

Графические Системы. Часть II

Лекция № 2

Программирование графического пользовательского интерфейса средствами X-WINDOW

Основы программирования в системе X Window System.

АРХИТЕКТУРА X 11



X Window System — оконная система, обеспечивающая стандартные инструменты и протоколы для построения графического интерфейса пользователя. Разрабатывалась для UNIX-подобных ОС.

X Window System часто называют **X11** или просто **X**, неформально **X**. X получила своё имя как преемница оконной системы **W Window System** (в латинском алфавите буква **X** следует сразу за буквой **W**).

Система **X Window System** была разработана в Массачусетском технологическом институте (**MIT**) в 1984 году. Нынешняя (по состоянию на февраль 2009 года) версия системы — **X11R7.4**. Первая (**X11**) появилась в сентябре 1987 года. Проект **X** возглавляет фонд **X.Org Foundation**.

В **X Window System** предусмотрена *сетевая прозрачность*: графические приложения могут выполняться на другой машине в сети, а их интерфейс при этом будет передаваться по сети и отображаться на локальной машине пользователя (в случае если это разрешено в настройках).

В контексте **X Window System** термины «клиент» и «сервер» имеют непривычное для многих пользователей значение: «сервер» означает локальный дисплей пользователя (*дисплейный сервер*), а «клиент» — программу, которая этот дисплей использует (она может выполняться на удалённом компьютере).

X Window System обеспечивает базовые функции графической среды: отрисовку и перемещение окон на экране, взаимодействие с мышью и клавиатурой.



Основы программирования в системе X Window System.

АРХИТЕКТУРА X 11



X Window System дает возможность пользователю общаться со многими программами одновременно.

Чтобы вывод из них не смешивался, X создает на экране дисплея «виртуальные» подэкраны – ОКНА. Каждое приложение (**X-клиент**) рисует, как правило, в своем ОКНЕ или ОКНАХ.

X Window System не определяет деталей интерфейса пользователя— этим занимаются **менеджеры окон**, которых разработано множество. По этой причине внешний вид программ в среде **X Window System** может очень сильно различаться в зависимости от возможностей и настроек конкретного оконного менеджера.

X Window System и ресурсы программ. Как правило, X-клиенты имеют набор **конфигурационных параметров – ресурсов**. Это может быть цвет окна, тип шрифта, в общем случае – модель поведения окна в зависимости от действий пользователя.

X Window System стандартизует способ задания ресурсов приложений (**X-клиентов**) и содержит ряд процедур для работы с ними. Эта совокупность функций называется **«менеджер ресурсов» (X resource manager – Xrm)**. Хранилище параметров X-клиента называется **базой данных ресурсов (Xrdb)**.

«Управляемость событиями» (event-driver architecture). Особенностью X является то, что она организует общение между самими X-Клиентами и между X-Клиентами и внешней средой путем рассылки событий.

Событие – есть единица информации, идентифицирующая в системе изменения или действия, и содержащая дополнительные сведения о них.

Рассмотрим, как создавать приложения для работы в X, для этого – об X подробнее



Основы программирования в системе X Window System.

Подробнее – устройство X Window System. X-сервер

АРХИТЕКТУРА X 11



X Window System это совокупность программ и библиотек.

«Сердце» X – это специальная программа - **X-сервер**.

X-сервер это отдельный UNIX-процесс, запущенный на ЭВМ, к которой присоединен дисплей. Именно X-сервер знает особенности конкретного ПТК, т.е «знает», что нужно предпринять, чтобы закрасить конкретный пиксел на экране, нарисовать любой графический примитив. Он же умеет интерпретировать сигналы, полученные от клавиатуры и мыши и других устройств ввода.

Вторая программа - называется **X-клиент**, рисовать физически не умеет, но знает что надо рисовать, и умеет командовать. Команды типа: "провести линию", "открыть окно", "опросить координаты мыши", и т.п. передаются **X-серверу**, а тот их исполняет.

Один **X-сервер** может быть подключен ко многим **X-Клиентам** и может обслуживать их одновременно. На графическом экране может быть открыто сразу много окон - каждое окно порождается его собственным клиентом.

