

ОБЪЕКТНО-СОБЫТИЙНОЕ ПРОГРАММИРОВАНИЕ: ОТ ТЕОРИИ К ПРАКТИКЕ

Шумчик Ф.С., Ручаевская Е.Г.

Белорусский государственный университет информатики и радиоэлектроники филиал Минский радиотехнический колледж, г. Минск, Беларусь, zam_ur@mrk-bsuir.by

Аннотация. Рассмотрена методика изучения основных понятий объектно-событийного программирования, представлен опыт практической реализации теоретического материала на примере создания проекта.

Ключевые слова. Визуальное программирование, оконное приложение, форма, объект, событие, обработчик события.

В рамках программы учебного предмета «Информатика» для специальностей учреждений среднего специального образования на основе общего базового образования на изучение темы «Введение в объектно-событийное программирование» отводится небольшое количество часов. Поэтому материал необходимо подать сжато, но в полном объеме.

На первом этапе знакомства с темой для понимания объектно-событийной модели работы программы необходимо обратить внимание учащихся прежде всего на понятия визуальное программирование и оконные приложения. Визуальное программирование – это проектирование интерфейса. Программы с оконным интерфейсом называют оконными приложениями [1, с. 4].

Далее идет знакомство с основными элементами управления (на примере среды программирования PascalABC.Net, поскольку данный язык является базовым в школьном изучении информатики). В среду PascalABC.Net встроен дизайнер форм, позволяющий создавать полноценные оконные приложения в стиле RAD (Rapid Application Development – быстрое создание приложений). Приступаем к выполнению следующей команды: *Файл* → *Новый проект* → *Приложение Windows Form*. Рассматриваем Палитру компонентов, раскрываем содержание панелей компонентов палитры и отдельно останавливаемся на панели компонентов «Стандартные элементы управления», где изучаем основные элементы управления, поочередно перетаскивая их на форму с помощью мыши: *кнопка (Button)*, *надпись (Label)*, *текстовое поле (TextBox)*, *флажок (CheckBox)*, *радиокнопка (RadioButton)*, *рисунок (PictureBox)*, *список (ListBox)* и др.

После изучения основных элементов интерфейса переходим к рассмотрению формы, на основе которой строится внешний вид оконного приложения. Здесь важно обратить внимание учащихся на то, что оконные приложения создаются как проект, структура которого содержит основной файл для описания структуры проекта, команд по созданию формы и запуску приложения, а также файлы для сохранения формы, один из которых содержит описание внешней стороны формы, другой фиксирует действия над элементом управления. Желательно учащимся продемонстрировать на примере предварительно подготовленного проекта.

Все элементы формы и сама форма – это объекты, взаимодействие которых основано на объектно-ори-

ентированном программировании. Здесь необходимо учащимся раскрыть понятие объектно-ориентированное программирование как технологию создания программ с помощью использования системы объектов. При этом важно акцентировать внимание на том, что объекты обладают свойствами, которые характеризуют их содержание, и методами, с помощью которых происходит обработка данных [2, с. 64].

На данном этапе занятия переходим к рассмотрению окна среды программирования PascalABC.Net при создании проекта Windows Form. Изучаем Инспектор объектов, открываем вкладку Свойства. В левом столбце вкладки представлены свойства объекта, которые можно использовать при проектировании приложения. Изучаем основные свойства:

- свойства, формируемые внешний вид объекта: *BackColor, Backgroundimage, Font, Forecolor, Text;*
- свойства, используемые при создании макета объекта: *Autoscroll, Autosize, Autosizemode, Location, Size;*
- свойства, определяющие состояние, поведение объекта: *Doublebuffered, Enabled, Imemode, Visible;*
- свойства по созданию стиля окна объекта: *Controlbox, Icon, Opacity, Showicon, Topmost* и т. д.

При этом важно обратить внимание учащихся на то, что каждый объект имеет как общие, так и особенные, отличные от других, свойства. Например, общими для многих объектов являются такие свойства, как *Text, Backcolor, Font, Size* и др. К отличительным свойствам объекта *ListBox*, например, относится свойство *Items* (редактор коллекции строк), объекта *TextBox* – свойство *MaxLenght* (задает максимальное число символов, которое можно ввести в поле редактирования), *ReadOnly* (задает, может ли изменяться текст в поле редактирования) и т. д.

В правом столбце вкладки Свойства перечислены значения свойств, которые можно выбирать из списка или вводить с помощью клавиатуры. Например, свойство *BackColor* имеет различные значения для подбора соответствующей цветовой гаммы объекта; значения свойства *Font* дают возможность подобрать нужный шрифт для текстового оформления объекта; значения свойства *Size* позволяют управлять размером объекта и т. д.

Далее переходим к понятию событийно-ориентированное программирование.

