Графические Системы. Часть II

Лекция № 3

Программирование графического пользовательского интерфейса средствами X-WINDOW.

ИПВУ. Управляющие объекты – widget'ы и их ресурсы

Основы программирования в системе X Window System.

Подробнее – устройство X Window System. ИПВУ.

X Toolkit Intrinsics (Xt)

Чтобы облегчить программирование в системе **X Window** было создано несколько инструментальных пакетов высокого уровня - **ИПВУ**. Стандартом де-факто в настоящее время стала библиотека **X Toolkit Intrinsics (Xt)**, которая входит в комплект стандартной поставки системы. **Xt** упрощает инициализацию программ и создание окон. Кроме того, библиотека содержит средства для создания объектов (управляющих элементов), используемых программами при общении с пользователями (элементы интерфейса). В терминах **Xt** управляющий элемент называется **widget**.

По сути **Xt** – это низкоуровневый набор процедур, написанных на языке С. Эти процедуры используют упомянутые **widget'ы**, которые являются предопределенными компонентами пользовательского интерфейса - **меню, кнопки, полосы прокрутки и пр.**

Можно сформулировать иначе: Виджет - это параметризуемая заготовка части пользовательского интерфейса (кнопка, часть меню, пиктограмма и т.д.), привязываемая к окну экрана терминала.

X Toolkit использует т.н. афинский набор **widget'ов** – библиотеку предопределенных элементов пользовательского интерфейса **(Xaw)**.

Xaw (X Window System Athena widget set) — набор **widget'ов** для реализации простых интерфейсов пользователя, основанный на **X Toolkit Intrinsics**. Предоставляется *X* консорциумом в комплекте стандартной поставки **X Window System**. **Athena widget set** был создан в ходе работы над базовыми приложениями и утилитами в ходе реализации проекта **проекта Athena** (MIT, Digital Equipment Corporation и IBM), результатом которого и является **X Window System.**

На настоящий момент на основе **Xt** реализованы различные наборы (библиотеки) управляющих объектов – **widget'ов**: OSF/Motif, OPEN LOOK и др. Эти наборы, в совокупности с самим **Xt**, применяются в качестве удобного инструмента для создания **GUI**. Они берут на себя ту рутинную работу, которую необходимо выполнить программисту при написании собственного приложения с использованием только процедур **X Window System**, т.е. – **Xlib и X-протокола**. Подобным же образом строятся современные, популярные ИПВУ.

Основы программирования в системе X Window System.

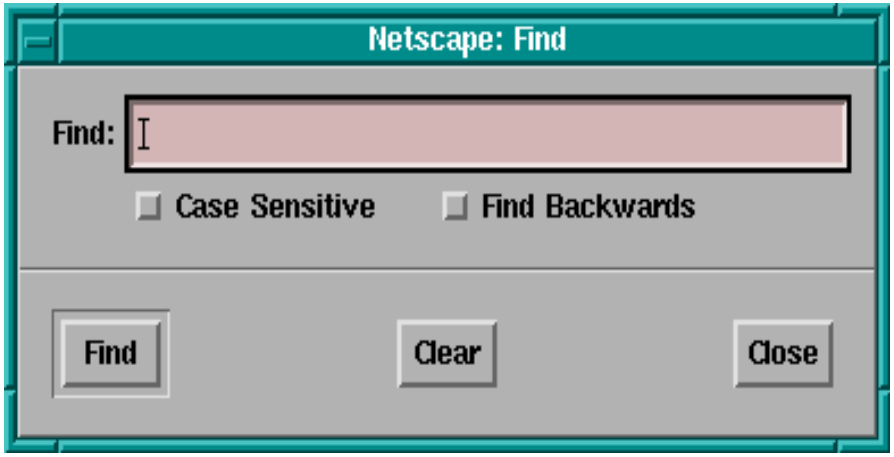
Управляющие объекты – widget'ы и их ресурсы