Сервер общается с **программами-клиентами**, посылая или принимая от них порции (пакеты) данных. Если сервер и клиент работают на разных машинах, то данные посылаются по сети, если же компьютер один, **то для передачи данных используется внутренний канал**. Например, если сервер обнаруживает, что нажата кнопка мыши, то он подготавливает соответствующий пакет и посылает его тому **X-Клиенту**, в чьем окне находится курсор мыши. И наоборот, если программе надо что-либо вывести на экран дисплея, то она создает необходимый пакет данных и посылает его серверу.

Состав пакетов и их последовательность определяются специальным протоколом, он называется **"X-протокол"**.



Основы программирования в системе X Window System.

Подробнее – устройство **X Window System. X-сервер**

АРХИТЕКТУРА X 11



В общем **случае X-сервер** может быть:

- Системной программой, контролирующей вывод графики на персональном компьютере;
- Приложением, отображающим графику в окно какой-то другой дисплейной системы;
- Выделенным компонентом аппаратного обеспечения.

Эта клиент-серверная терминология — **пользовательский терминал** в качестве «сервера» и **удалённые приложения** в качестве «клиентов» — зачастую запутывает новых пользователей **X Window System**, так как обычно эти термины имеют обратные значения.

Но **X Window System** принимает точку зрения программы, а не конечного пользователя аппаратуры:

- Локальный дисплей **предоставляет услуги отображения графики** программам, и потому выступает в роли сервера.
- Удалённые программы **используют** эти услуги, и потому играют роль клиентов.



Основы программирования в системе X Window System.

Подробнее – устройство X Window System. X-сервер

Протокол, с помощью которого сообщаются сервер и клиент, является прозрачным для сети: клиент и сервер могут находиться как на одной и той же машине, так и на разных. В частности, они могут работать на различных архитектурах под управлением разных операционных систем — результат будет одинаковым. **Клиент и сервер могут даже безопасно взаимодействовать через Интернет посредством туннелирования(*) соединения сквозь зашифрованный сетевой сеанс.**

Чтобы запустить удалённую клиентскую программу, выводящую графику на локальный **X-сервер**, пользователь обычно открывает эмулятор терминала и подключается к удалённой машине при помощи **telnet** (или SSH). Затем он отдаёт команду, указывающую дисплей, на который следует выводить графику (например, `export DISPLAY=[имя компьютера пользователя]:0 .0` при использовании `bash`).

Наконец, пользователь запускает клиентскую программу. Она подключится к локальному **X-серверу** и будет отображать графику на локальный экран и принимать ввод от локальных устройств ввода.

Другой вариант — использовать небольшую вспомогательную программу, которая подключается к удалённой машине и запускает на ней нужное клиентское приложение.

Использование удалённых клиентов может быть полезно, например, в таких ситуациях:

- графическое администрирование удалённой машины;
- выполнение интенсивных ресурсоёмких вычислений на удалённой UNIX-машине и вывод результатов на локальной Windows-машине;
- выполнение графических программ одновременно на нескольких машинах, с одним дисплеем, одной клавиатурой и одной мышью.

Например



Основы программирования в системе X Window System.

Подробнее – устройство X Window System. Примечание

(*) Туннелирование в компьютерных сетях — это метод построения сетей, при котором один сетевой протокол инкапсулируется в другой. От обычных многоуровневых сетевых моделей (таких как OSI или TCP/IP) туннелирование отличается тем, что инкапсулируемый протокол относится к тому же или более низкому уровню, чем используемый в качестве туннеля.

Туннелирование обычно применяется для согласования транспортных протоколов либо для создания защищённого соединения между узлами сети.

Инкапсуляция в компьютерных сетях — это метод построения модульных сетевых протоколов, при котором логически независимые функции сети абстрагируются от нижележащих механизмов путём включения или инкапсулирования в более высокоуровневые объекты.

Инкапсулирование в более высокоуровневые объекты — свойство языка программирования, позволяющее объединить данные и код в объект и скрыть реализацию объекта от пользователя. При этом пользователю предоставляется только спецификация (интерфейс) объекта. Пользователь может взаимодействовать с объектом только через этот интерфейс.

