

## Практическая работа № 2, Анатомия проекта

### Настройка IDE

Загрузите **Lazarus**. По умолчанию включена опция загрузки последнего проекта, поэтому у вас должен был загрузиться проект из первой практической работы. Это очень удобно, если вы работаете над каким-нибудь большим проектом, на разработку которого уходит не один день. Однако это будет неудобно при изучении **Lazarus** - ведь мы будем делать множество небольших программ, выполняющих разные задачи, и для этого нам нужно будет вначале закрыть старый проект, а затем создать новый. А это неудобно.

Выберите в главном меню команду **Сервис -> Параметры**, откроется окно **Параметры IDE**. Нам нужен раздел **Файлы**, расположенный в ветке **Окружение**:

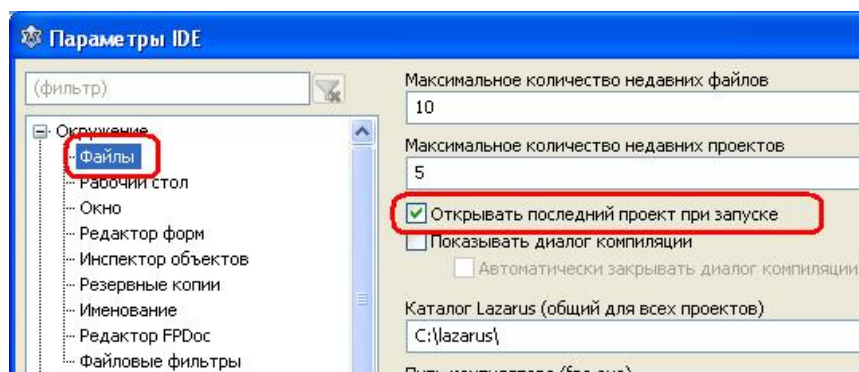


Рис. 2.1. Параметры - Файлы

В этом разделе имеется флажок **Открывать последний проект при запуске**. Если этот флажок включен, то каждый раз при запуске будет открываться последний проект. Если выключен - будет создаваться новый проект, что нам и нужно. Выключим его, нажмем кнопку **OK**, закроем **Lazarus** и снова запустим. Ну вот, совсем другое дело, создался новый проект. Ничего в проекте не меняя, просто сохраним его в папку **02**, предварительно ее создав.

Нажмем кнопку **Запустить** на **Панели инструментов** (зеленая стрелочка вправо), чтобы проект скомпилировался, и полученная программа сразу же загрузилась в память. Появится простое окошко нашей программы, ведь в ней ничего, кроме формы, нет. Закройте полученную программу, оставив **Lazarus** с проектом открытым.

Затем откройте папку с проектом с помощью **Проводника Windows** или привычного для вас файлового менеджера (TotalCommander, Far, и т.п.). В **Проводнике** установите вид отображения **Таблица**. Вы увидите примерно такую картинку:

