

ЧАСТЬ ПЕРВАЯ

Первая часть этой книги представляет собой обзор языка программирования Smalltalk-80. В первой главе вводятся основные понятия и словарь языка без описания его синтаксиса. Здесь описываются объекты, сообщения, классы, экземпляры и методы. Эти понятия весьма подробно обсуждаются в последующих четырех главах книги, по мере описания соответствующих частей синтаксиса языка программирования. Во второй главе описывается синтаксис выражений. Выражения позволяют ссылаться на объекты и описывать сообщения. Третья глава посвящена основам синтаксиса классов и методов. Классы и методы позволяют создавать новые виды объектов и модифицировать существующие. Последние две главы посвящены описанию подклассов и метаклассов — двух важных понятий, уточняющих роль классов в системе Smalltalk-80.

Конфиденциально

1

Объекты и сообщения

Объекты и сообщения

Классы и экземпляры

Пример приложения

Системные классы

Арифметика

Структуры данных

Управляющие структуры

Среда программирования

Просмотр и взаимодействие

Коммуникации

Словарь терминов

Конфиденциально

Объекты и сообщения

Объект — это компонент системы Smalltalk-80. Объектами, например, являются

- числа
- строки символов
- очереди
- словари
- прямоугольники
- каталоги файлов
- редакторы текстов
- программы
- компиляторы
- вычислительные процессы
- финансовые отчеты
- визуальные представления информации

Каждый объект состоит из некоторой принадлежащей ему памяти и множества операций. Природа операций над объектом зависит от типа компонента, представляемого объектом. Объекты, представляющие числа, вычисляют арифметические функции. Объекты, представляющие структуры данных, хранят и обрабатывают информацию. Объекты, представляющие позиции и области, выдают информацию об их расположении относительно других подобных объектов.

Сообщение — это запрос к объекту выполнить одну из операций. Сообщение определяет, какая требуется операция, но не определяет как эта операция должна быть выполнена. Только *получатель* — объект, которому послано сообщение, — определяет, как выполнять требуемую операцию. Например, сложение выполняется посылкой сообщения объекту, представляющему число. Сообщение указывает, что затребована операция сложения, а также определяет, какое число должно быть добавлено к получателю. Как уже отмечалось, сообщение не определяет, как будет выполнено сложение. Это определяет получатель сообщения, в данном случае число. Вычисление рассматривается как встроенная способность объекта, которая может быть единообразно вызвана посылкой сообщения.

Множество всех сообщений, на которые может отвечать объект, называется интерфейсом объекта с остальной частью системы. Воздействовать на объект можно только через его интерфейс. Основное свойство объекта состоит в том, что его памятью можно управлять только посредством собственных операций. Основное свойство сообщений состоит в том, что они представляют единственный способ воздействия на объекты. Эти свойства гарантируют, что реализация любого объекта не зависит от внутреннего устройства других объектов, а определяется только сообщениями, на которые этот объект может отвечать.

Сообщения обеспечивают модульность системы, так как они задают тип требуемой операции, а не то, как эта операция будет выполняться. Например, в системе Smalltalk-80 существует несколько представлений числовых значений. Малые и большие целые числа, рациональные числа, числа с плавающей точкой представляются различными способами. Все

они понимают сообщение, запрашивающее вычисление их суммы с другим числом, но каждое представление предполагает различные способы вычисления этой суммы. Для взаимодействия с числом необходимо лишь знать, какие сообщения оно может принимать, но не нужно знать, как это число представлено.

Другие среды программирования также используют объекты и сообщения для облегчения модульного проектирования. Например, Simula использует объекты для описания имитационных моделей, а Hydra — для описания средств распределенной операционной системы. В Smalltalk-80 объекты и сообщения используются для обеспечения всей среды программирования. Как только объекты и сообщения будут поняты, становится доступной вся система.

Например, одна из широко применяемых в программировании структур данных — словарь, который связывает имена и значения. В системе Smalltalk-80 словарь представлен объектом, который может выполнять две операции: связывать имя с новым значением и находить значение по заданному имени. Программист, использующий словарь, должен знать, как задать эти две операции в виде сообщений. Словарные объекты понимают сообщения, которые делают запросы типа: “соединить имя *Brett* со значением *3*” и “какое значение связано с именем *Dave*?”. Так как в Smalltalk-80 все является объектом, то имена *Brett* и *Dave*, значения *3* и *30*, также представлены объектами. Хотя любопытный программист и может пожелать узнать, как представлена связь имени со значением в словаре, такая информация о внутреннем устройстве не требуется для успешного использования словаря. Знание внутренней реализации словаря важно только тому программисту, который работает над определением словаря как такового.

