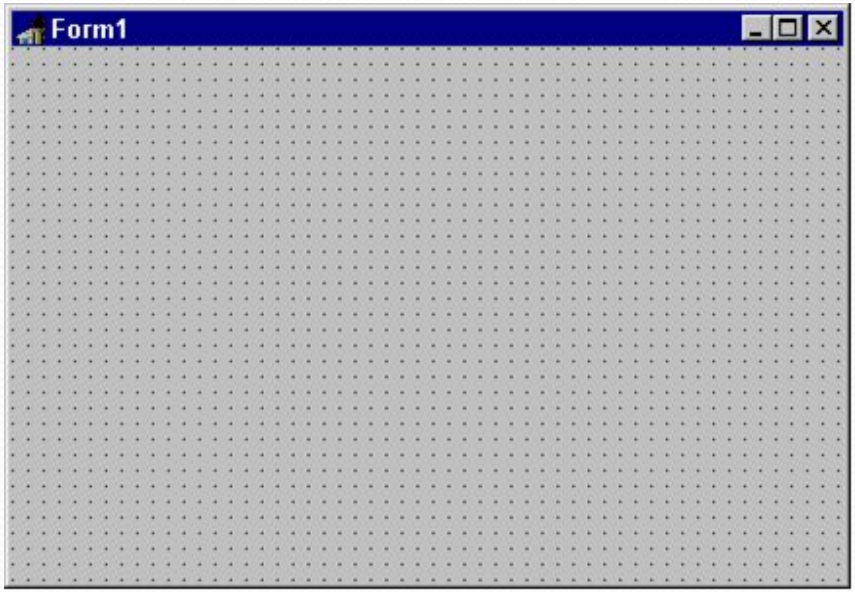
# Среда программирования Delphi

МОУ «Средняя общеобразовательная школа № 14
с углубленным изучением отдельных предметов»
авт. Кудимова Н.В.

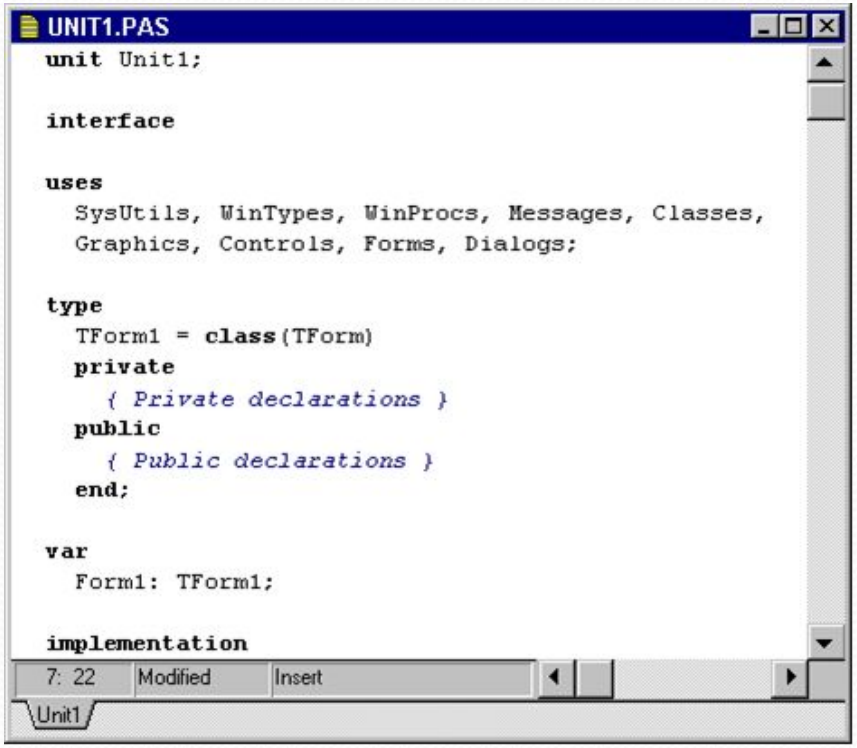
# Главные составные части среды программирования

- Дизайнер Форм (Form Designer)
- Окно Редактора Исходного Текста (Editor Window)
- Палитра Компонент (Component Palette)
- Инспектор Объектов (Object Inspector)
- Справочник (On-line help)

**Дизайнер Форм - то место, где Вы создаете визуальный интерфейс программы.**

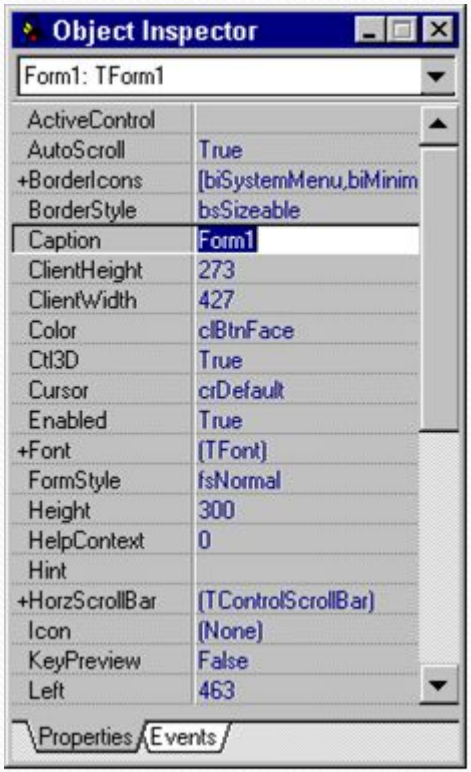



**В окне Редактора Вы создаете логику управления программой.**



**Палитра Компонент - место, где Вы выбираете объекты, которые будут помещены на вашу форму.**

**Инспектор Объектов позволяет определять свойства и поведение объектов, помещенных на форму.**



## Палитра компонент

- Для использования Палитры Компонент просто первый раз щелкните мышкой на один из объектов и потом второй раз - на Дизайнере Форм. Выбранный Вами объект появится на проектируемом окне и им можно манипулировать с помощью мыши.
- Палитра Компонент использует постраничную группировку объектов. Внизу Палитры находится набор закладок - Standard, Additional, Dialogs и т.д. Если Вы щелкнете мышью на одну из закладок, то Вы можете перейти на следующую страницу Палитры Компонент. Принцип разбиения на страницы широко используется в среде программирования Delphi и его легко можно использовать в своей программе. (На странице Additional есть компоненты для организации страниц с закладками сверху и снизу).

