

Пирогова М.А.

Сборник лабораторных работ

по курсу

ГРАФИЧЕСКИЕ СИСТЕМЫ

(магистры)

Оглавление

Лабораторная работа № 4. X Window System. Ресурсы. Управление ресурсами стандартных X-клиентов в X11 в сетевых структурах - ОС.....	3
Лабораторная работа № 5. Программирование GUI с использованием ИПВУ Tcl/Tk.....	8
Лабораторная работа № 6. Программирование GUI для подзадачи отображения проекта в гипотетической системе PLM с использованием ИПВУ Tcl/Tk.....	11

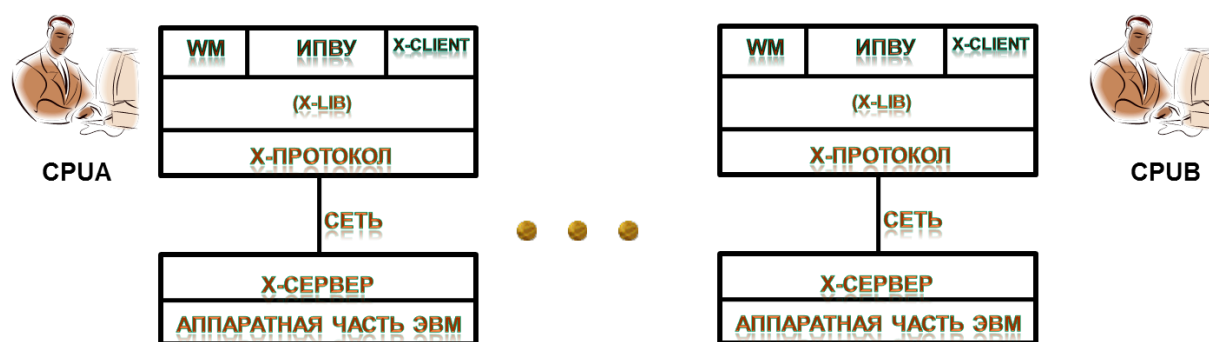
Лабораторная работа № 4. X Window System. Ресурсы. Управление ресурсами стандартных X-клиентов в X11 в сетевых структурах - ОС

Цель лабораторной работы

Целью данной работы является дальнейшее изучение стандартных для X11 способов управления X-Клиентами в распределенной, гетерогенной Вычислительной системе – Открытой системе, т.е. – изучение способов и средств запуска локального приложения (X-клиента) на удаленной машине и запуск удаленного **X-Клиента** на локальной машине.

Пояснения к лабораторному заданию

Запуск локального X-Клиента на удаленной машине



Вспомним, что все **X-Клиенты** унифицированы на сетевом уровне (**X-протокол**). При запуске **X-Клиента(CPUA)** на машине **CPUB** его отображением должен заниматься **X-Сервер CPUB**.

Для реализации поставленной задачи необходимо задействовать специальный ресурс **СЕТЕВОГО X-Протокола: -display или (-d)**, а также специальную **X-Утилиту xhost**, с помощью которой может быть изменен список машин (хостов), которым разрешено пользоваться X-Сервером локальной машины для отображения **X-Клиентов**.

X-сервер по умолчанию имеет возможность обрабатывать графические приложения, оформленные в соответствии со спецификацией **X11**, и запущенные на локальной машине, или графические приложения, запущенные на хостах, занесенных в список, размещенный в файле **/etc/Xn.hosts** (где **n** - номер дисплея сервера) локальной машины.

Значение ресурса **-display** или **(-d)** строится по следующему закону:



-display [host]:server[.screen]

Имя/IP-адрес удаленной машины, на которой воспроизводится окно локального X-Клиента: CPUB

№ X-Сервера, обрабатывающего воспроизведение окна локального X-Клиента: № X-Сервера CPUB; Т.к. мы не рассматриваем мультисерверные хосты, у нас № X-Сервера всегда = 0

№ Экрана дисплея машины, воспроизводящей окно локального X-Клиента: № Экрана CPUB; у нас № Экрана = 0, т.к. мы не рассматриваем общий случай, когда к одному Серверу м.б. подключено несколько экранов

CPUA\$>xlogo -d CPUB:0.0

Значение ресурса **-display** или **(-d)** очень важно для работы в Открытых Системах и поэтому для его именования, сохранения и передачи может быть использована специальная системная переменная : **DISPLAY**.