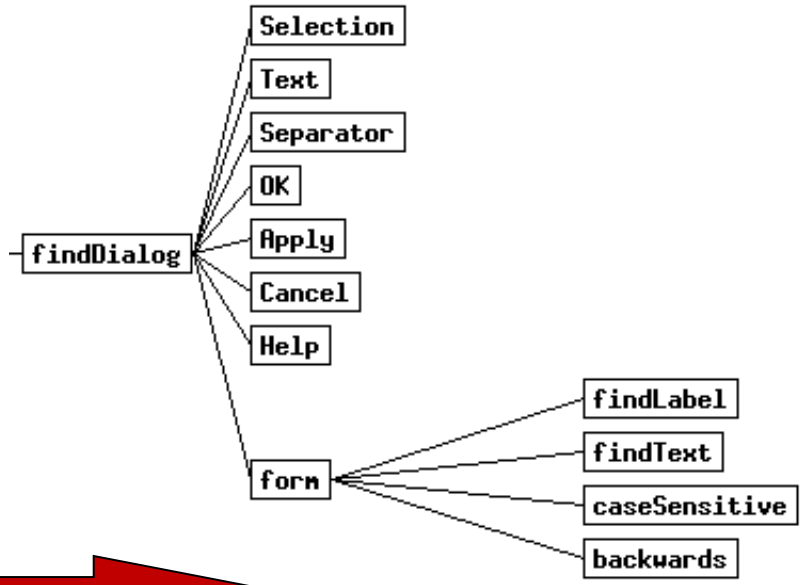
Итак, как было упомянуто ранее, ИПВУ представляет собой базу для создания т.н. управляющих элементов - **widget'ов.**

Widget - это, бесспорно, объект. Как правило, он связан с окном в смысле библиотеки **XLib**. Вообще говоря, верно и обратное: каждое окно при программировании с участием **XToolkit** связано с некоторым **widget'ом** (одним).

Цель **widget'ов** - контроль за своими окнами, их правильное отображение и, что самое главное, поддержка определенного поведения окна, т.е. стиля реакции элементов окна на события, происходящие в сеансе работы с **X Window**. **Widget'ы** служат более или менее автономными кирпичиками, из которых составляется интерфейс почти всех **X-Клиентов**. Наиболее очевидные примеры: кнопки, полосы прокрутки (Scroll Bars), окна диалога и многие другие.

Весь интерфейс программ в **X-Window** состоит из объектов- **widget'ов**. Например, диалоговое окно поиска в программе Netscape содержит **widget'ы:** строка ввода, два переключателя, полосу-разделитель и три кнопки:





Дерево объектов окна «**Find**» программы **Netscape**

Основы программирования в системе X Window System.
Управляющие объекты – widget'ы и их ресурсы

В смысле Xt (все рассматриваем на примере этого ИПВУ) – это просто структура, поля которой включают идентификатор самого элемента, идентификатор его окна, если таковое имеется, и многое другое.

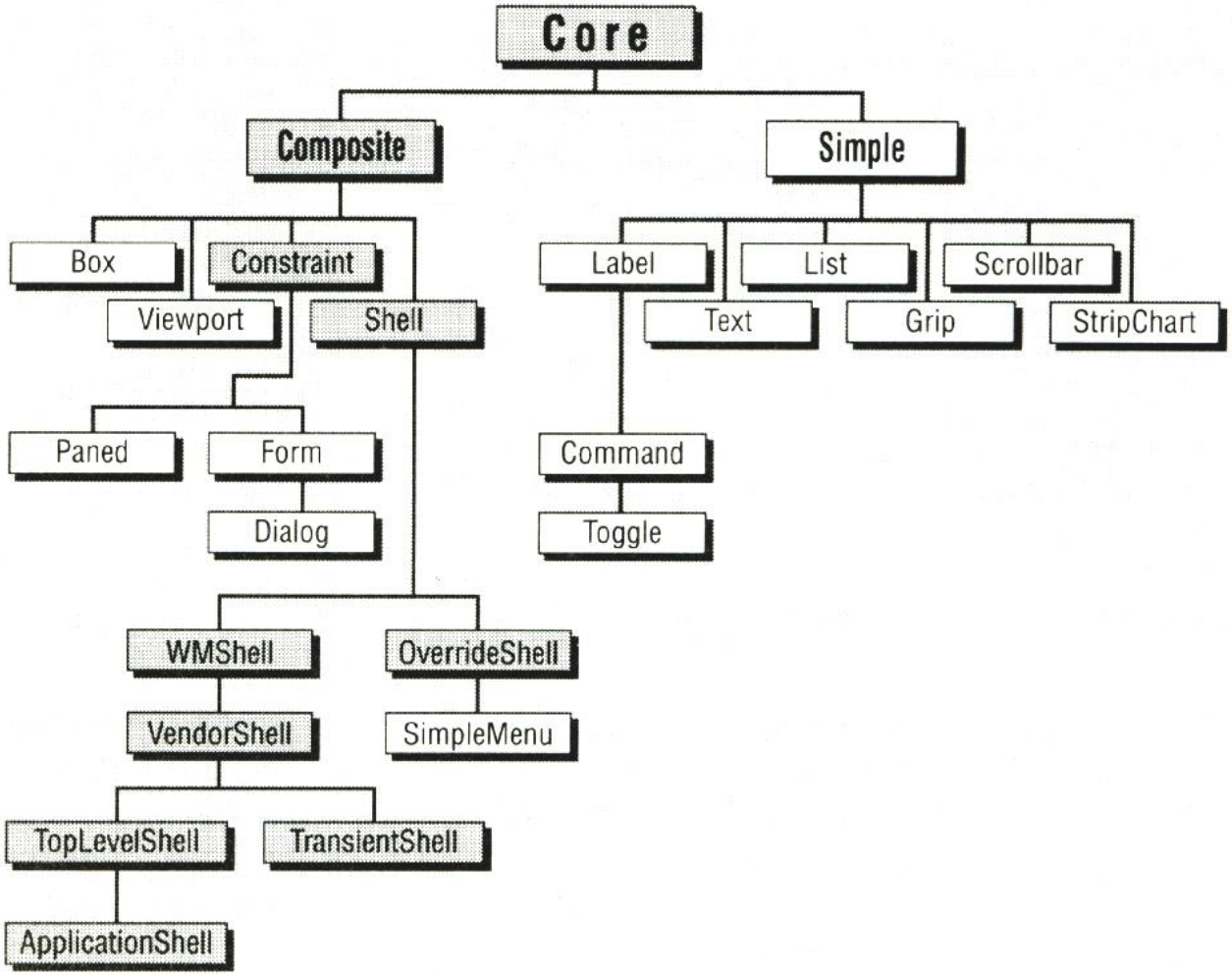
Структура объектов-widget'ов практически совпадает со структурой их окон, вложенных в друг друга. Естественно, первое существенное деление widget'ов - на простые и составные. В отличие от простых, составные widget'ы в состоянии управлять своими потомками. В классической схеме объектно-ориентированного программирования эта иерархия так и называется - иерархия объектов.

