

Лабораторная работа 1

Тема: Знакомство со структурой среды программирования Lazarus. Создание простейших Windows-приложений

Цель работы: познакомиться со структурой среды программирования Lazarus и научиться создавать простейшие Windows-приложения.

Теоретическое введение и упражнения

1 Концепции объектно-ориентированного программирования

Объектно-ориентированное программирование (ООП) – это методика разработки программ, в основе которой лежит понятие объекта. **Объект** — это сущность, которой можно посылать сообщения, и которая может на них реагировать, используя свои данные. **Объект** — это экземпляр класса. Данные объекта скрыты от остальной программы. Скрытие данных называется **инкапсуляцией**.

Наличие инкапсуляции достаточно для объектности языка программирования, но ещё не означает его объектной ориентированности — для этого требуется наличие **наследования** (свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью). Класс, от которого производится наследование, называется базовым, родительским или суперклассом. Новый класс — потомком, наследником или производным классом.

Но даже наличие инкапсуляции и наследования не делает язык программирования в полной мере объектным с точки зрения ООП. Основные преимущества ООП проявляются только в том случае, когда в языке программирования реализован **полиморфизм**, то есть возможность объектов с одинаковой спецификацией иметь различную реализацию.

Таким образом, в основе ООП лежат три **концепции**: инкапсуляция, наследование и полиморфизм.

В основе всех современных сред программирования лежат концепции ООП. И все среды разработки являются средами визуального проектирования, включая и среду разработки Lazarus. Основное преимущество сред визуального проектирования – освобождение разработчика программ от необходимости рутинной работы по созданию стандартизированных элементов интерфейса разрабатываемого проекта (кнопок, меню, строк ввода и тому подобное). Эти элементы уже созданы разработчиком среды и их состав, свойства, принципы использования являются практически промышленным стандартом и одинаковы в любой среде разработки современных программ.

Рассмотрим создание простейших Windows-приложений.

2 Структура среды программирования Lazarus

Среда программирования Lazarus состоит из нескольких отдельно расположенных окон, рисунок 1.

Окна могут перемещаться по экрану, частично или полностью перекрывая друг друга.

После запуска Lazarus перед вами появятся основные окна:

1. Главное окно (Project1).
2. Окно формы (Form1).
3. Окно Инспектора Объектов (Object Inspector).
4. Окно кода программы или окно редактора кода (Unit1.pas).

Первоначально окно кода перекрыто окном формы, для переключения между которыми осуществляется клавишей F12.

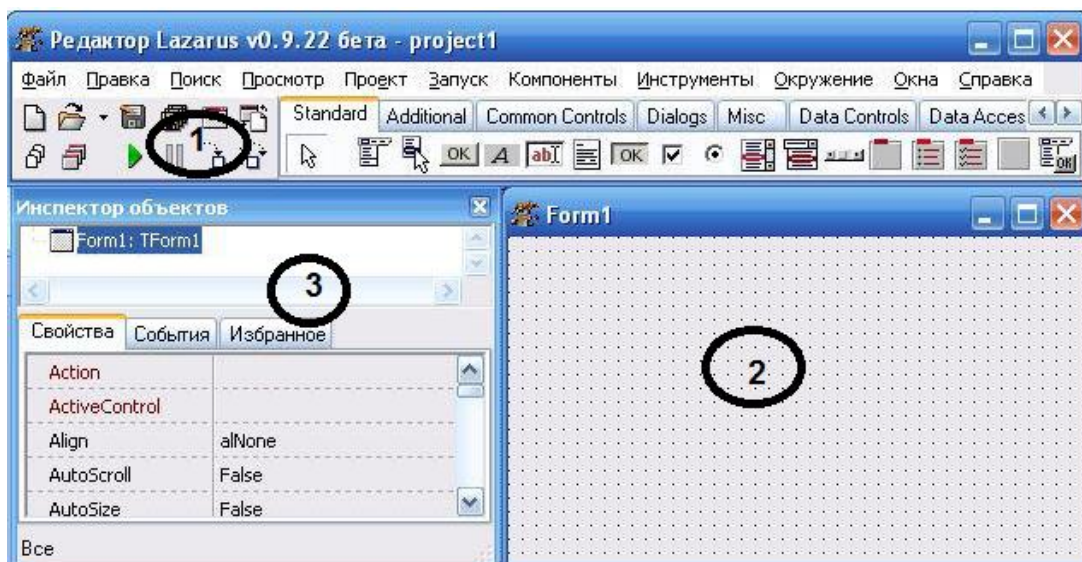


Рисунок 1 - Среда программирования Lazarus

3 Lazarus – объектно-ориентированная среда

Объектно-ориентированное программирование – это методика разработки программ, в основе которой лежит понятие объекта.