Установка:

C-shell: %>setenv DISPLAY CPUB:0.0

Born-shell: \$> DISPLAY = CPUB:0.0

 \$> export DISPLAY

Запуск удаленного X-Клиента на локальной машине

Производится в два этапа:

- Разрешение доступа к локальному X-Серверу для воспроизведения удаленного X-Клиента:

CPUA \$>xhost + [Для всех доступных по сети машин] или

CPUA \$>xhost + CPUB [Для конкретной машины]

- Собственно запуск удаленного X-Клиента:

CPUA\$>telnet CPUB

CPUB%>login user1

CPUB%> Password user01

CPUB%>xlogo -d CPUA:0.0

Домашняя подготовка

Выполнение пунктов лабораторной работы потребует изучения теоретических вопросов, находящихся в документе [Lecture_to_LAB02_03.pdf](#).

Лабораторное задание

1. Зарегистрироваться, войти в систему, стартовать оконную систему.
2. Найти сведения о возможно доступных рабочих станциях в сегменте сети.

Для этого проанализировать файл **/etc/hosts**, где перечислены **IP-адреса** и имена возможно подключенных по сети рабочих станций.

Для анализа реально подключенных машин в сети использовать команду **ping**;

Для справки о возможностях этой команды запустить **man ping**;

Для поиска пути к команде **ping** используйте утилиту **UNIX - find**, например:

```
...%>find / -name ping -print
```

Для автоматизации процесса подключения можно составить скрипт, в цикле проверяющий "существование" подключенных рабочих станций сегмента сети.

Учитывать при этом, что машины одного сегмента имеют одинаковы первые 3 поля **IP** - адреса и различаются в последнем.

3. Создать в каталоге **\$HOME** подкаталог **MYGS**. В этом подкаталоге должен быть только один скрипт-файл (например, **xvasja**), содержащий нестандартный, пользовательский X - клиент с определенным набором заранее заданных ресурсов. (В качестве этих нестандартных X-клиентов использовать видоизмененные стандартные, взятые из задания на курсовую работу членов бригады).

Удостовериться в работоспособности этого X-клиента.

4. Запустить на удаленной машине несколько стандартных X-клиентов (**xload**, **xlogo**, ...).

Преодолев, объяснить возникшие конфликтные ситуации.

5. Запустить **xvasja** на удаленной машине. Применяемая опция - **display**. Преодолев, объяснить возникшие конфликтные ситуации.

6. Запуск удаленного **X-клиента** на локальной машине.

6.1 Используя сессию **telnet**, на удаленной машине в подкаталоге **\$HOME/GUEST** создать еще один нестандартный клиент (например, **xguest** - см. пункт 3.)

6.2 Разрешить оконному менеджеру локальной машины производить отображение удаленных клиентов, используя утилиту **xhost** (если это не было сделано ранее).

6.3 Осуществить отображение удаленного X-клиента на локальной машине, используя установку системной переменной **DISPLAY**.

Примечание

При выполнении пунктов, связанных с запуском локального X-клиента на удаленной машине и удаленного X-клиента на локальной машине обратить внимание на следующую утилиту:

xhost- программа управления доступом к **X-серверу**.

Используется для изменения списка хостов, которым разрешено пользоваться **X-сервером** локальной машины в открытой системе.

Таким образом, обеспечивается элементарная защита локальных данных, а также контроль доступа к данным.

X-сервер по умолчанию имеет возможность обрабатывать графические приложения, оформленные в соответствии со спецификацией **X11**, и запущенные на локальной машине или графические приложения, запущенные на хостах, занесенных в список, размещенный в файле **/etc/Xn.hosts** (где **n** - номер дисплея сервера) локальной машины.

Утилита **xhost** обычно запускается из стартового файла или, в случае необходимости, запускается в интерактивном режиме локальным пользователем - для получения доступа к графическим ресурсам других хостов или предоставления им своих графических ресурсов.

Утилита **xhost** имеет следующие опции:

[+]hostname

здесь **hostname** - указанный номер хоста, который добавляется в список имеющих доступ к локальному X-серверу. Знак + необязателен.

-hostname

здесь **hostname** - указанный номер хоста, который удаляется из списка имеющих доступ к локальному **X-серверу**. Знак - обязателен.