## Инспектор Объектов

- Инспектор Объектов состоит из двух страниц. Первая страница - это список свойств, вторая - список событий. Если нужно изменить что-нибудь, связанное с определенным компонентом, то Вы обычно делаете это в Инспекторе Объектов. К примеру, Вы можете изменить имя и размер компонента TLabel изменяя свойства Caption, Left, Top, Height, и Width.
- Страница событий связана с Редактором; если Вы дважды щелкнете мышкой на правую сторону какого-нибудь пункта, то соответствующий данному событию код автоматически запишется в Редактор, сам Редактор немедленно получит фокус, и Вы сразу же имеете возможность добавить код обработчика данного события.

# Стандартные компоненты

- **TMainMenu** позволяет Вам поместить главное меню в программу. При помещении TMainMenu на форму это выглядит, как просто иконка. Иконки данного типа называют "невидимыми компонентом", поскольку они невидимы во время выполнения программы. Создание меню включает три шага: (1) помещение TMainMenu на форму, (2) вызов Дизайнера Меню через свойство Items в Инспекторе Объектов, (3) определение пунктов меню в Дизайнере Меню.
- **TPopupMenu** позволяет создавать всплывающие меню. Этот тип меню появляется по щелчку правой кнопки мыши.
- **TLabel** служит для отображения текста на экране. Вы можете изменить шрифт и цвет метки, если дважды щелкнете на свойство Font в Инспекторе Объектов. Вы увидите, что это легко сделать и во время выполнения программы, написав всего одну строчку кода.
- **TEdit** - стандартный управляющий элемент Windows для ввода. Он может быть использован для отображения короткого фрагмента текста и позволяет пользователю вводить текст во время выполнения программы.

- **TMemo -** иная форма TEdit. Подразумевает работу с большими текстами. TMemo может переносить слова, сохранять в Clipboard фрагменты текста и восстанавливать их, и другие основные функции редактора. TMemo имеет ограничения на объем текста в 32Кб, это составляет 10-20 страниц. (Есть VBX и “родные” компоненты Delphi, где этот предел снят).
- **TButton** позволяет выполнить какие-либо действия при нажатии кнопки во время выполнения программы. В Delphi все делается очень просто. Поместив TButton на форму, Вы по двойному щелчку можете создать заготовку обработчика события нажатия кнопки.
- **TCheckBox** отображает строку текста с маленьким окошком рядом. В окошке можно поставить отметку, которая означает, что что-то выбрано. Например, если посмотреть окно диалога настроек компилятора (пункт меню Options | Project, страница Compiler), то можно увидеть, что оно состоит преимущественно из CheckBox’ов.

- **TRadioButton** позволяет выбрать только одну опцию из нескольких. Если Вы опять откроете диалог Options | Project и выберете страницу Linker Options, то Вы можете видеть, что секции Map file и Link buffer file состоят из наборов RadioButton.
- **TListBox** нужен для показа прокручиваемого списка. Классический пример ListBox'а в среде Windows - выбор файла из списка в пункте меню File | Open многих приложений. Названия файлов или директорий и находятся в ListBox'е.
- **TComboBox** во многом напоминает ListBox, за исключением того, что позволяет водить информацию в маленьком поле ввода сверху ListBox. Есть несколько типов ComboBox, но наиболее популярен выпадающий вниз (drop-down combo box), который можно видеть внизу окна диалога выбора файла.
- **TScrollbar** - полоса прокрутки, появляется автоматически в объектах редактирования, ListBox'ах при необходимости прокрутки текста для просмотра.

- **TGroupBox** используется для визуальных целей и для указания Windows, каков порядок перемещения по компонентам на форме (при нажатии клавиши TAB).
- **TPanel** - управляющий элемент, похожий на TGroupBox, используется в декоративных целях. Чтобы использовать TPanel, просто поместите его на форму и затем положите другие компоненты на него. Теперь при перемещении TPanel будут передвигаться и эти компоненты. TPanel используется также для создания линейки инструментов и окна статуса.
- **TScrollBox** представляет место на форме, которое можно скроллировать в вертикальном и горизонтальном направлениях. Пока Вы в явном виде не отключите эту возможность, форма сама по себе действует так же. Однако, могут быть случаи, когда понадобится прокручивать только часть формы. В таких случаях используется TScrollBox.

# Сохранение программы

- Первый шаг - создать поддиректорию для программы. Лучше всего создать директорию, где будут храниться все Ваши программы и в ней - создать поддиректорию для данной конкретной программы. Например, Вы можете создать директорию MYCODE и внутри нее - вторую директорию TIPS1, которая содержала бы программу, над которой Вы только что работали.
- После создания поддиректории для хранения Вашей программы нужно выбрать пункт меню File | Save Project. Сохранить нужно будет два файла. Первый - модуль (unit), над которым Вы работали, второй - главный файл проекта, который "владеет" Вашей программой. Сохраните модуль под именем MAIN.PAS и проект под именем TIPS1.DPR. (Любой файл с расширением PAS и словом “unit” в начале является *модулем*.)