Важная часть разработки программ на языке Smalltalk-80 — определение того, какие виды объектов должны быть описаны в системе и какие имена сообщений обеспечат необходимый словарь взаимодействий этих объектов. Всегда, когда программист определяет сообщения, которые могут посылаться объекту, он конструирует язык. Походящий выбор объектов зависит, конечно же, от цели, ради которой объект включается в систему, и от структуры информации, которой объект будет управлять. Например, если моделирование парка развлечений создается с целью накопления данных об очередях на различные аттракционы, полезно ввести объекты, представляющие аттракционы, рабочих, обслуживающих аттракционы и очереди ожидания, и людей, посетивших парк. Если, дополнительно, целью моделирования является контроль за потреблением продуктов питания отдыхающими в парке, то потребуются объекты, представляющие эти расходуемые ресурсы. Если же в модели предусматривается подсчет количества потраченных денег, то в модели должна быть представлена информация и о стоимости аттракционов.

Выбор объектов, необходимых для поставленных целей — первый шаг проектирования приложения в системе Smalltalk-80. На самом деле, нельзя сказать ничего определенного о некоем “правильном способе” выделения объектов. Как и в любом другом процессе проектирования — это приобретаемое мастерство. Различный выбор объектов обеспечивает различные основы и для расширения приложений, и для использования объектов в других задачах. Квалифицированный программист на языке Smalltalk-80 должен помнить, что объекты, создаваемые им, могут оказаться полезными и во многих других приложениях, если определить семантически полный набор функций для создаваемых объектов. Например, словарь, в котором пары “имя-значение” можно и удалять и добавлять, конечно же, более полезен, чем словарь, в котором пары можно только добавлять.

Классы и экземпляры

Класс описывает реализацию множества объектов, которые представляют подобные компоненты системы. Отдельные объекты, описываемые классом, называются его *экземплярами*.

Класс описывает структуру памяти своих экземпляров и как они выполняют свои операции. Например, в системе существует класс, который описывает реализацию объектов, представляющих прямоугольные области. В этом классе описывается, как отдельные экземпляры "запоминают" свое положение, а также как экземпляры выполняют необходимые операции над прямоугольными областями. Каждый объект в системе Smalltalk-80 является экземпляром некоторого класса. Даже объект, представляющий уникальный компонент системы, реализуется как единственный экземпляр класса. Программирование в системе Smalltalk-80 состоит из создания классов, экземпляров классов и последовательности сообщений, которыми обмениваются эти объекты.

Экземпляры одного класса подобны как своими внешними, так и внутренними свойствами. Внешние свойства объекта — это сообщения, составляющие его интерфейс. Все экземпляры класса имеют одинаковый интерфейс. Внутренние свойства объекта — это множество *переменных экземпляра*, составляющих его индивидуальную память, и множество *методов*, которые описывают, как выполнять его операции. Переменные и методы экземпляров непосредственно не доступны для других объектов. Все экземпляры класса используют одно и то же множество методов для описания своих операций. Например, экземпляры, представляющие прямоугольники, отзываются на одно и то же множество сообщений, и все они используют одни и те же методы, чтобы определить, как реагировать на сообщения. Каждый экземпляр класса имеет свой собственный набор переменных экземпляра, но все они имеют, вообще говоря, одинаковое количество переменных экземпляра. Например, все экземпляры, представляющие прямоугольники, имеют две переменные экземпляра.

Каждый класс имеет имя, которое описывает тип компонента системы, представляемый его экземплярами. Например, класс, экземпляры которого представляют последовательности символов, называется **String** (Строка); класс, экземпляры которого представляют положение в пространстве, называется **Point** (Точка); класс, экземпляры которого представляют прямоугольные области, называется **Rectangle** (Прямоугольник); класс, экземпляры которого представляют вычислительные процессы, называется **Process** (Процесс). Имена классов, как элементы языка программирования, будут набираться в этой книге всегда одним и тем же специальным шрифтом, который также будет использоваться и во всех текстах, представляющих выражения языка Smalltalk-80.

Каждая переменная экземпляра в принадлежащей экземпляру памяти ссылается на некоторый объект, называемый значением этой переменной. Значения двух переменных экземпляра класса **Rectangle** являются экземплярами класса **Point**, которые представляют противоположные углы его прямоугольной области. Те факты, что экземпляр класса **Rectangle** имеет две переменные, и что эти переменные ссылаются на экземпляры класса **Point**, — строго внутренняя информация, недоступная за пределами данного прямоугольника.