+

разрешает доступ к локальному **X-серверу** всем физически

подсоединенным машинам, даже если они отсутствуют в списке имеющих доступ.

-

доступ к локальному **X-серверу** от других хостов запрещен.

Контрольные вопросы

1. Стандартные средства Операционной Системы для проверки соединений в сетях на основе **TCP/IP**. Утилита **ping**.
2. Специальные ресурсы X11 для настроек при работе в сети.
3. Запуск локального X-клиента на удаленной машине.
4. Запуск удаленного X-клиента на локальной машине.
5. Утилиты **TCP/IP** в сочетании с ресурсом **display** для организации запуска X-клиентов в Открытых Системах.
6. Способы объявления системных переменных Операционной системы для обеспечения работы с X-клиентами.

Лабораторная работа № 5. Программирование GUI с использованием ИПВУ Tcl/Tk.

Цель лабораторной работы

Целью данной работы является изучение инструментальных средств программирования графических пользовательских интерфейсов приложений (GUI), построенных в соответствии с концепциями **X Window System**. Лабораторное задание выполняется с помощью конструкций ИПВУ Tcl/Tk.

Пояснения к лабораторному заданию

При подготовке заданий, которые необходимо выполнить в процессе данной лабораторной работы, следует обратить особое внимание на наиболее полное использование базовых widget'ов, доступных программисту в Tcl/Tk:

Widget'ы	Описание
toplevel	Контейнер для других Widget'ов , управляется непосредственно WM
frame	Контейнер для других Widget'ов
label	Текстовая метка
entry	Окно для ввода текстовых данных
message	Отображает текстовое сообщение, возможно состоящее из нескольких строк
text	Сложный Widget для ввода и редактирования текста
checkboxbutton	Отображает сообщение, ассоциированное с переменной, имеющей два значения
radiobutton	Отображает сообщение, позволяет выбрать опцию из предопределенного набора
scale	«шкала», позволяет легко изменять действительные значения
listbox	Отображает список элементов для выбора
scrollbar	Вертикальная и горизонтальная прокрутка, может быть связана с одним или более Widget'ами
canvas	Отображает фигуры и графики

Кроме того, особое внимание следует уделить освоению имеющихся в Tcl/Tk диспетчеров компоновки для реализации максимально удобного и дружелюбного интерфейса пользователя:

pack	Geometry manager : упаковывает компоненты - Widget'ы вокруг границ полости родительского окна (caviti - пространство для размещения компонентов)
grid	Geometry manager : упаковывает компоненты - Widget'ы «по сетке»
place	Geometry manager : жестко фиксирует компоненты - Widget'ы в полости родительского окна или размещает их в соответствии с принципом « rubber-sheet »

Домашняя подготовка

Выполнение пунктов лабораторной работы потребует изучения ряда теоретических вопросов, а также основ языка **Tcl/Tk**. Вся необходимая информация размещена по в разделе «**Материалы по языку Tcl/Tk**» следующему адресу:

<http://a0601.narod.ru/TclTk.htm>

Лабораторное задание

1. Зарегистрироваться в системе, стартовать оконный интерфейс.
2. Определить наличие в файловой системе командной оболочки **TCL/TK - wish**. Сохранить полный путь к **wish** первым в списке путей поиска.
3. Создать шаблон **wish**-скрипта для следующих пунктов лабораторного задания:

```
#!/usr/bin/wish -f

set tk_strictMotif 1

toplevel .vasja
```

4. Создать несколько **widgets** типа **label** с различными значениями ресурсов **text**, **background**, **foreground**, **relief**, **font**. Расположить их различными способами (опция **-side** команды **pack**).
5. Создать **widget** типа **entry** с видимой областью ввода и заданной шириной.
6. Создать набор **widgets** типа **label** и **entry** для контроля вводимых переменных. Пару расположить горизонтально, вертикально.
7. Создать **widget** типа **scale** для ввода целочисленной переменной, значение которой лежит в диапазоне 0 -100, с шагом 10. Ориентировать **widget** горизонтально. Начальное значение слайдера - 40.
8. Создать **widget** типа **button** заданной ширины для запуска - при нажатии на нее - одного из известных вам X-клиентов.
9. Используя **widgets** типа **frame**, **scale**, **button**, (возможно **-radiobutton**, **checkbutton**), **label**, **entry**, разработать пользовательское X-окно для работы с аудио-аппаратурой одной из известных фирм.