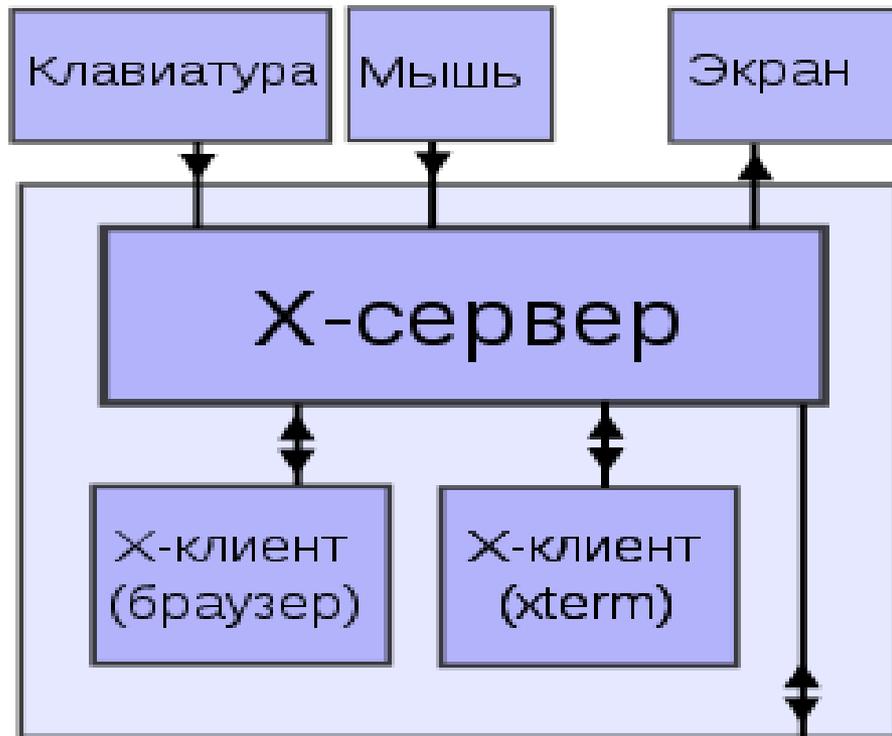
Инкапсуляция — один из четырёх важнейших механизмов объектно-ориентированного программирования (наряду с абстракцией, полиморфизмом и наследованием).



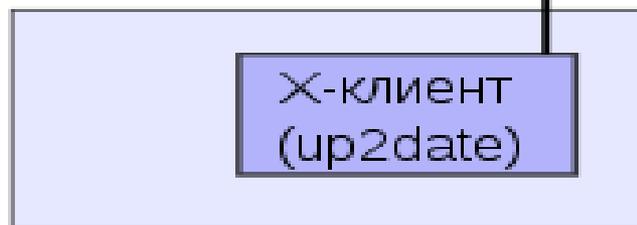
Основы программирования в системе X Window System.

Подробнее – устройство X Window System. X-сервер

Компьютер пользователя



Сеть



Удалённая машина

В этом примере X-сервер принимает ввод с клавиатуры и мыши и производит вывод на экран. На пользовательской рабочей станции выполняются **веб-браузер** и эмулятор терминала (**xterm**). Программа обновления системы (up2date) работает на удалённом сервере, но управляется с машины пользователя.

Удалённое приложение работает так же, как если бы оно выполнялось локально.

