить благодарность тем, кто помог Адели (Adele), Дейву (Dave), Глену (Glenn) и Питеру (Peter) создать ParcPlace Systems из фирмы Xerox в 1988 году. Это были Русс Пенсин (Russ Pencin), Нанет Хартер (Nanette Harter), Рон Картер (Ron Carter), Дейвид Лейбс (David Leibs), Стивен Поуп

(Stephen Pope), Фрэнк Джексон (Frank Jackson), Элан Шифман (Allan Schiffman) и Ричард Стейджер (Richard Steiger).

Мы надеемся, что эта книга и ее верные друзья будут способствовать распространению понятий языка Smalltalk в компьютерном мире. Если это произойдет, этот успех мы рады будем разделить с нашими коллегами из Xerox Palo Alto Research Center.

---

## PostScriptum о печати этой книги

Текст оригинала этой книги был представлен издателю на магнитной ленте. Лента содержала определяющие коды форматирования различных видов текстовых объектов в рукописи. Реальное создание каждого типа объекта обеспечивалось издателем. Этот процесс прошел гладко в значительной степени благодаря усилиям и терпению Ейлин Колэхан (Eileen Colahan) из International Computaprint Corporation и Джима Деволфа (Jim DeWolf) из Addison-Wesley.

Многие графических изображения, которые представляют в книге примеры графики экрана системы ST-80 и все художественные заставки к частям и главам, были напечатаны с помощью системы Platemaker, разработанной Гари Старквейзером (Gary Starkweather) в Imaging Sciences Laboratory of PARC. Мы хотели бы поблагодарить Гари, Эрика Ларсона (Eric Larson) и Джулиан Опп (Julian Orr) за создания Platemaker, доступного для нас.

Адель Голдберг (Adele  
Goldberg)  
Дэвид Робсон (David Robson)

*г. Пало-Альто, Калифорния*