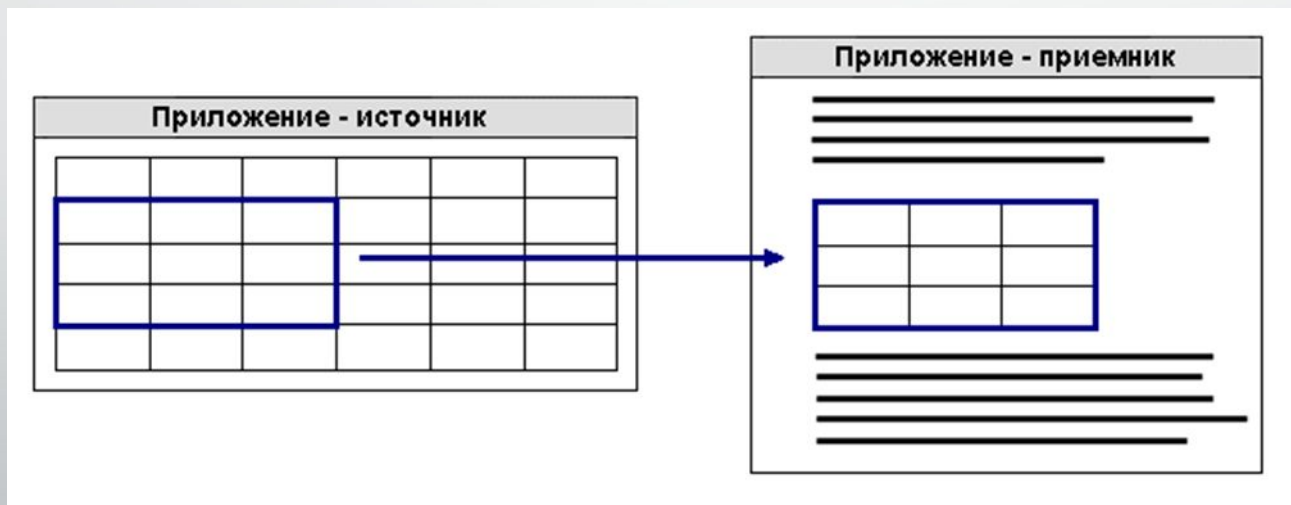


# Принцип обмена информацией между Windows. Технология OLE и COM

*Лекция*

# OLE-объекты

Объектом (OLE-объектом) называется произвольный элемент, созданный в каком-либо одном приложении (приложении-источнике, серверном приложении), который может быть помещен в файл другого приложения (приложения-приемника, приложения-клиента).



# Примеры использования объектов:

- В документ MS Word помещен рисунок, созданный графическим редактором с логотипом предприятия или рабочая книга MS Excel, которая используется в качестве отчета.
- Вы хотите поместить в базу данных MS Access иллюстрацию с изображением описываемого данной записью физического объекта.
- Отображение некоторой информации необходимо отложить до более подходящего момента. Например, в изображении блок-схемы, каждый элемент такой схемы может быть текстовым документом, содержащим десятки или сотни строк. Вставив соответствующие документы в виде значков, можно скрыть эти длинные описания до того момента, когда они понадобятся пользователю.

# Что такое OLE?

- Основа взаимодействия приложений в Windows 95 и выше - технология связывания и внедрения объектов (**Object Linking and Embedding, OLE**). OLE обеспечивает совместное использование данных приложениями, соответствующими стандарту OLE.
- Windows 3.1/3.11 имели механизм **динамического обмена данными DDE (Dynamic Data Exchange)** между различными приложениями. Его смысл заключается в том, что приложения связаны между собой общими файлами, так что изменение файла приложения-сервера тут же сказывается на документах, с которыми работают приложения-клиенты.

# Внедрение и установление связи

Данные можно **внедрять (Embedding)** и **связывать (Linking)**.

Связанный объект - это ссылка на исходный документ, динамически обновляющаяся при изменении источника. Внедренный объект - копия данных источника. При этом связь с собственно данными исходного документа отсутствует.

# Внедрить или связать?

- **Объем памяти, занимаемый объектом.** Если объем дисковой памяти для объекта является настолько большим, что хранить дополнительные копии объекта затруднительно, то следует остановиться на установлении связи.
- **Изменяется ли объект.** Если вы хотите, чтобы изменяющееся состояние объекта отражалось в документе-клиенте, то необходимо использовать установление связи.
- **Доступ к объекту.** Например, вы установили связь с объектом, получив к нему доступ по компьютерной сети. Если впоследствии сетевой диск окажется недоступным или хозяин исходного объекта изменит его местоположение или закроет к нему доступ, то лучше внедрить такой объект.
- **С помощью какого приложения создан объект.** Если исходное приложение поддерживает DDE, но не поддерживает OLE, то можно установить связь, но не внедрить такой объект. Если исходное приложение не поддерживает ни OLE ни DDE, то возможно удастся вставить такой объект как текстовый или графический файл.

## Подключение библиотек:

- ComObj;
- Excel2000;
- Word2000.

VAR E,W: VARIANT;

# Excel

- Создание объекта – *E:=CreateOleObject('Excel.Application');*
- Делаем видимым – *E.Visible:=true;*
- Создание стандартной книги – *E.Workbooks.add ();*
  - Создание 1 листа: *E.Workbooks.add (xlWBatWorkSheet);*
  - Название листа: *E.Workbooks[l].Worksheets[l].Name:='Лист 1'*
- Работа с ячейками:
  - *E.Cells[i, l]:='значение'; - ЗАПИСЬ*
  - *excelsheet.range['A1:e1'].merge - ОБЪЕДИНЕНИЕ*
  - *excelsheet.columns[i].columnwidth:=30; - ШИРИНА ЯЧЕЙКИ*



# Word

- Создание объекта – *W:=CreateOleObject('Word.Application');*
- Делаем видимым – *W.Visible:=true;*
- Создание документа– *W.Dokuments.add ();*
- Работа со строками:

*Word.ActiveDocument.Range.InsertAfter('текст после Hello World') - ЗАПИСЬ*

*WordDocument1.Range.InsertAfter(#13) – ПЕРЕХОД НА НОВУЮ СТРОКУ*