Далее, widget'ы делятся по типам: кнопки - это один тип, полосы прокрутки - другой... Но точно ли кнопки - это один тип, а не несколько, очень похожих один на другой, но все же различающихся? Или точно ли при реализации widget'ов не приходится отлаживать одни и те же механизмы? Экономить усилия помогает иерархия классов. Действительно, вместо того чтобы создавать класс объектов с нуля, разумнее и быстрее взять уже существующий класс объектов, сколько-нибудь похожий на требуемый, и модифицировать его.

Основы программирования в системе X Window System.
Управляющие объекты – widget'ы и их ресурсы

Итак, получаем – каждый объект-widget принадлежит к одному из предопределенных классов (widget-класс). Класс можно рассматривать как множество экземпляров (объектов-widget'ов), имеющих одинаковые характеристики. Классы образуют иерархию.

Такая иерархия была разработана в проекте Athena, посмотрим на нее – в первый раз.



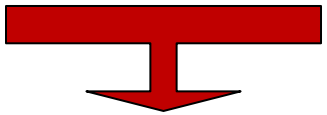
Основы программирования в системе X Window System.

Управляющие объекты – widget'ы и их ресурсы

Гибкому использованию иерархии классов **widget'ов** способствует и их параметризация. Не следует заводить пару **widget'ов** "красная кнопка" и "зеленая кнопка", если можно завести один **widget** "кнопка" с параметром "цвет" (тем более, в дальнейшем разработчику понадобится еще и "желтая кнопка"). Поэтому **widget** должен быть достаточно универсальным: у него должно быть много параметров, списки которых отличаются от **widget'а** к **widget'у**, и, в зависимости от параметров, объект - **widget** должен рисоваться и реагировать на события по-разному.
Такие (поименованные) параметры или атрибуты **widget'ов** называются **ресурсами**. Ресурсами **widget'ов** могут быть, например, цвет фона его окна, шрифт выводимого текста, цвет границы окна; механизм ресурсов может быть использован для определения модели поведения **widget'а** в зависимости от происходящих событий. Ресурс – это системная переменная, связанная с той или иной характеристикой окна.

В X реализован оригинальный способ конфигурирования ресурсов и наделения **widget'ов X-Клиента** необходимыми ему ресурсами. (Будет рассмотрен ниже)

Теперь можно сделать попытку дать короткое определение понятия **«widget»**:



Это объект, связанный с окном X Window, имеющий строго определенный список именованных параметров, называемых **ресурсами**,
и выполняющий, в соответствии со значениями этих параметров, строго определенные действия, в основном, по поддержке интерфейса приложения в своем окне.