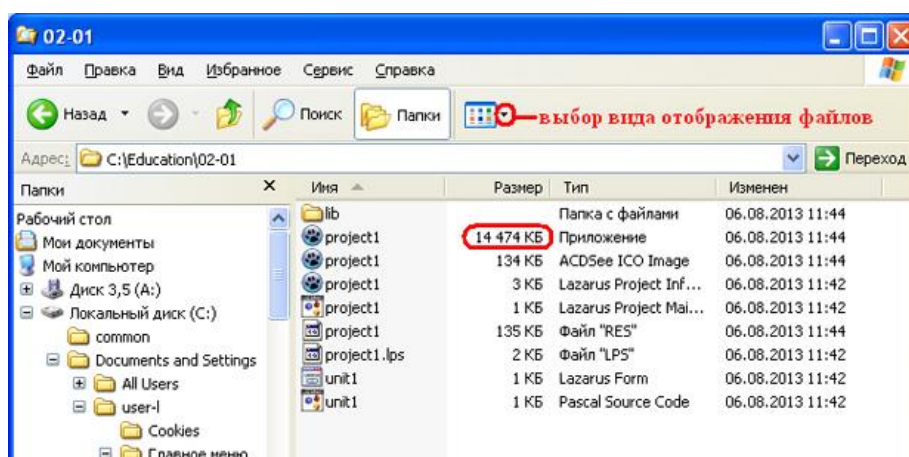


Рис. 2.2. Папка проекта с пустой формой

Наша программа получила название **project1.exe**, в колонке **Тип** значится **Приложение**. А вот в колонке **Размер** указано 14 474 килобайта (у вас может чуть отличаться, это не страшно). Ого! Программа с пустым окошком, которая совсем ничего не делает, "весит" больше 14 мегабайт!!! Как такое может быть?!

Дело в том, что при компиляции **Lazarus** по умолчанию добавляет в программу всевозможную отладочную информацию, в результате чего и получается такой большой размер. Это полезно, когда вы отлаживаете программу, но лишнее в нашем случае. К сожалению, отключить эту функцию можно только для текущего проекта. Делается это так: выберите команду **Главного меню Проект -> Параметры проекта**:

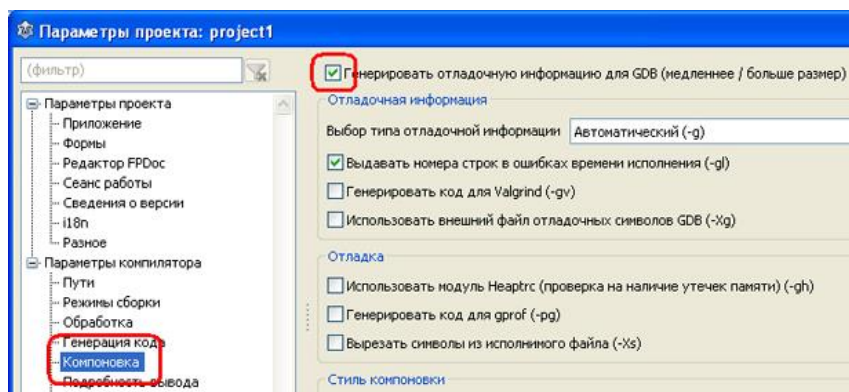


Рис. 2.3. Настройки проекта

В разделе **Компоновка** снимите флажок **Генерировать отладочную информацию для GDB**, нажмите **ОК** и заново запустите программу. После того, как программа запустится, закройте ее и снова посмотрите на полученный размер. Ну вот, теперь порядок, размер чуть больше полутора мегабайт. Тоже немало, если разобраться, но вполне терпимо, учитывая подключенные к проекту стандартные модули с поддержкой кроссплатформенности. В дальнейшем вы можете либо не обращать внимания на размер учебных программ - снимать этот флажок (после отладки, если она нужна) только для своих готовых реальных проектов, либо снимать его в каждом новом проекте. Это на ваше усмотрение. А если вам хочется получить совсем уж малый размер программы, то можно дополнительно воспользоваться каким-нибудь программным упаковщиком. **ASPack**, например, позволяет сжимать \*.exe и \*.dll файлы до 80%!

Также в настройках можно изменить размеры ячеек в сетке из точек, которая отображается в **Редакторе форм**. Эта сетка нужна для того, чтобы программист мог выравнивать компоненты на форме относительно друг друга. Выберите команду **Главного меню Сервис -> Параметры** и в разделе **Окружение** выберите **Редактор форм**. В правой верхней части расположен раздел **Сетка**, где можно изменять ее размеры (по умолчанию, установлено по 8 пикселей для оси X и для оси Y), разрешать или запрещать показ сетки, и т.д. Некоторые программисты предпочитают делать ячейки сетки меньше - 6 или даже 4 пикселя по обеим осям, мотивируя это тем, что чем чаще сетка, тем точнее можно подровнять компоненты. Может быть, они и правы, однако имейте в виду, что разработчики **Lazarus** выставляли по умолчанию наиболее удобные параметры. В общем, и тут установите настройки на свой вкус, или оставьте их по умолчанию.