Пояснение:

На панели управления обязательно должны быть отображены регуляторы правого/левого канала записи, правого/левого каналов воспроизведения, кнопки выбора режимов запись/воспроизведение, индикатор состояния (запись/воспроизведение), кнопка включения, окно для ввода номера выбираемой для записи/прослушивания композиции.

Контрольные вопросы

1. Какова структура ИПВУ **Tcl/Tk**.
2. Библиотека параметризованных заготовок графического интерфейса (widget'ов) и командный язык инструментальных средств.
3. Основные преимущества разработчиков GUI – пользователей **Tcl/Tk**.
4. Интерпретаторы **Tcl/Tk**.
5. Основные синтаксические конструкции языка.
6. Основные классы widget'ов **Tcl/Tk**. Назначение, связь с иерархией widget'ов X11.
7. Воспроизведение widget'ов в рамках проектируемого GUI. Диспетчеры компоновки – геометрические менеджеры **Tcl/Tk**.
8. Контейнеры для widget'ов. Способы создания т.н. вложенных интерфейсов – группирование widget'ов.

Лабораторная работа № 6. Программирование GUI для подзадачи отображения проекта в гипотетической системе PLM с использованием ИПВУ Tcl/Tk

Цель лабораторной работы

Целью данной работы является дальнейшее изучение инструментальных средств программирования графических пользовательских интерфейсов приложений (GUI), построенных в соответствии с концепциями **X Window System** и получение навыков применения изученных инструментальных средств для разработки GUI приложений подсистем PLM. Лабораторное задание выполняется с помощью конструкций ИПВУ Tcl/Tk.

Пояснения к лабораторному заданию

Данная лабораторная работа выполняется на стыке курсов «Графические Системы» и «Промышленная Логистика»: при подготовке к лабораторному заданию данной лабораторной работы следует обратить особое внимание на базовые понятия подсистемы **Project Management** системы **PLM**, а именно «Этап жизненного цикла изделия», «Фаза», «Шлюз или Переход», «Диаграмма Гантта», «Состав работ этапа».

Разработка GUI подсистем **Project Management** систем **PLM** заключается в создании оконных приложений, предназначенных для ввода значений каждого из базовых понятий подсистемы **Project Management** систем **PLM** или отображения полученных значений для каждого из базовых понятий подсистемы **Project Management** систем **PLM**.

Задания к лабораторной работе составлены таким образом, что возможна взаимная проверка полученных решений: результаты применения GUI для ввода значений могут быть непосредственно использованы для работы GUI отображения. Так результаты применения GUI, разработанного в ходе выполнения работы бригадами №№ 2,6,10 могут проверяться за счёт применения GUI, разработанного бригадами №№ 1,5,9, а результаты применения GUI, разработанного в ходе выполнения работы бригадами №№ 4,8,12 могут проверяться за счёт применения GUI, разработанного бригадами №№ 3,7,11.

Домашняя подготовка

Выполнение пунктов лабораторной работы потребует ознакомления с общепринятыми в среде пользователей подсистем **Project Management** систем **PLM** элементами графического отображения состояния проекта, выполнимости работ каждого из этапов проекта и представления план-графиков проектов. Ознакомление проводится в рамках курса «Промышленная Логистика». Вся необходимая информация размещена в файле **Lecture-4-LR4.pdf**.

Оформление GUI в соответствии с заданием на лабораторную работу потребует изучения ряда теоретических вопросов, а также конструкций языка Tcl/Tk. Вся необходимая информация размещена в разделе «Материалы по языку Tcl/Tk» по следующему адресу:

