

- генерирование визуального представления документа;
- отображение позиций экрана на элементы внутреннего представления. Это позволит определить, что имел в виду пользователь, когда указал на что-то в визуальном представлении.

Помимо данных целей имеются и ограничения. Во-первых, текст и графику следует трактовать единообразно. Интерфейс приложения должен позволять свободно помещать текст внутрь графики и наоборот. Не следует считать графику частным случаем текста или текст – частным случаем графики, поскольку это в конечном итоге привело бы к появлению избыточных механизмов форматирования и манипулирования. Одного набора механизмов должно хватить и для текста, и для графики.

Во-вторых, в нашей реализации не может быть различий во внутреннем представлении отдельного элемента и группы элементов. При одинаковой работе Lexi с простыми и сложными элементами можно будет создавать документы со структурой любой сложности. Например, десятым элементом на пересечении пятой строки и второй колонки мог бы быть как один символ, так и сложно устроенная диаграмма со многими внутренними компонентами. Но, коль скоро мы уверены, что этот элемент имеет возможность изображать себя на экране и сообщать свои размеры, его внутренняя сложность не имеет никакого отношения к тому, как и в каком месте страницы он появляется.

Однако второе ограничение противоречит необходимости анализировать текст на предмет выявления орфографических ошибок и расстановки переносов. Во многих случаях нам безразлично, является ли элемент строки простым или сложным объектом. Но иногда вид анализа зависит от анализируемого объекта. Так, вряд ли имеет смысл проверять орфографию многоугольника или пытаться переносить его с одной строки на другую. При проектировании внутреннего представления надо учитывать эти и другие потенциально конфликтующие ограничения.

### **Рекурсивная композиция**

На практике для представления иерархически структурированной информации часто применяется прием, называемый *рекурсивной композицией*. Он позволяет строить все более сложные элементы из простых. Рекурсивная композиция дает нам способ составить документ из простых графических элементов. Сначала мы можем линейно расположить множество символов и графики слева направо для формирования одной строки документа. Затем несколько строк можно объединить в колонку, несколько колонок – в страницу и т.д. (см. рис. 2.2.).

Данную физическую структуру можно представить, введя отдельный объект для каждого существенного элемента. К таковым относятся не только видимые элементы вроде символов и графики, но и структурные элементы – строки и колонки. В результате получается структура объекта, изображенная на рис. 2.3.

Представляя объектом каждый символ и графический элемент документа, мы обеспечиваем гибкость на самых нижних уровнях дизайна Lexi. С точки зрения отображения, форматирования и вкладывания друг в друга единообразно трактуются текст и графика. Мы сможем расширить Lexi для поддержки новых наборов

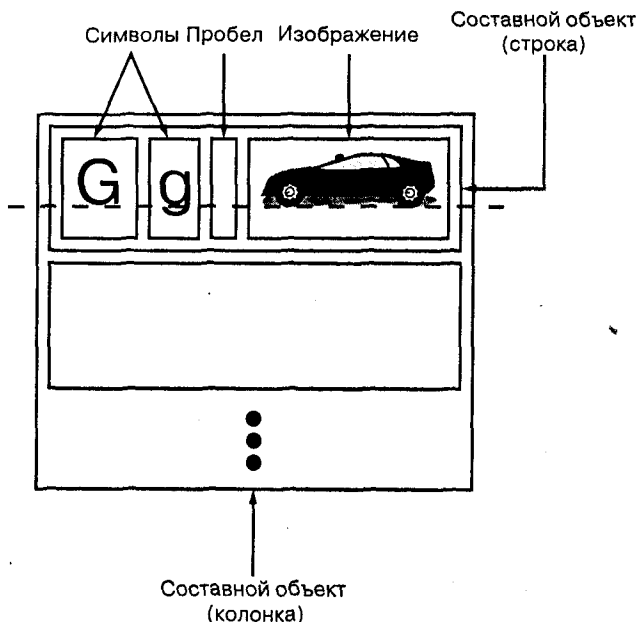


Рис. 2.2. Рекурсивная композиция текста и графики

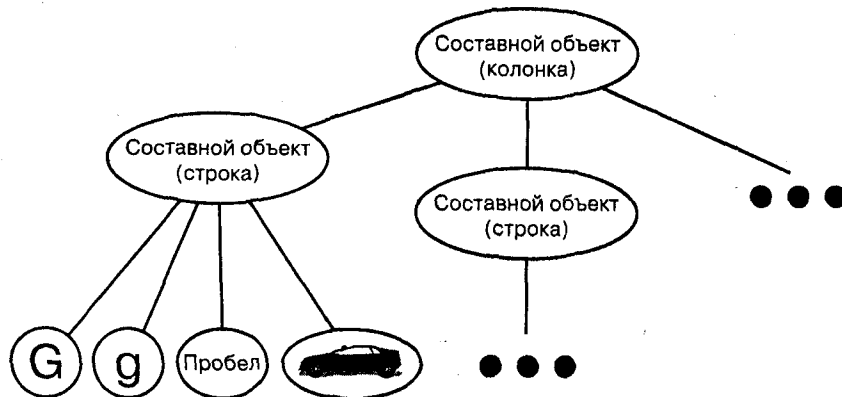


Рис. 2.3. Структура объекта для рекурсивной композиции текста и графики

символов, не затрагивая никаких других функций. Объектная структура Lexi точно отражает физическую структуру документа.

У описанного подхода есть два важных следствия. Первое очевидно: для объектов нужны соответствующие классы. Второе, менее очевидное, состоит в том, что у этих классов должны быть совместимые интерфейсы, поскольку мы хотим унифицировать работу с ними. Для обеспечения совместимости интерфейсов в таком языке, как C++, применяется наследование.