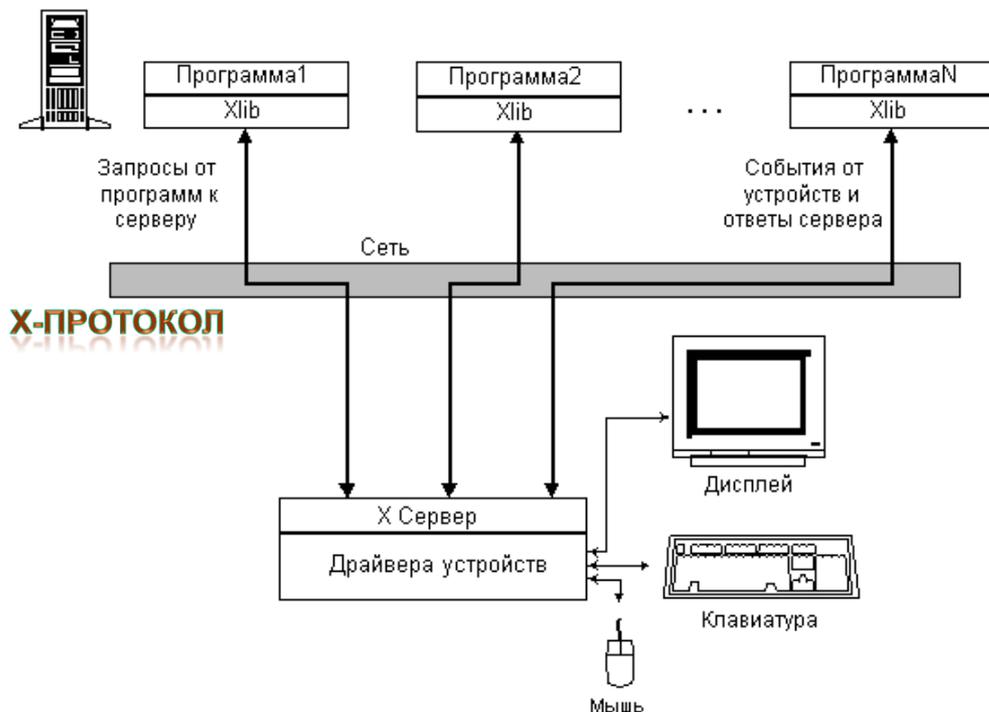


Основы программирования в системе X Window System.

Подробнее – устройство X Window System

Чтобы программировать для **X**, совсем не обязательно знать детали реализации сервера и протокола обмена. Система предоставляет **библиотеку процедур**, с помощью которых программы осуществляют доступ к услугам **X** «на высоком уровне». Так, для того, чтобы вывести на экран точку, достаточно вызвать процедуру **XDrawPoint()**, передав ей соответствующие параметры. Последняя выполняет всю черновую работу по подготовке и передаче пакетов данных серверу. Эта библиотека называется **Xlib** и находится в файле **IX11.a**, который, как правило, находится в каталоге **/usr/lib**. Прототипы функций библиотеки, используемые структуры данных, типы и прочее определяется в файлах-заголовках из директории **/usr/include/X11**.

Общая схема общения клиентов и X-сервера



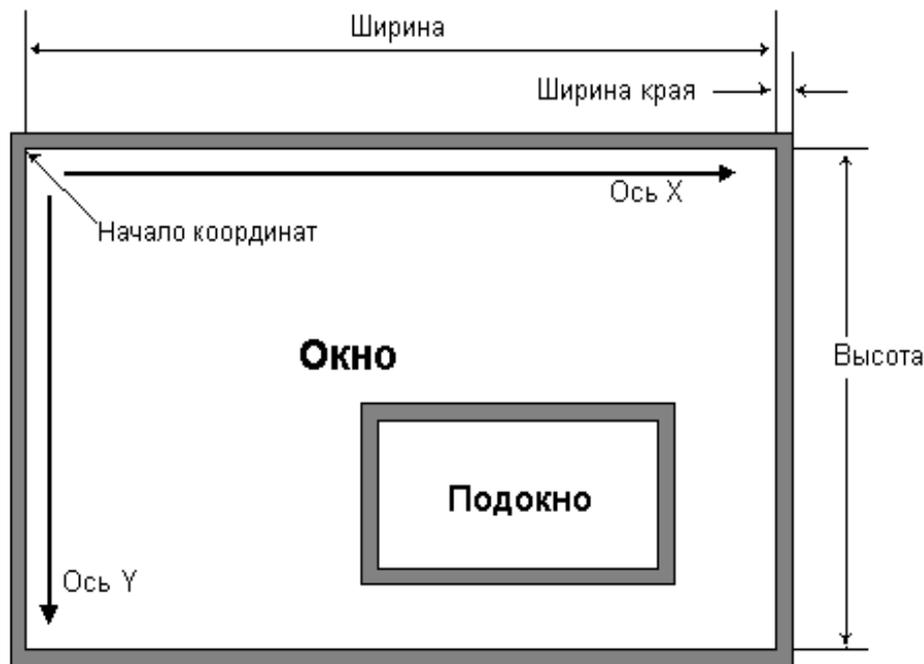
Посылка порций данных, особенно если она осуществляется по сети, операция достаточно медленная. Чтобы повысить производительность системы, **Xlib** не отправляет пакеты сразу, а буферизует их в памяти машины, на которой выполняется **X-Клиент**. Большинство запросов не требует немедленного выполнения. В буфер кладутся запросы как **X-Клиента** к серверу, так и события от сервера, находясь в буфере до тех пор, пока **X-Клиент** не прочитает их. Передача выполняется в тот момент, когда **X-Клиент** вызывает процедуру, ожидающую получения событий от сервера, например **XNextEvent()**. **X-Клиент** может явно инициировать отправку пакетов, обратившись к функциям **XFlush()** или **XSync()**.