Каждый метод в классе показывает, как выполнить операцию, затребованную соответствующим сообщением. Когда некоторое сообщение посылается экземпляру класса, выполняется соответствующий этому сообщению метод. Методы, используемые всеми прямоугольниками, описывают, как выполнять их операции над двумя точками, представляющими противоположные углы прямоугольника. Так, если сообщение запрашивает прямоугольник о местоположении его центра, соответствующий метод показывает, как найти точку посередине между двумя противоположными углами.

Класс включает в себя метод для каждого типа операций, которые его экземпляры могут выполнить. Метод может сделать некоторые изменения в собственной памяти объекта и/или послать другие сообщения. Метод также определяет объект, который должен быть возвращен как значение вызвавшего метод сообщения. Методы объекта имеют доступ к переменным экземпляра этого объекта, но не имеют доступа к переменным экземпляра других объектов. Например, метод экземпляра класса **Rectangle**, используемый для вычисления центра прямоугольника, имеет доступ к двум его переменным, экземплярам класса **Point**; однако, этот метод не имеет доступа к внутренним переменным этих экземпляров класса **Point**. Данный метод должен определить сообщения, которые будут посланы экземплярам класса **Point**, чтобы выполнить требуемые вычисления.

Небольшая часть методов системы Smalltalk-80 не выражена средствами языка Smalltalk-80. Такие методы называются примитивными. Примитивные методы встроены в виртуальную машину и не могут изменяться программистами. Вызываются и выполняются эти методы точно так же, как и все другие методы системы. Примитивные методы выполняют операции низкого уровня, например, арифметические операции или доступ к индексированным переменным экземпляра; они обеспечивают доступ ко всем устройствам компьютера, например, к дискам и клавиатуре. Кроме того, примитивные методы используются для реализации некоторых операций высокого уровня, критичных с точки зрения производительности.

Пример приложения

Примеры — важная часть описания языка и среды программирования. Многие примеры, представленные в данной книге, взяты из классов, находящихся в стандартной системе Smalltalk-80. Другая часть примеров взята из классов, которые могут быть добавлены в систему для расширения сферы ее применения. В первой части книги демонстрируется пример, который можно добавить в систему для того, чтобы обеспечить возможность учета простых финансовых операций частными лицами, домовладельцами, руководителями предприятий малого бизнеса. Приложение позволяет вводить информацию о финансовых сделках и отображать ее на дисплее. Рис. 1.1 показывает пример того, в каком виде информацию можно представить на экране дисплея в системе Smalltalk-80. Два верхних окна картинке показывают два вида диаграмм отчета о тратах денег. Следующая диаграмма показывает, как количество наличных денег колебалось со временем, пока совершались сделки. Внизу находятся два поля ввода информации, в которые пользователь может записывать свои доходы и расходы, чтобы добавлять их к отчету. Когда вводится новая информация, все три диаграммы должны автоматически изменяться, отражая произошедшие изменения. На рис. 1.2 показан вид картинке после добавления расходов на питание.

Рис. 1.1

Это приложение требует добавления в систему Smalltalk-80 нескольких новых классов. Новые классы описывают различные виды окон, а также основную информацию о финансовых операциях. Класс, который фактически записывает финансовую информацию, называется `FinancialHistory` (ФинансовыйОтчет). Он будет приводиться как пример в следующих четырех главах. Этот пример использует некоторые уже существующие в системе классы: числа — чтобы представлять суммы денег, и строки — чтобы указывать причины расходов и источники доходов.

Класс `FinancialHistory` предназначен для знакомства с основными понятиями языка Smalltalk-80, поскольку его структуру, сообщения и методы легко описать. Он обеспечивает шесть операций:

1. Создает новый объект финансового отчета с некоторой начальной суммой наличных денег.
2. Запоминает, на какие нужды была потрачена определенная сумма денег.
3. Запоминает источник каждого получения определенной суммы денег.
4. Выясняет, сколько денег есть в наличии.
5. Выясняет, сколько было потрачено и на какие цели.
6. Выясняет, сколько денег было получено и из каких источников.

Реализация этих операций определяется в описании класса, приведенном на форзаце книги. Форма описания класса будет приведена в главах 3, 4 и 5.

Системные классы

Система Smalltalk-80 включает в себя множество классов, которые обеспечивают стандартные возможности языка и среды программирования. Среди них арифметика, различные структуры данных, управляющие структуры и средства ввода-вывода. Эти классы системы будут подробно описаны во второй части книги. Рисунок 1.3 — это схема системных классов, представленных во второй части книги. Родственные классы здесь объединены линиями в группы; числа у границы группы указывают главы второй части, в которых дано описание классов этой группы.