Объект – это совокупность свойств (параметров) определенных сущностей и методов их обработки (программных средств).

Описание объекта рассмотрим на примере объекта автомобиль. Каждый объект имеет:

- **свойства** – характеристики объекта (автомобиль характеризуется маркой, цветом, салоном, местонахождением руля, коробкой передач и т.д.);
- **методы** – действия объекта, что объект может делать (для автомобиля действиями являются движение вперед, назад, торможение, остановка, поворот и др.);
- **события** – изменения в окружающей объект обстановке (события, которые влияют на объект автомобиль: сигнал светофора, знаки милиционера и др.);
- **реакция на события** – описания действий, которые необходимо совершить при данном событии.

Все объекты реализованы в Lazarus в виде палитры визуальных компонентов, рисунок 2.

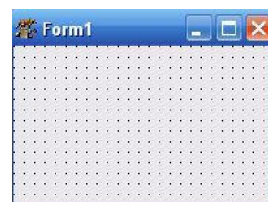


Рисунок 2 - Среда программирования Lazarus

Компоненты сгруппированы в отдельные страницы, каждая из которых снабжена закладкой. Если щелкнуть мышью на одну из закладок, то можно перейти на соответствующую ей страницу. При подведении курсора мыши к компоненте (или нажатие правой кнопки мыши на компоненте) появляется подсказка – название компоненты.

4 Форма. Инспектор объектов. Окно кода программы

Форма – окно будущей программы. Окно формы представляет собой проект окна Windows: имеет заголовок, кнопку вызова системного меню, кнопку максимизации, минимизации и закрытия окна, рамку окна. На форме размещаются компоненты, тем самым, формируя заготовку



(макет) окна разрабатываемого приложения.

Окно **Инспектор Объектов** содержит страницы: **Свойства** (Properties) и **События** (Events), каждая из которых разделена на две части.

Страница **Свойства** служит для указания свойств объекта. На этой странице указаны все свойства активного (выделенного) объекта. В левой части страницы находится название свойства, а в правой – его значение. «+» слева от названия свойства указывает на то, что свойство состоит из нескольких значений. Значениями свойств могут быть слова, числа, значения из раскрывающегося списка. Нажав на «▼» справа от поля значения свойства, можно получить раскрывающийся список значений. При нажатии на «...» вызывается специальное диалоговое окно.

Если значением свойства является число или текст, то после его набора необходимо нажать Enter, иначе оно может быть не зафиксировано. При нажатии на Esc ввод отменяется.

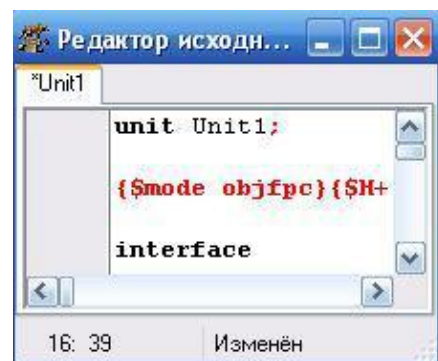
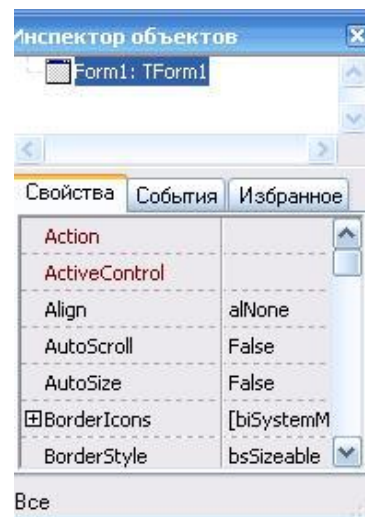
Страница **События** используется для задания реакции на событие. Страница также состоит из двух частей. В первой – название события, а во второй – название процедур, обрабатывающих данное событие. Если в правой части ничего не написано, то программа на данное событие не реагирует. Название процедуры состоит из двух частей: **названия объекта + название события**. Для создания новой реакции на событие необходимо дважды щелкнуть в правой половине напротив нужного события. В этом месте появится название новой процедуры, а на экране автоматически открывается окно редактора кода.