Прежде всего рассматриваем понятия событие и обработчик события. Взаимодействие между программой, основанной на использовании системы

объектов, и пользователем управляется событиями. Событие – это действия пользователя с помощью мыши, клавиатуры, функций операционной системы, направленные на объект для выполнения определенного программой алгоритма обработки данных [3, с. 49]. «Каждое событие связано с каким-либо объектом, которому передается управление в тот момент времени, когда происходит событие» [4, с. 9]. Процедуру, вызванную событием, принято называть обработчиком событий. Для каждого события, как правило, определен свой обработчик, который создается при двойном клике в пустой строке напротив выбранного события. Процедура генерируется автоматически, после этого программа переключается на страницу, на которой пишется код.

Таким образом, событийно-ориентированное программирование – это метод программирования, основанный на выборке и обработке события [2, с. 64]. Как известно, обработка таких событий является ключевой задачей в написании современных программ [5, с.83].

Изучаем основные события, которыми являются события мыши, события клавиатуры и системные события.

Событие мыши возникает при щелчке элемента управления, когда производится какое-либо действие с мышью:

– обработчики событий *Click, DblClick, MouseClick, MouseDown, MouseEnter, MouseUp, MouseMove, MouseHover, MouseLeave* и др.).

Событие клавиатуры происходит при нажатии клавиш на клавиатуре:

– обработчики событий *KeyPress, KeyDown, PreviewKeyDown, KeyUp*).

Системные события проявляются как реализация функций операционной системы:

– обработчики событий *Paint, Enter, Leave, Layout, Resize* и др.).

Снова, как закрепление знаний, обращаем внимание учащихся на важный факт: каждый объект имеет по умолчанию свой обработчик событий; обработчик событий создается с помощью двойного клика в пустой строке напротив определенного объекта; в результате данного действия среда переключается на страницу, на которой пишется код.

Откроем в окне среды программирования Pascal-ABC.Net при создании проекта Windows Form вкладку События, рассмотрим список событий для различных объектов. Список событий представим по категориям: внешний вид, действие, ключ, макет, мышь, перетащить и отпустить, поведение и т.д. Как видим, со многими объектами можно производить самые различные процедуры, вызываемые событиями. Но при этом для каждого объекта определено основное событие: например, для формы основным событием является событие Load, для текстового поля (TextBox) – событие TextChanged (происходит при изменении компонента), для кнопки (Button) и надписи (Label) – событие Click (возникает при щелчке элемента управления), для Timer – событие Tick (происходит по истечении заданного временного интервала) и т. д.

Далее переходим к практической реализации теоретического материала. Для этого предлагается учащимся выполнить следующее задание:

создать проект, разместив на форме две кнопки; при нажатии на первую кнопку цвет формы должен измениться на зеленый, а при нажатии на вторую – должен измениться на цвет элемента управления, заданный цветовой схемой Windows.

Параметры проекта:

– высота и ширина формы – 580;530;

– наименование формы – «Цветовая палитра»;

– фон формы – цвет aqua;

– на первую кнопку поместить текст «Даёшь зелёный!» (тип шрифта – Courier New, цвет – зелёный, размер – 24, стиль – полужирный курсив; фон кнопки – жёлтый);

– на вторую кнопку поместить текст «Цвет элемента управления» (тип шрифта – Arial, цвет – красный, размер – 20, стиль – полужирный подчеркнутый; фон кнопки – любое изображение).

Для создания проекта выполняем команду *Файл → Новый проект → Приложение Windows Form*. Устанавливаем высоту и ширину формы, для чего задаем параметры свойства Size (580; 530), или развернем свойство, нажав значок >, введем соответствующие значения Width и Height. Нужный размер формы, как и любого другого объекта, помещаемого на форму, можно установить вручную с помощью ключевых точек, установленных в вершине нижнего правого угла, а также в центре правой и нижней стороны формы. Главное, чтобы по условию нашего задания форма соответствовала установленным параметрам.

Далее установим название формы, для чего в свойстве Text формы изменяем с помощью клавиатуры значение «Form1» на «Цветовая палитра».

Фон формы зададим с помощью изменения значения свойства BackColor. Щелкаем левой кнопкой мыши по значению Control свойства BackColor, напротив значения справа появится квадратик со значком v. Щелкаем левой кнопкой мыши по значку, откроется меню с цветовой палитрой, далее выбираем «Интернет», прокручиваем бегунок, находим нужную нам вкладку с цветом Aqua и левой кнопкой мыши щелкаем по ней. В результате форма примет соответствующий цвет.

Создаем на форме две кнопки button. В Палитре на панели компонентов «Стандартные элементы управления» левой кнопкой мыши щелкаем по объекту Button, затем щелкаем по форме. В результате на форме помещается кнопка с именем «button1». Таким же образом помещаем на форму и вторую кнопку, которая получает имя «button2».

Установим название кнопок button1 и button2, для чего в свойстве Text кнопок изменяем значение «button1» и «button2» на «Даёшь зелёный!» и «Цвет элемента управления» соответственно.

Изменим свойства шрифтов для кнопок button1 и button2.