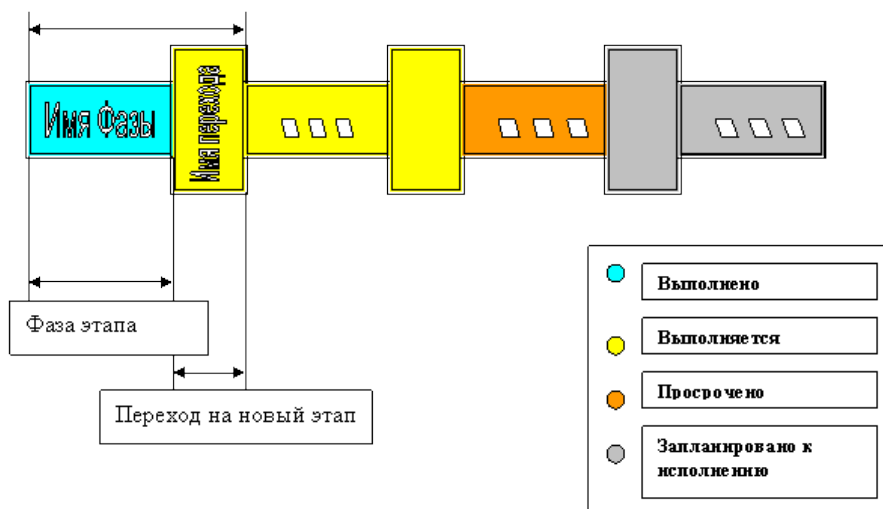
<http://a0601.narod.ru/TclTk.htm>

Варианты Лабораторных заданий:

1. Задание для бригад №№ 1, 5, 9

Разработать оконный интерфейс для подзадачи отображения проекта в гипотетической **Project Management**.

1. Выбор конкретного проекта – из списка доступных имён проектов. Список имён проектов графически доступен как содержимое **listbox**.
2. При выборе конкретного имени проекта из **listbox** списка доступных имён проектов отображаются этапы выбранного проекта. Каждый элемент отображения (представленный, например, в виде **button**) содержит имя этапа (фазы, перехода).
3. При активизации каждого из элементов отображения (путём считывания файла проекта) элемент отображения прокрашивается соответствующим цветом, отображающим степень выполнения этапа, соответствующего активизированному элементу. На рабочем поле отображается цветовая «легенда» для расшифровки покрашенной последовательности этапов.



Примечание: файл проектов – текстовый файл, каждая из K строк которого имеет вид:

P_k;X₁;P₁;Y₁;Q₁; X₁;P₁;Y₁;Q₁... X_N;P_N;<END>

Где

P_k – имя проекта, k={1,K}

X_i- имя фазы этапа, string или char[]

Y_i- имя перехода на следующий этап, string или char[]

P_i- процент выполнения работ фазы этапа, int , {-1,100}

Q_i- процент выполнения работ перехода, int , {-1,100}

«Выполнено», если процент выполнения = 100

«Просрочено», если процент выполнения = -1

«Запланировано к исполнению», если процент исполнения =0

«Выполняется», если процент исполнения ={1,99}

Необходимые пояснения для выполнения задания находятся в файле
[Explntn-4-LR401.pdf](#).

2. Задание для бригад №№ 2, 6, 10

Разработать оконный интерфейс для подзадачи отображения контроля исполнения проекта в гипотетической **Project Management**.

1. Выбор конкретного проекта – из списка доступных имён проектов. Список имён проектов графически доступен как содержимое **listbox**.
2. При выборе конкретного имени проекта из **listbox** списка доступных имён проектов отображаются этапы выбранного проекта. Каждый элемент отображения (представленный, например, в виде **button**) содержит имя этапа (фазы, перехода).

При активизации каждого из этапов выбранного проекта пользователю предоставляется возможность задать наименование фазы активизированного этапа, процент выполненных работ фазы, наименование перехода от активизированного этапа к следующему (если таковой есть), процент выполнения текущих работ по переходу. В результате в файловой системе должен появиться текстовый файл, каждая строка которого имеет вид:

P_k;X₁;P₁;Y₁;Q₁; X₁;P₁;Y₁;Q₁... X_N;P_N;<END>

где

Pk – Имя проекта

Xi- имя фазы этапа, string или char[]

Yi- имя перехода на следующий этап, string или char[]