Основы программирования в системе X Window System.

Подробнее – устройство X Window System. ОКНО

Окно - это базовое понятие в **X Window System**. Оно представляет прямоугольную область (в базовом решении) на экране, предоставляемую системой программе-клиенту, которая использует окно для вывода графической информации.

Общий вид окна X Window



Из рисунка видно, что окно имеет внутренность и край. Основными атрибутами окна являются ширина и высота внутренности, а также ширина края. Упомянутые параметры окна называются **геометрией окна**.

Окно обладает собственной системой координат. Ее начало находится в левом верхнем углу окна. Ось **X** направлена вправо, а ось **Y** – вниз. Единица измерения (в базовом решении) по обеим осям в пикселях.

Окна могут быть двух типов: **InputOutput** (для ввода-вывода) и **InputOnly** (только для ввода). Окно первого типа - это обычное окно. Окно второго типа не может использоваться для рисования. У данного окна нет края, оно "прозрачно". Окна этого типа используются достаточно редко.

X Window System позволяет программе создавать несколько окон одновременно. Окна образуют строгие иерархии на основании простого правила - в каждом экране по умолчанию существует одно уникальное окно-родитель (корневое или **root-окно**), каждое окно должно иметь **родителя** и само может быть **родителем** для других окон. Иерархия устроена подобно генеалогическому дереву, за исключением того, что только один **родитель** может создавать **потомка**.

Основы программирования в системе X Window System.

Подробнее – устройство X Window System. ОКНО

Сам сервер на каждом экране создает одно основное окно, которое является самым верхним родителем всех остальных окон. Это окно будем называть главным или **корневым (CORE)**.

Управление окнами

Окна могут располагаться на экране произвольным образом, перекрывая друг друга.

X Window System имеет набор средств, пользуясь которыми **X-Клиент** может изменять размеры окон и их положение на экране. Иными словами **X-сервер** умеет ловко манипулировать окнами, но сам никогда ничего не делает, ждёт команд от пользовательской программы.

Особенностью системы является то, что она не имеет встроенной возможности управлять окнами с помощью клавиатуры или мыши.

Чтобы это можно было осуществить, нужен специальный X-Клиент, который называется **«менеджер окон» - window manager**.

Эта самостоятельная программа, которая **отслеживает перекрытие окон, фокус, занимается изменением размера окон, их перемещением и управлением окнами с помощью клавиатуры или мыши**.

Менеджер окон X Window System — приложение, работающее «поверх» **X Window System** и определяющее интерфейс и взаимодействие с пользователем. В Unix-подобных операционных системах пользователь может выбрать любой оконный менеджер по своему усмотрению.

В стандартном дистрибутиве **X Window** есть подобная программа - это **X-Клиент twm**. Возможности этого встроенного менеджера ограничены, но, тем не менее, он позволяет осуществлять базовые действия: перемещать окна, изменять их размер и т.д.

Основы программирования в системе X Window System.

Подробнее – устройство X Window System. ОКНО

Различные менеджеры окон могут отличаться такими показателями, как:

- ❑ Возможности настройки внешнего вида и функциональности:
 - способы запуска различных приложений
 - средства настройки и управления окружением
 - наличие и взаимодействие с несколькими рабочими или виртуальными столами
- ❑ Потребление оперативной памяти и прочих ресурсов компьютера
- ❑ Степень интеграции со средой рабочего стола, предоставляющего более полный набор средств для взаимодействия с операционной средой и различными пользовательскими приложениями.

Наиболее известные менеджеры окон X Window System

AfterStep, Blackbox (минималистичный), evilwm,
Enlightenment (наиболее похожий на среду рабочего стола),
Fluxbox (лёгкий, основанный на Blackbox), FVWM, IceWM, Ion,
Kwin (ранее назывался KWM, используется в KDE),
Metacity (используется в GNOME), MWM (Motif Window Manager),
Motif Window Manager, Openbox (основан на Blackbox, используется в LXDE),
Sawfish (раньше назывался Sawmill и использовался в GNOME),
twm, Window Maker, Xfwm4 (оконный менеджер Xfce),
JWM (Joe's Windows Manager)

Основы программирования в системе X Window System.