Окно кода программы предназначено для создания и редактирования текста программы. Этот текст составляется по определенным правилам.

Для **написания** программы необходимо:

1. Разместить необходимые компоненты на форме.
2. Задать свойства выбранных компонент.
3. Определить событие.
4. Задать реакцию на событие.

Для размещения компонент на форме нужно щелкнуть мышью на компоненте, после щелкнуть на форме.



5 Краткая характеристика некоторых компонентов

У всех компонент есть свойство **Name** – имя компонента. Имена создаются средой Lazarus по такому принципу: сначала идет название компонента (Form), а за ним порядковый номер размещенного на форме компонента (1). То есть если добавить еще одну форму, то она получит имя Form2, следующее – Form3 и т.д. Имя, заданное по умолчанию можно изменить, но при этом желательно использовать только английские буквы и цифры.

Запомните! Свойство **Name** задается первоначально, это имя переменной с которой вы будете работать при написании программы.

5.1 Компонент Form (экранная форма)

Форма представляет не только внешний вид окна приложения, но и сама является полноценным компонентом с собственными свойствами и событиями, хотя на палитре компонентов ее нет.

Основные свойства и события компонента Form приведены в таблицах 1.1 и 1.2.

Таблица 1.1 – Свойства компонента Form

Свойство	Значение свойства
Caption	Задаёт заголовок окна формы
Color	Задаёт цвет формы
Font	Задаёт атрибуты шрифта формы

Значения свойств можно задать либо в окне свойств объекта, либо в программе.

Пример использования в программе:

Form1.Color:=clred; {задание красного цвета формы}

Таблица 1.2 – Основные события компоненты Form

События	Значение события
OnCreate	Создание формы
OnClose	Закрытие формы

5.2 Компонент Label (надпись или метка)

Компонент Label можно использовать для вывода ответа или пояснения вводимых данных. Основные свойства компонента Label приведены в таблице 1.3.

Таблица 1.3 – Основные свойства компоненты Label

Свойство	Значение свойства
Caption	Задаёт заголовок надписи, выводимой на экран
Font	Задаёт шрифт, используемый для отображения текста
Visible	Задаёт видимость надписи на экране. Имеет два значения. Если значение True, то надпись видна, False – нет.
Alignment	Расположение текста в метке: taRightJustify – текст прижат к правому краю; taCenter – текст размещён по центру; taLeftJustify – текст прижат к левому краю.

Пример использования в программе:

Label1.Caption:='Результат';

5.3 Компонент Edit (однострочное окно ввода/вывода)

Компонент Edit используется для ввода/вывода чисел и текста в программу. Основные свойства и методы компонента Edit приведены в таблицах 1.4 и 1.5.

Таблица 1.4 – Основные свойства компонента Edit

Свойство	Значение свойства
AutoSize	Задаёт необходимость изменения размера компонента при изменении размера шрифта (если True)
Text	Задаёт содержимое строки редактирования (изначально задается пустым)

Таблица 1.5 – Основные методы компонента Edit

Метод	Значение метода
Clear	Удаляет весь текст
SetFocus	Устанавливает фокус ввода

5.4 Компонент Panel (контейнер общего назначения)

Компонент Panel используется для размещения на ней сгруппированных компонент. Основные свойства компонента Panel приведены в таблице 1.6.

Таблица 1.6 – Основные свойства компонента Panel

Свойство	Значение свойства	
AutoSize	Задаёт необходимость изменения размера компонента при изменении размера шрифта (если True)	
BevelOuter	Внешняя рамка	bvRaised (выпуклая) bvLowered (вдавленная) bvNone (отсутствует)
BevelInner	Внутренняя рамка	
BevelWidth	Ширина рамки	

5.5 Компонент Button (командная кнопка)

Компонент Button используется для задания реакции на событие. Основные свойства и события компонента Form приведены в таблицах 1.7 и 1.8.