## Из чего состоит проект

Выберите команду **Главного меню Сервис -> Параметры**, и в ветке **Окружение** перейдите на раздел **Файловые фильтры**. Вы увидите 6 основных типов файлов, которые могут встречаться в проекте:

- Модуль Lazarus (\*.pas;\*.pp)
- Проект Lazarus (\*.lpi)
- Форма Lazarus или Delphi (\*.lfm;\*.dfm)
- Пакет Lazarus (\*.lpk)
- Исходный код проекта Lazarus (\*.lpr)
- Иной файл Lazarus (\*.inc;\*.lrs;\*.lpl)

Если мы перейдем в папку с нашим проектом, то увидим, что он состоит из восьми файлов:

- **project1.exe** (Исполняемый файл программы).
- **project1.ico** (Файл с "иконкой" проекта - изображением в виде лапы гепарда, которое появляется в верхнем левом углу окна программы).
- **project1.lpi** (Информационный файл проекта). Если вы желаете открыть данный проект в **Lazarus**, то запускать нужно именно этот, информационный файл.
- **project1.lpr** (Исходный файл проекта). Запуск этого файла также приведет к запуску **Lazarus** с загрузкой данного проекта.
- **project1.lps** (Конфигурация проекта в виде xml-кода)
- **project1.res** (Файл ресурсов, используемых в проекте)

- **unit1.lfm** (Файл формы модуля. В нем в текстовом виде отображены настройки всех компонентов, используемых в модуле. Редактировать этот файл в ручную настоятельно не рекомендуется, для редактирования этих данных нужно использовать Редактор форм).
- **unit1.pas** (Исходный код модуля на языке Object Pascal).

Файлы с именем project1 - это файлы всего проекта в целом, файлы с именем unit1 - это файлы модуля.

**Модуль** - это отдельная единица исходного кода, выполненная в виде файла с расширением \*.pas. Совокупность таких единиц составляет программу.

Когда мы создаем окно, то для него создается два файла: модуль - файл \*.pas с исходным кодом, и файл \*.lfm, в котором содержатся настройки используемых на форме компонентов. Текст модуля мы можем видеть в **Редакторе кода**. Однако модуль не всегда имеет окно, это может быть и просто текстовый файл с исходным кодом. В нашем проекте всего один модуль, но вообще их может быть сколько угодно. И каждый модуль будет представлен этой парой файлов.

Кроме того, в папке проекта находится папка **lib**, в которой располагаются подключаемые к проекту данные и информация о компиляции. Если же вы изменяли проект, и сохраняли эти изменения, то появится также папка **backup**, в которой будут храниться резервные копии старых вариантов проекта.

Нередко программист добавляет в проект и свои типы файлов. Например, в проекте можно использовать базу данных, какой-нибудь текстовый файл или **ini-файл** для сохранения пользовательских настроек. Разумно располагать эти файлы также в папке с проектом.

Теперь пару советов по поводу наименования проекта и модулей. Проект следует называть так, как мы хотим, чтобы называлась наша программа. Например, проекту из первой практической работы было бы уместней дать имя "Hello" вместо нейтрального "project1".

Модули же нужно называть, исходя из их значения. Всегда в проекте есть **главный модуль**. В наших проектах пока что было по одному окну. Модуль, созданный для этого окна, и будет главным. В учебной литературе есть множество рекомендаций, как обозначать модули, остановимся на одной из них. Давайте договоримся в будущем главный модуль называть Main (англ. main - главный), а другим модулям давать смысловые названия, например, Options, Editor и т.п. Форму этого модуля (точнее, свойство **Name** формы) будем называть также, но с приставкой f-, обозначающей форму. То есть, **fMain, fOptions, fEditor** и так далее. Закрепим этот материал на практике.

Откройте **Lazarus**, если он у вас закрыт, или закройте старый проект и начните новый. В данный момент в **Редакторе форм** у нас пустая форма, в заголовке которой написано Form1 - это имя формы по умолчанию. Мы договорились называть главный модуль Main, а его форме добавлять приставку f-. В **Инспекторе объектов** найдите свойство **Name**, и вместо Form1 впишите туда **fMain**. Как только вы нажмете **<Enter>**, заголовок формы изменится на новый. Теперь сохраним проект и модуль главной формы. Нажмите кнопку **Сохранить все** на **Панели инструментов**, или выберите команду **Главного меню Файл -> Сохранить все**.