Сначала установим тип, размер и стиль шрифта. Для чего нажимаем квадратик со значком ... (три точ-

ки, расположенные в нижней части квадрата) в поле Font и выбираем:

- 1) для кнопки button1:
 - тип шрифта – Courier New;
 - размер – 24;
 - стиль – полужирный курсив;
- 2) для кнопки button2:
 - тип шрифта – Arial;
 - размер – 20;
 - стиль – полужирный подчеркнутый.

Затем установим цвет шрифта кнопок. Для этого изменяем значение свойства ForeColor кнопок button1 и button2:

- button1 – цвет Green,
- button2 – цвет Red.

Далее устанавливаем фон кнопок. Для кнопки button1 изменяем значение свойства BackColor на Yellow. Фон кнопки button1 станет желтым. По условию задания фоном для кнопки button2 может быть любое изображение. Поэтому изменяем свойство Backgroundimage для button2. Для чего нажимаем квадратик со значком ... (три точки снизу) в поле Backgroundimage и выбираем в заранее подготовленной папке файл с изображением (желательно в формате png).

Для того, чтобы по условию задания при нажатии на первую кнопку цвет формы изменился на зеленый, а при нажатии на вторую – изменился на цвет элемента управления, заданный цветовой схемой Windows, необходимо создать обработчик события для кнопок «button1» и «button2». Основным событием кнопки является событие мыши – нажатие левой кнопки мыши (Click). Для создания обработчика события мыши для кнопки на форме щелкаем мышкой по кнопке, потом запускаем вкладку События, выбираем Click на вкладке События и выполняем двойной щелчок в поле напротив Click. После этого мы переходим на страницу с программным кодом. Для кнопки «button1» в фрагмент кода программы

```
procedure Form1.button1_Click  
(sender: Object; e: EventArgs);  
begin  
end;
```

вставим команду `BackColor := Color.Green;`

Цветом элемента управления, заданным цветовой схемой Windows, является SystemColors.Control. Поэтому в фрагмент кода программы для кнопки «button2»

```
procedure Form1.button2_Click  
(sender: Object; e: EventArgs);  
begin
```

end;

вставим команду `BackColor := SystemColors.Control;` Сохраним проект и запустим.

В итоге мы получили внешний вид формы, состоящий из 3 позиций.

1-я позиция – форма «Цветовая палитра» с фоном, с двумя кнопками с соответствующими текстами с определенным типом, размером и стилем шрифта, а также фоном;

2-я позиция – при нажатии на кнопку «button1» цвет формы изменился на зеленый;

3-я позиция – при нажатии на кнопку «button2» зеленый цвет формы изменился на цвет элемента управления, заданный цветовой схемой Windows. В нашем случае это серый цвет.

Таким образом, на первом занятии учащиеся познакомились с понятиями визуальное программирование, оконное приложение, элементы управления, объектно-ориентированное программирование, окно среды программирования, событийно-ориентированное программирование, событие, обработчик события. Теоретические знания они закрепили путем создания проекта оконного приложения. Полученные знания и умения по основам объектно-событийного программирования станут базой углубленного изучения на последующих занятиях по событийно-ориентированному программированию.

Литература

1. Бровка, Н.В. Обучение учащихся основам алгоритмизации и программирования / Н.В. Бровка, А.А. Францкевич // Весці БДПУ. Сер. 3: Фізіка. Матэматыка. Інфарматыка. Біялогія. Геаграфія. – Минск, 2020. С. 1 – 9.
2. Заборовский, Г.А. Структурно-модульное и событийно-ориентированное программирование в среде Pascal ABC / Г.А. Заборовский // Весці БДПУ. Сер. 3: Фізіка. Матэматыка. Інфарматыка. Біялогія. Геаграфія. – Минск, 2014. С. 63 – 65.
3. Ананенко, В.В. Объектно-событийное программирование / В.В. Ананенко, А.В. Михалькевич // Электронные системы и технологии: Материалы 55-ой юбилейной научной конференции аспирантов, магистрантов и студентов БГУИР. – Минск, 2019. С. 49.
4. Котов, В.М. Информатика : учебное пособие / В.М. Котов, А.И. Лапо, Ю.А. Быкадоров, Е.Н. Войтехович. – Минск, 2021. – 111 с.
5. Долинер, Л.И. Основы программирования в среде PascalABC.NET : учебное пособие / Л. И. Долинер. – Екатеринбург, 2014. – 128 с.

OBJECT-EVENT PROGRAMMING: FROM THEORY TO PRACTICE

F.S. Shumchik, E.G. Ruchaevskaya

Belarusian State University of Informatics and Radioelectronics affiliate Minsk Radioengineering College, Minsk, Belarus, zam_ur@mrk-bsuir.by

Abstract. The methodology for studying the basic concepts of object-event programming is considered, and experience in the practical implementation of theoretical material is presented using the example of creating a project.

Keywords. Visual programming, window application, form, object, event, event handler.