Объекты- **widget'ы** образуют иерархию вложенных друг в друга (и, соответственно, подчиненных) объектов;

Классы **widget'ов** образуют иерархию в соответствии с вложенными друг в друга списками ресурсов и реализацией методов производных **widget'ов** с использованием методов базовых **widget'ов**.

Основы программирования в системе X Window System.

Управляющие объекты – widget'ы и их ресурсы

Во время работы программа X-Клиент создает сами объекты- widget'ы (экземпляры widget-классов). Они образуют совокупности, каждая из которых также представляет собой некоторую иерархию. Каждая такая иерархия называется деревом объектов – widget tree.

Widget-tree X-Клиента xcalc.


```
XCalc xcalc
                Form  ti  or  rpn        (the name depends on the mode)
                    Form  bevel
                        Form  screen
                            Label  M      (the memory indicator on the screen)
                            Toggle LCD    (where the data is displayed)
                            Label  INV    (the inverted indicator on the display)
                            Label  DEG    (the degrees indicator on the display)
                            Label  RAD    (the radians indicator on the display)
                            Label  GRAD   (the gradians indicator on the display)
                            Label  P      (the Parenthesis indicator on the display)
                    Command  button1      (the actual calculator buttons)
                    Command  button2      (buttons are numbered from right to left)
                    Command  button3      (See the app-defaults file for associations
                                           and so on...)
                    Command  button38     (between widget names and default labels)
                    Command  button39
                    Command  button40     (Only 39 buttons in HP mode)
```

Основы программирования в системе X Window System.

Управляющие объекты – widget'ы и их ресурсы

Корнем дерева **Widget-tree** обязательно является **widget**, принадлежащий к одному из подклассов специального класса – **Shell** (вспомним схему). Такие объекты часто называют **shell-widget**.

Если среди двух **widget'ов** A и B дерева объектов первый ближе к корню, чем второй, то A является родительским объектом ("родителем") для B, а B - подобъектом (или "дочерним" объектом (потомком)) для A.

Таким образом, **shell-widget** является родительским **widget'ом** для всех остальных **widget'ов** данного дерева объектов (**Widget-tree**).

!!Именно он осуществляет взаимодействие программы и менеджера окон!!

Описанная иерархия **widget'ов** соответствует взаимосвязи их окон, что является свойством **X Window**. В этой связи для любого **X-Клиента** можно говорить о **window-tree**.

Кроме этого, на объекты- **widget'ы** накладывается и другая иерархия. Дело в том, что во время работы одни объекты могут управлять другими. Например, если некий объект имеет подобъекты, то при изменении геометрии он может автоматически перестроить геометрии своих потомков. Чтобы это могло осуществиться, между **widget'ами** устанавливается связь - связь "по управлению". Каждый объект может иметь один или несколько "управляемых" **(managed)** им подобъектов.

Теперь рассмотрим, как в программа, использующая **Xt**, заимодействует с **widget'ами** и **X Window**. Предусмотрены три механизма.

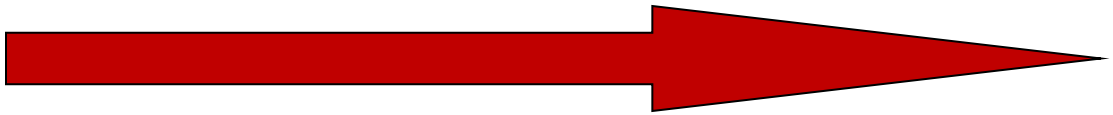
Основы программирования в системе X Window System.
Управляющие объекты – widget'ы и их ресурсы

Callback - процедуры ("процедуры обратного вызова"). Для любого класса определена совокупность действий, на которые должны реагировать принадлежащие ему объекты. Например, для любого класса предусмотрена реакция на уничтожение **widget'a**. Когда действие производится, происходит вызов либо стандартной функции **Xt**, либо одной или нескольких процедур, предоставляемых программой. Такие функции и называются **callback** - процедурами или просто **callback**.

Action -процедуры. Программа может заказать реакцию на то или иное сложное событие (группу событий), приходящее от **Xt**. Если событие происходит, **Xt** осуществляет поиск и вызов соответствующей функции.

Event handlers - обработчики событий. Этот способ аналогичен предыдущему, но более быстр и менее гибок. Он позволяет реагировать только на простые (единичные) события, но не на их последовательности.

Подробнее эти механизмы будут рассмотрены позже.



Прежде чем переходить к подробному рассмотрению иерархии классов **widget'ов**, остановимся подробнее на механизме **ресурсов**.