Таблица 1.7 – Основные свойства компонента Button

Свойство	Значение свойства
Caption	Задаёт название кнопки
Height	Задаёт высоту кнопки
Width	Задаёт ширину кнопки
Left	Задаёт расстояние от левой границы кнопки до левой границы формы
Top	Задаёт расстояние от верхней границы кнопки до верхней границы формы

Таблица 1.8 – Основные события компонента Button

События	Значение события
OnClick	Происходит, когда пользователь щёлкает основной (левой) кнопкой мыши на объекте
OnDbClick	Происходит, когда пользователь дважды щёлкает основной (левой) кнопкой мыши на объекте
OnClose	Закрытие окна формы

Задание 1. Изменить заголовок окна формы с Form1 на «Первая программа», используя свойство Caption.

Задание 2. Изменить цвет формы на clAqua, используя свойство Color.

Задание 3. Разместить в центре формы компоненту Label. Задать: надпись метки – СВЕТОФОР, цвет метки – серый. Изменить свойство Font: шрифт – Times New Roman, начертание – жирный, размер – 14.

Задание 4. Расположить на форме компоненты Panel1, Panel2, Panel3, для которых поочередно задать свойство Caption пустым.

Задание 5. Расположить на форме три командные кнопки Button1, Button2, Button3. Задать над-писи на этих кнопках «Красный», «Желтый», «Зеленый». В результате должна получиться форма, согласно рисунка 3.



Рисунок 3 – Результат заданий 1-5

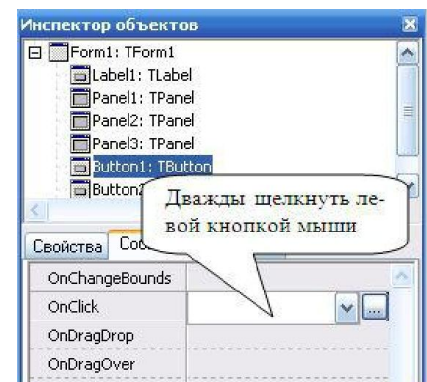
6 События. Реакция на события. Окно кода программы

События устанавливаются в окне **Инспектора Объектов** на странице **События**.

В данной программе при нажатии кнопки мыши (событие **OnClick**) на командные кнопки будет соответственно изменяться цвет панели.

Задание 6. Задать событие для первой командной кнопки Button1.

Для этого необходимо выделить данный компонент и перейти в **Инспекторе Объектов** на страницу События. Затем напротив события **OnClick** дважды щелкнуть левой кнопкой мыши. После выбора события автоматически открывается окно кода программы.



Окно кода программы предназначено для создания и редактирования текста программы. Этот текст составляется по определенным правилам. Совокупность правил записи текста называется языком программирования. В системе программирования Lazarus используется язык программирования Free Pascal.

После задания события в коде программы автоматически создается процедура обработки события, внутри которой описывается реакция на событие. Заголовок процедуры формируется следующим образом:

```
Procedure TForm1.Button1Click (Sender:TObject); {событие нажатие мыши на компоненте  
Button1, расположенной на форме Form1 }  
Begin  
  
End;
```


Между **Begin** и **End** указывается реакция на события – перечень совершаемых действий.

Задание 7. При нажатии на командную кнопку Button1 с надписью «Красный» цвет компоненты Panel1 будет меняться на красный, а цвет компонент Panel2 и Panel3 будет меняться на белый. Записать в процедуре следующую последовательность действий:

Procedure TForm1.Button1Click (Sender:TObject);

Begin

Panel1.Color:=clRed;

Panel2.Color:=clWhite;

Panel3.Color:=clWhite;

End;

Задание 8. Создать событие и реакцию на событие для командной кнопки Button2: цвет компоненты Panel2 будет меняться на желтый, а цвет компонент Panel1 и Panel3 будет меняться на белый. Если вы не знаете, как записать название цвета, посмотрите возможные цвета свойства Color в Инспекторе Объектов.

Задание 9. Создать событие и реакцию на событие для командной кнопки Button3: цвет компоненты Panel3 будет меняться на зеленый, а цвет компонент Panel1 и Panel2 будет меняться на белый.

7 Запуск и сохранение программы


Перед первым запуском программы ее необходимо сохранить.

Проект Lazarus представляет собой набор программных единиц – модулей, которые хранятся в отдельных файлах. Поэтому каждую программу сохраняют в отдельной папке.