Подробнее – устройство X Window System. ОКНО

Но **window manager** не может корректно управлять окнами, ничего о них не зная.

В одних случаях удобно иметь заголовки окон, в других – окно не может быть сделано меньше определенных размеров или – не может быть слишком увеличено.

Окно может быть минимизировано (т.е. превращено в пиктограмму), в этом случае **WM** должен знать имя пиктограммы.

Для того, чтобы сообщить **WM** свои пожелания относительно окон, **X-Клиенты** могут использовать два способа:

Во-первых, при создании окна **X Window System** могут быть переданы рекомендации (**hints**) о начальном положении окна, его ширине и высоте, минимальных и максимальных размерах и т.д.

Во-вторых, можно использовать встроенный в **X Window** способ общения между программами - **механизм свойств**.

Основы программирования в системе X Window System.

Подробнее – устройство X Window System. Свойства

Разработчики X нуждались в способе позволить клиентам общаться произвольными данными друг с другом, для этого используется механизм свойств (**properties**). *Свойство* - это **именованный набор данных**, связанная с некоторым объектом (например, окном), и которая доступна всем клиентам X. Свойства используются клиентами, чтобы хранить информацию, которая может понадобиться другим клиентам, или если необходимо знать информацию используемую другими клиентами.

Схожесть с моделью "**ресурс-идентификатор**" очевидна, но и различия весьма существенны - если "**ресурсы**" можно потенциально разделять между программами-клиентами, работающими с одним X-сервером, то "**свойства**" являются общедоступными (для клиентов одного сервера). В этом же правиле можно заметить и дополнительный нюанс - атомы, в отличие от идентификаторов ресурсов, уникальны только для одного окна - ведь одно "свойство" может быть присуще разным окнам.

Каждое свойство имеет название и уникальный компактный идентификатор - *атом*. Обычно имена свойств записываются большими буквами, например: **MY_PROPERTY**. Атомы используются для доступа к содержимому свойств с тем, чтобы уменьшить количество информации, пересылаемой по сети между клиентами и X сервером.

В X предусмотрен набор процедур, позволяющих перевести им свойства в уникальный атом, и, наоборот, по атому получить необходимые данные.

Некоторые свойства и соответствующие им атомы являются predetermined и создаются в момент инициализации сервера. Этим атомам соответствуют символические константы, определенные в файлах-заголовках библиотеки Xlib. Эти константы начинаются с префикса **XA_**, например **XA_WM_HINTS**.

Ни к "содержимому" свойств, ни к их именам X Window не предъявляет фактически никаких особых требований, но существует фиксированное множество имен и соглашения о содержимом свойств, которые необходимы для нормального функционирования любого приложения в среде X Window. Эти соглашения об именах/содержании свойств с predetermined атомами называются ICCCM - "Соглашения о взаимодействии между программами-клиентами".

Основы программирования в системе X Window System.

Подробнее – устройство **X Window System**. **Графические возможности**

Система создавалась для потенциальной поддержки любых устройств растрового графического вывода.

В подобного рода устройствах изображение представляется матрицей светящихся точек - пикселей. Каждый пиксел кодируется определенным числом бит (как правило 2, 4, 8, 16 или 24). Число бит-на-пиксель называют "**толщиной**" или **глубиной дисплея**.

Биты с одинаковыми номерами во всех пикселях образуют как бы плоскость, параллельную экрану. Ее называют **цветовой плоскостью**. **X Window** позволяет рисовать в любой цветовой плоскости (или плоскостях), не затрагивая остальные. Визуально это способ, по которому определяется воспроизведение цветной или одноцветной картинке на мониторе.

Значение пикселя не задает непосредственно цвет точки на экране. Последний определяется с помощью специального массива данных, называемого **палитрой**. Цвет есть содержимое ячейки палитры, номер которой равен значению пикселя. **X-сервер** поддерживает как разделяемую между всеми клиентами общую палитру, так и возможность создания для каждого клиента собственной виртуальной палитры.

X Window System имеет большой набор процедур, позволяющих рисовать графические примитивы – точки, линии, дуги, выводить текст и работать с областями произвольной формы.