Pi- процент выполнения работ фазы этапа, int , {-1,100}

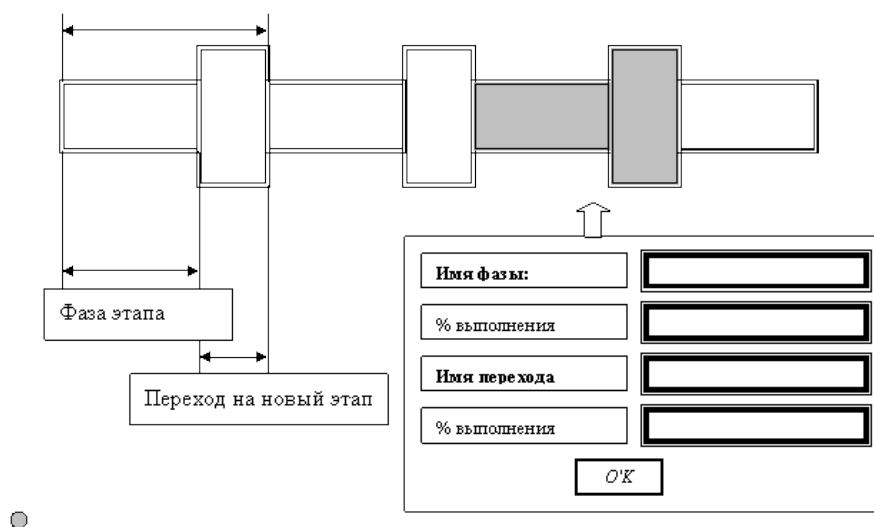
Qi- процент выполнения работ перехода, int , {-1,100}

«Выполнено», если процент выполнения = 100

«Просрочено», если процент выполнения = -1

«Запланировано к исполнению», если процент исполнения =0

«Выполняется», если процент исполнения ={1,99}



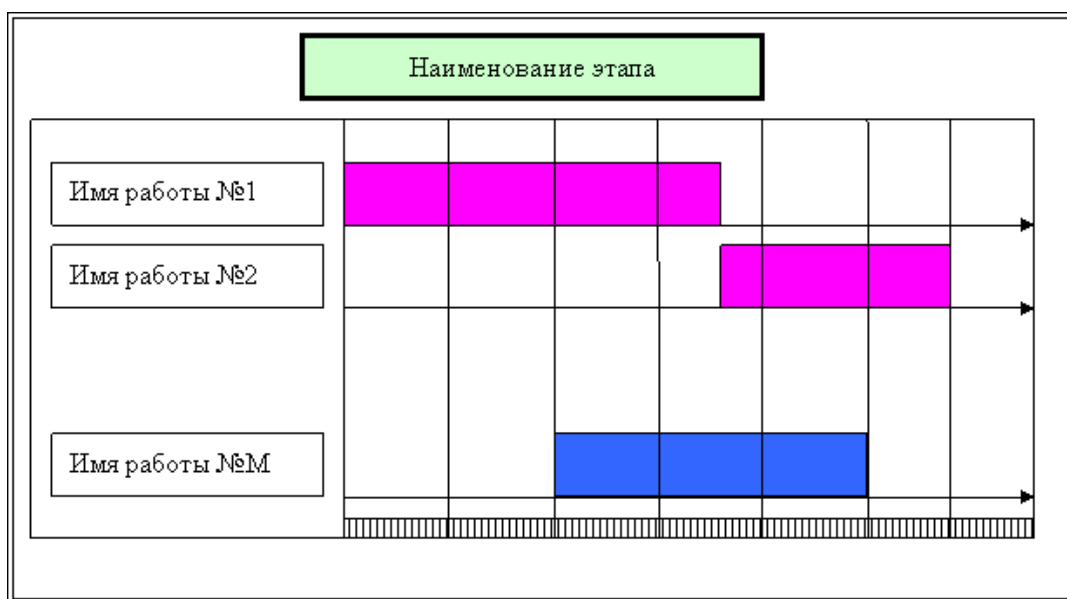
Примечание: изначально существует файл имен проектов, каждая строка которого состоит из строки символов Pk. ($k=\{1,K\}$)

Необходимые пояснения для выполнения задания находятся в файле [Explntn-4-LR402.pdf](#).

3. Задание для бригад №№ 3, 7, 11

Разработать оконный интерфейс для подзадачи отображения диаграммы Гантта по работам любого из этапов некоторого проекта в гипотетической **Project Management**.