Для **сохранения программы** надо выбрать **Файл → Сохранить как**. Появится диалоговое окно **Сохранить проект project1 (*.lpi)**, в котором сохраняется проект, работающим с программой, с расширением ***.lpi**.

Далее появляется диалоговое окно **Сохранить Unit1 (*.pas)**, в котором на первом этапе сохраняется программа с расширением ***.pas**.

Запомните! Чтобы открыть программу, необходимо открыть проект с расширением **.lpi**. В среде Lazarus программу можно сохранить и командой **Файл → Сохранить все**.

Если после первого сохранения программы в нее были внесены изменения, то для их сохранения можно нажать на панели инструментов кнопку  или выбрать команду **Сохранить все**.

Запустить программу на исполнение можно одним из следующих способов:

1. Выбрать команду **Запуск → Запуск (Run → Run)**.
2. Нажать клавишу **F9**.
3. Нажать кнопку  на панели инструментов.

После запуска программы она может зависнуть. Для остановки зависшей программы необходимо нажать комбинацию клавиш **Ctrl+F2** или выбрать команду **Запуск → Останов**.

Задание 10. На сетевом диске в своей папке для первой лабораторной работы сохраните программу.

Задание 11. Добавить появление на компоненте Panel1 при нажатии на командную кнопку Button1 информации «СТОЙТЕ» белым цветом, жирным шрифтом, размер шрифта – 12.

Для этого необходимо в имеющуюся процедуру добавить следующие действия:

Procedure TForm1.Button1Click (Sender:TObject);

Begin

Panel1.Color:=clRed;

Panel2.Color:=clWhite;

Panel3.Color:=clWhite;

Panel1.Caption:='СТОЙТЕ'; {задание на панели надписи}

Panel1.Font.Color:=clWhite; {задание цвета шрифта}

Panel1.Font.Size:=12; {задание размера шрифта}

Panel1.Font.Style:=[fsBold]; {задание начертания шрифта}

End;

Задание 12. Добавить появление на компоненте Panel2 при нажатии на командную кнопку Button2 информации «ВНИМАНИЕ» белым цветом, жирным шрифтом, размер шрифта – 12.

Задание 13. Добавить появление на компоненте Panel3 при нажатии на командную кнопку Button3 информации «ИДИТЕ» белым цветом, жирным шрифтом, размер шрифта – 12.

Задание 14. Сохранить изменения в программе и запустить ее на исполнение.

8 Окна сообщения

При работе с любой программой можно выводить окна сообщения. Для этого существует команда:

ShowMessage('текст сообщения');

Задание 15. На форме добавить командную кнопку Button4. Задать для нее надпись «Автор». При нажатии на кнопку должно выводиться сообщение об авторе программы. Для реализации данного задания задать для нее реакцию на событие:

Procedure TForm1.Button4Click(Sender:TObject);

begin

ShowMessage('Программа разработана Ивановым С.');

end;

Задания для самостоятельного выполнения

1. Составьте программу, которая при выборе соответствующей командной кнопки, изменяет содержание надписи «Язык программирования» на «Система программирования».

2. Составьте программу, которая при выборе соответствующей командной кнопки, выводит сообщение «Спасибо за работу» затем завершает работу программы.

3. Составьте программу, которая при выборе соответствующей командной кнопки, меняет цвет формы на голубой, фиолетовый, красный.

4. Составьте программу, которая при выборе соответствующей командной кнопки, выводится сообщение об условии задачи.

5. Составьте программу, которая при выборе соответствующей командной кнопки, меняет вид отображения рамки панели «выпуклая», «вдавленная», «отсутствует».

6. Составьте программу, которая выводит сообщение о назначении компонентов при щелчке мыши на данной компоненте (для кнопки и панели).

Контрольные вопросы

- 1 Какие концепции лежат в основе ООП?
- 2 Что такое наследование, инкапсуляция и полиморфизм в ООП?
- 3 Перечислите элементы окна Lazarus и их назначение.
- 4 Как задать значение свойств компонент?
- 5 Как создать процедуру реакции на нажатие кнопки, помещенной на форму?
- 6 Как вывести текст на форме?
- 7 Как запустить созданное приложение?
- 8 Как сохранить приложение?
- 9 С какими расширениями создаются файлы при сохранении приложения?