Арифметика

Система Smalltalk-80 включает объекты, представляющие целые, рациональные и вещественные (действительные) числа. Вещественные числа могут быть представлены с точностью до 6 цифр. Целые числа, по абсолютной величине меньшие 2^{524288} , представляются точно. Рациональные числа можно представить целыми числами. Есть также классы для представления линейно упорядоченных величин (таких, например, как дата или время) и классы для представления генераторов случайных чисел.

Структуры данных

Большинство объектов в системе Smalltalk-80 — это структуры данных некоторого вида. В то время как многие объекты имеют свою собственную функциональность, в системе есть классы, представляющие более или менее чистые структуры данных. Например, это классы, представляющие различные виды наборов. Элементы одних наборов находятся в неупорядоченном виде, элементы других — упорядочены. Среди наборов с неупорядоченными элементами есть простые наборы, которые допускают дублирование своих элементов, и есть множества, которые не допускают дублирования. Существуют словари, которые связывают пары объектов. Среди наборов упорядоченных элементов одни определяют порядок внешним образом — по добавлению элементов, другие — на основе свойств самих элементов. Такие обшеупотребительные структуры данных как массивы и строки представляются классами с ассоциативным поведением (элементы связываются с индексами) и внешним упорядочением (соответствующим упорядочению индексов).

Рис. 1.3

Управляющие структуры

Система Smalltalk-80 включает объекты и сообщения, которые реализуют стандартные управляющие структуры, существующие в большинстве современных языков программирования. Они обеспечивают условный выбор подобно оператору if-then-else в Алголе и условное повторение подобно его же операторам while и until. Есть также объекты для представления независимых процессов, механизмы планирования и синхронизации взаимодействующих процессов. Для поддержки управляющих структур в систему введены два класса. Класс логических величин представляет два значения: истину и ложь. Блоки представляют последова-

тельности действий. Логические объекты и блоки применяются также и для создания новых видов управляющих структур.

Среда программирования

Существуют несколько классов системы Smalltalk-80, которые содействуют процессу программирования. Есть отдельные классы, представляющие методы системы в исходной (удобочитаемой для человека) форме, и в скомпилированной (выполняемой машиной) форме. Объекты, представляющие синтаксические анализаторы, компиляторы и декомпиляторы, работают с этими двумя формами методов. Объекты, представляющие классы, связывают методы с теми объектами, которые эти методы используют (экземплярами классов).

Объекты, представляющие организационные структуры для классов и методов, помогают программисту ориентироваться в системе, а объекты, представляющие историю модификации среды программирования помогают согласовывать независимую работу нескольких программистов. Даже состояние выполнения метода представляется объектом. Эти объекты называются контекстами и аналогичны стековым кадрам или активационным записям в других системах.

Просмотр и взаимодействие

Система Smalltalk-80 включает классы объектов, которые могут применяться для просмотра и редактирования информации. Определены классы для представления графических изображений, таких как точки, линии, прямоугольники и дуги. Так как система Smalltalk-80 ориентирована на дисплеи с побитовой адресацией, существуют классы для представления и управления битовыми картами образов. Существуют классы для представления и управления специальными битовыми картами для изображений шрифтов, текстов, курсора.

Из этих первоначальных графических объектов можно строить другие графические объекты, представляющие прямоугольные окна, меню команд и области выбора. Также есть объекты, отражающие действия пользователя на устройствах ввода и объекты, представляющие информацию на устройствах вывода. Классы, дающие возможность просмотра и редактирования, построены из этих компонентов и обеспечивают просмотр классов, контекстов и документов, содержащих текст и графику. Просмотр классов обеспечивается встроенным в систему фундаментальным механизмом взаимодействия с системой в интерактивном режиме. Эти возможности системы не затрагиваются в данной книге.

Коммуникации

Система Smalltalk-80 допускает связь с внешним носителем данных. Стандартный внешний носитель данных — файловая система на диске. Объекты системы представляют как отдельные файлы, так и каталоги файлов. Если есть сеть для передачи данных, то ее также можно представить посредством объектов и сделать доступной для использования.

Словарь терминов

объект

компонент системы, представленный некоторой собственной памятью и множеством операций.

сообщение

обращение к объекту с запросом выполнить операцию.

получатель

объект, которому послано сообщение.

интерфейс	множество сообщений, на которые отвечает объект.
класс	описание группы подобных объектов.
экземпляр	один из объектов, описанный классом.
переменная экземпляра	часть собственной памяти объекта.
метод	описание того, как выполнять одну из операций над объектом.
примитивный метод	операция, непосредственно выполняемая виртуальной машиной системы Smalltalk-80.
системные классы	множество классов, которые входят в стандартную систему Smalltalk-80.
FinancialHistory	имя класса, используемого в этой книге как пример.

Конфиденциально