1. Выбор конкретного этапа – из списка доступных имён этапов проекта. Список имён этапов проекта графически доступен как содержимое **listbox**.
2. При выборе конкретного имени этапа из **listbox** списка доступных имён этапов отображаются работы выбранного этапа.
3. Каждый элемент отображения, соответствующий отображаемой работе, имеет горизонтальную протяжённость, пропорциональную запланированной длительности работы. В левой части зоны отображения располагается соответствующее данному элементу отображения наименование (имя работы).
4. Взаимное расположение элементов отображения соответствует запланированной календарной последовательности выполнения работ.
5. Работы фазы отображаются одним цветом, перехода – другим.



Примечание: все необходимые данные считываются из файла работ этапов проекта – текстовый файл, каждая из R строк которого имеет вид:

Sr;A1X1;DA1;XD1;A2X1; DA2;XD2;A3X1; DA3;XD3... AMX1; DAM;XDM<END>

Или

Sr;A1Y1; DA1;YD1;A2Y1; DA2;YD2 ... AMY1; DAM;YDM<END>

Где

Sr – имя фазы или имя перехода этапа проекта, $r=\{1,R\}$

AmXi- имя работы фазы выбранного этапа проекта, string или char[]

AmYi- имя работы перехода (если он есть) выбранного этапа проекта, string или char[]

Dam – абсолютный номер дня начала работы, int

XDm или YDm – продолжительность работы, int

Необходимые пояснения для выполнения задания находятся в файле

[Explntn-4-LR403.pdf](#) .

4. Задание для бригад №№ 4, 8, 12

Разработать оконный интерфейс для подзадачи создания диаграммы Ганта по работам любого из этапов некоторого проекта в гипотетической **Project Management**.

Выбор конкретного этапа – из списка доступных имён этапов проекта. Список имён этапов проекта графически доступен как содержимое **listbox**. Список имён этапов проекта изначально существует, каждая строка такого файла состоит из строки символов $Sr, r=\{1,R\}$.

1. При выборе конкретного имени этапа из **listbox** списка доступных имён этапов отображается формуляр графического интерфейса ввода информации для диаграммы Ганта.
2. Пользователь в формуляре для каждой работы фазы заполняет поля «Имя работы», «Номер дня начала», «Длительность работы в днях»
3. Пользователь в формуляре для каждой работы перехода (если этап – не последний) заполняет поля «Наименование работы», «Номер дня начала», «Длительность работы в днях»
4. Предусмотреть механизм (кнопку) завершения последовательности ввода работ каждой фазы каждого из этапов.
5. Предусмотреть механизм (кнопку) завершения последовательности ввода работ перехода каждого из этапов, кроме последнего.
6. В результате выполнения заполнения всех формуляров в файловой системе должен быть сформирован текстовый файл, вида:

Sr;A1X1;DA1;XD1;A2X1; DA2;XD2;A3X1; DA3;XD3... AMX1; DAM;XDM<END>

Или

Sr;A1Y1; DA1;YD1;A2Y1; DA2;YD2 ... AMY1; DAM;YDM<END>

Где

Sr – имя фазы или имя перехода этапа проекта, $r=\{1,R\}$

Наименование этапа

Имя Фазы	Наименование работы:	
Номер дня начала	<div>1</div> <div>2</div> <div>3</div>	<div>Длительность</div> <div></div>
		OK

Имя Перехода	Наименование работы:	
Номер дня начала	<div></div>	<div>Длительность</div> <div></div>
		O'K

Работа по этапу завершена		O'K
---------------------------	--	-----

Имя Пере хода	Наименование работы:		
Номер дня начала		Длитель ность	
			О'К

Работа по этапу завершена	O'K
---------------------------	-----

Необходимые пояснения для выполнения задания находятся в файле **Explntn-4-LR404.pdf** .

Контрольные вопросы

1. Базовые понятия подсистемы **Project Management** системы **PLM**.
2. Жизненный цикл изделия, этапы. Фаза, Шлюз или переход.
3. Состав работ этапа – Workflow.
4. Результативность проекта. Диаграммы Ганта.
5. Обеспечение выполнения действий жизненного цикла изделия.
6. Графическое представление процессов.
7. Основные классы widget'ов **Tcl/Tk**. Назначение, связь с иерархией widget'ов X11.
8. Воспроизведение widget'ов в рамках проектируемого GUI. Диспетчеры компоновки – геометрические менеджеры **Tcl/Tk**.
9. Контейнеры для widget'ов. Способы создания т.н. вложенных интерфейсов – группирование widget'ов.