В запросе вместо имени проекта project1 укажите новое имя Hello. Как только вы нажмете кнопку **Сохранить**, выйдет запрос на сохранение главного модуля. Форму мы назвали fMain, значит, модулю дадим название просто Main. В **Lazarus** строчные и заглавные буквы не различаются, однако для удобочитаемости кода лучше приучиться использовать заглавные буквы, чтобы выделять названия. Например, FileEdit, SaveAll и т.п.

В свойстве **Caption** формы впишем слово "Приветствие" (разумеется, без кавычек), это будет более понятным заголовком для окна. Не забывайте после ввода новых значений свойств в **Инспекторе объектов** нажимать **<Enter>**, чтобы изменения вступили в силу. Теперь установим на форму компонент **TLabel** (метку), который позволит выводить на форме текст. Компонент находится на вкладке **Standard**:



**Рис. 2.4.** Метка TLabel

Подсказка: если подвести указатель мыши к компоненту и какое-то время не нажимать кнопку, выйдет всплывающая подсказка с именем компонента.

Щелкните мышкой по метке, затем по форме, в верхней части окна. Поскольку метка у нас одна, то можно оставить ей имя (свойство **Name**) по умолчанию - **Label1**. А вот в свойстве **Caption** метки напишите:

### Как вас зовут?

Ниже метки поместите компонент **TEdit** - редактируемое текстовое поле, в котором пользователь сможет написать что-то:

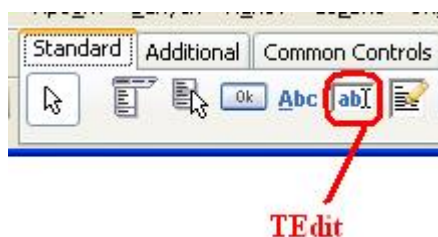


Рис. 2.5. Текстовое поле TEdit

У этого компонента свойство **Name** также оставим по умолчанию - **Edit1**. Как вы можете заметить, у этого компонента свойства **Caption** нет, зато появилось свойство **Text** - именно тут и содержится текст, отображенный в поле. По умолчанию, он совпадает с именем компонента. Просто очистим это свойство, удалив из него старый текст (не забывайте про **<Enter>**).

Еще ниже установим кнопку **TButton**. Оставим ее имя по умолчанию, а в свойстве **Caption** напомним

### Выполнить

Изменим положение и размеры компонентов и самой формы так, чтобы форма приняла примерно такой вид:

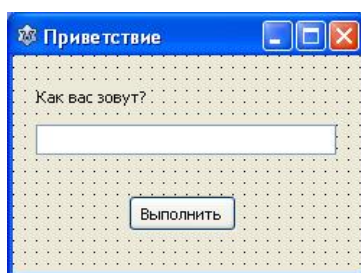


Рис. 2.6. Окончательный вид главной формы

Теперь запрограммируем нажатие кнопки. Щелкните по ней дважды, чтобы сгенерировалось событие, и автоматически открылся **Редактор кода**. В месте, где мигает курсор, впишем следующий код:

```
ShowMessage('Привет, ' + Edit1.Text + '!');
```

Редактор кода должен выглядеть так:

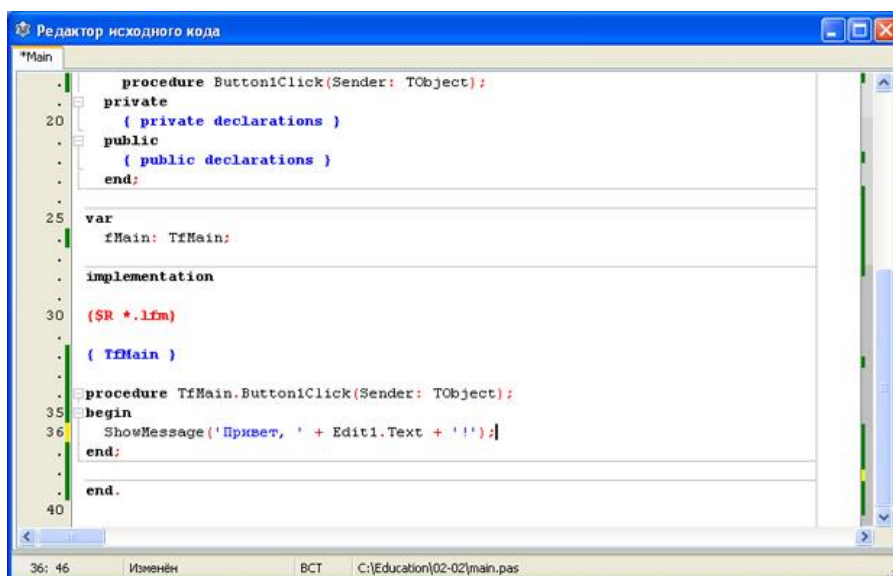
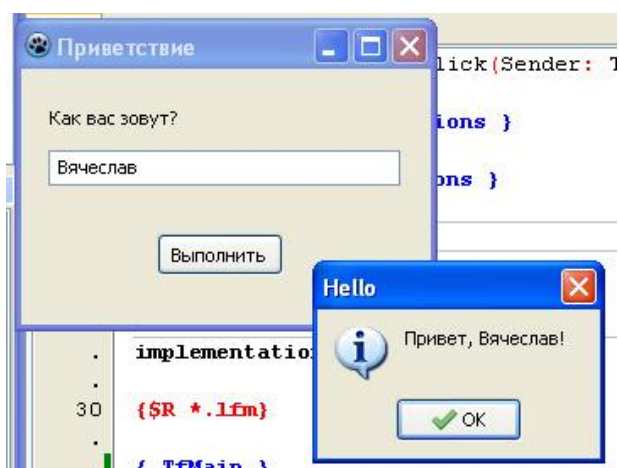


Рис. 2.7. Введенный код

Сохраните проект и запустите его. Когда программа загрузится, впишите в окне **Edit1** свое имя и нажмите кнопку **Выполнить**. Вы должны получить примерно такой результат:



**Рис. 2.8.** Программа Hello.exe в действии

Вы можете сколько угодно раз менять имя в текстовом поле и снова нажимать кнопку **Выполнить**, получая приветствие уже с другим текстом. У нас получилась настоящая **интерактивная** программа, то есть, программа, взаимодействующая с пользователем.

Добавим на форму еще одну кнопку **TButton**, в свойстве **Caption** напишем **Выход**.

Теперь запрограммируем нажатие кнопки. Щелкните по ней дважды, чтобы сгенерировалось событие, и автоматически открылся **Редактор кода**. В месте, где мигает курсор, впишем следующий код:

**Close;**

Сохраните проект. Теперь при нажатии на эту кнопку будет проект закрыт.

## Краткое описание плана разработки программы

В этом разделе показано, как можно кратко описать план разработки программы.

Для краткости в дальнейшем будем использовать этот способ записи.

1. Откройте новый проект.
2. Разместите в форме экземпляры компонентов компоненты: метку **Label**, текстовое поле **Edit** и две кнопки **Button**. Выполните следующие действия:

Выделенный объект	Вкладка окна Инспектора объектов	Имя свойства/Имя события	Значение/Действие
Form1	Свойства	Caption	Мой проект
		Name	fMain
Edit1	Свойства	Caption	Удалить текст
Button1	Свойства	Caption	Выполнить
	События	OnClick	ShowMessage('Привет, ' + Edit1.Text + '!');
Button2	Свойства	Caption	Выход
	События	OnClick	Close;

4. Сохраните проект и код программы под именами **Hello.lpr** и **Main.pas**.

5. Запустите проект, выполните его, а затем закройте окно проекта кнопкой «Выход».