

**MainForm** приложения Delphi - форма, которая создается первой в главном теле приложения.

Если Вам нужно организовать допуск к Вашему приложению, Вы могли бы отобразить форму ввода пароля для входа в приложение.

Короче говоря, идея состоит в том, чтобы создать, отобразить и уничтожить диалоговое окно "Вход в Систему" перед созданием Главной формы.

### MainForm Delphi

Когда создается новый проект Delphi, *Form1* автоматически становится **MainForm** (глобальный объект Application). Чтобы назначить другую форму для свойства **MainForm**, используйте вкладку Формы диалогового окна *Project | Options* во время проектирования.

Когда Главная форма закрывается, приложение заканчивает свою работу.

### Диалоговое Окно Ввода Пароля

Давайте сначала создадим Главную форму приложения. Создайте новый проект Delphi, который содержит одну форму. Эта форма в соответствии с проектом - Главная форма.

Если Вы измените имя формы на *TMainForm* и сохраните модуль как *main.pas*, исходный текст будет таким (проект был сохранен как *PasswordApp*):

```
program PasswordApp; uses Forms, main in 'main.pas' {MainForm}; {$R *.res}
begin Application.Initialize; Application.CreateForm(TMainForm, MainForm) ;
Application.Run; end.
```

Теперь добавим вторую форму в проект.

В соответствии с проектом, когда вторая форма будет добавлена, она будет присутствовать в списке *Auto-Create Forms* в диалоговом окне *Project Options*.

Назовите вторую форму как *TLoginForm* и удалите ее из списка автоматически создаваемых форм. Сохраните этот модуль под именем *login.pas*

Добавьте компоненты *Label*, *Edit* и *Button* на форму.

Добавьте метод класса для создания, отображения и закрытия диалогового окна ввода пароля. Метод **Execute** возвращает *True*, если Пользователь ввел правильный пароль в окне ввода пароля.

Вот полный исходный код:

```
unit login; interface uses Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls; type TLoginForm = class(TForm)
  LogInButton: TButton; pwdLabel: TLabel; passwordEdit: TEdit; procedure
  LogInButtonClick(Sender: TObject) ; procedure FormCreate(Sender: TObject) ; public
  class function Execute : boolean; end; implementation {$R *.dfm} class function
  TLoginForm.Execute: boolean; begin with TLoginForm.Create(nil) do try Result :=
  ShowModal = mrOk; finally Free; end; end; procedure
  TLoginForm.LogInButtonClick(Sender: TObject) ; begin if passwordEdit.Text = 'delphi' then
  ModalResult := mrOK else ModalResult := mrAbort; end; end.
```

Метод **Execute** динамически создает экземпляр *TLoginForm* и модально отображает его, используя метод **ShowModal**

**ShowModal** не возвращается, пока форма не будет закрыта. Когда форма закрывается, она возвращает свойство **ModalResult**.

Обработчик события **OnClick** для *LoginButton* назначает *mrOk* свойству *ModalResult*, если Пользователь ввел правильный пароль ( *delphi* в вышеупомянутом примере). Если Пользователь ввел неправильный пароль, *ModalResult* устанавливается в *mrAbort* (или что-нибудь вроде *mrNone* ).

Установив значение *ModalResult*, форма закрывается. **Execute** возвращает *True*, если *ModalResult* равен *mrOk*, т.е. Пользователь ввел правильный пароль.

### Не Создаем MainForm Перед Входом в Систему

Теперь, Вы должны удостовериться, что главная форма не была создана, если Пользователь ввел неправильный пароль.

Исходный код проекта выглядит так:

```
program PasswordApp; uses Forms, main in 'main.pas' {MainForm}, login in
'login.pas' {LoginForm}; {$R *.res} begin if TLoginForm.Execute then begin
Application.Initialize; Application.CreateForm(TMainForm, MainForm); Application.Run;
end else begin Application.MessageBox('Вы не авторизованы в этом приложении.
Правильный пароль "delphi".', 'Приложение защищенное паролем'); end; end.
```

Обратите внимание на использование, чтобы определить, была ли главная форма создана. Если **Execute** возвращает *False*, **MainForm** не создается, а приложение закрывается, даже не запустившись.

Вот и все!!!