

ВИЗУАЛЬНОЕ ПРОГРАММИРОВАНИЕ В УЧЕБНОМ ПРОЦЕССЕ

Трушкова Любовь Андреевна, trushkova_la@mail.ru

*ГОУ ВО МО «Государственный социально-гуманитарный университет»,
г. Коломна*

Современные инструментальные программные средства, поддерживающие визуальные объектно-ориентированные технологии позволяют относительно легко и наглядно моделировать широкий круг явлений реального мира, что позволяет отнести их к операционным моделирующим педагогическим программным средствам. С другой стороны изучение в общем курсе информатики алгоритмизации и программирования в настоящее время на основе лишь классического процедурного программирования создает у студентов недостаточное представление о технологии проектирования и разработки современных программных продуктов.

Более раннее поколение программистов и преподавателей шло от программирования в кодах, к процедурному программированию на языках высокого уровня, затем к объектно-ориентированному и визуальному программированию, приобретая пользовательские навыки работы с прикладными средствами современных персональных вычислительных машин. В настоящее время оптимальным является совсем иной путь. Студент должен прийти к необходимости изучения основ алгоритмизации и программирования, уже имея пользовательские навыки, чтобы моделировать на компьютере ситуации, несколько отличающиеся от стандартных. Исходя из этого, существует необходимость методической разработки начального курса алгоритмизации и программирования на основе современных объектных и визуальных методов.

Студентам можно предложить изучение основ современных визуальных, объектных технологий программирования на примере языка

программирования для приложений офиса Visual Basic for Applications. Visual Basic for Applications не является языком объектно-ориентированного программирования в строгом смысле этого слова, но в нем широко используются элементы объектно-ориентированного подхода и связанные с ним понятия.

Также Visual Basic for Applications реализует концепцию визуального программирования, управляемого событиями. При помощи встроенного средства макрорекордер можно автоматизировать и визуализировать процесс создания кода. Компьютеру достаточно показать, что вы хотите получить в конечном счете, а макрорекордер сам переведет эти действия в программный код.

К достоинствам Visual Basic for Applications следует отнести и то, что он не требует специальных действий по установке и может использоваться на любом компьютере, на которой установлен MS Office.

В курсе информатики понятия объектно-ориентированного программирования (свойства, методы, события) рассматриваются в лабораторном практикуме на примере объектов входящих в класс Фигуры. Объект - это все, над чем может совершаться какое-либо действие или то, что имеет определенные характеристики. Почти каждый объект внутри себя имеет подобъекты, которые, в свою очередь, являются полноценными объектами и могут иметь свои подобъекты. С помощью языка программирования Visual Basic for Applications можно обратиться к любому объекту, если знать его иерархию - то есть все те объекты, чьим подобъектом он является. Свойство - это любая характеристика объекта. Большинство свойств объектов Visual Basic for Applications можно задавать программно. Многие свойства объектов также задаются через стандартные диалоговые окна, однако их всегда можно задать и в программе, а соответствующие диалоговые окна использовать при записи макроса для того, чтобы посмотреть синтаксис команды задания того или

иного свойства. Метод - это какое-либо действие над объектом. У многих методов есть параметры метода, позволяющие задать параметры действия.

Рассмотрим пример. На рабочем листе MS Excel создать две фигуры, например, звезды из группы Иллюстрации на ленте Вставка. Далее при включенном макрорекордере созданные фигуры последовательно выделяются, поворачиваются на неопределенный угол в разные стороны и сдвигаются. В результате в среде Visual Basic for Applications создается стандартный модуль, в котором генерируется следующий код.

```
Sub Макрос1()  
    ActiveSheet.Shapes("AutoShape 1").Select  
    Selection.ShapeRange.IncrementRotation 58  
    ActiveSheet.Shapes("AutoShape 2").Select  
    Selection.ShapeRange.IncrementRotation -81  
    ActiveSheet.Shapes("AutoShape 1").Select  
    Selection.ShapeRange.IncrementLeft 78.75  
    Selection.ShapeRange.IncrementTop 11.25  
    ActiveSheet.Shapes("AutoShape 2").Select  
    Selection.ShapeRange.IncrementLeft 34.5  
    Selection.ShapeRange.IncrementTop 43.18  
End Sub
```

После запуска написанного макроса первая фигура развернется на 58° по часовой стрелке, а вторая – на 81° против часовой стрелки. Здесь AutoShape 1 и AutoShape 2 - объекты, используются методы Select (выделение), IncrementRotation (поворот), IncrementLeft (положение слева), IncrementTop (положение сверху). Аналогично можно изучать свойства объектов с помощью макрорекордера, если определять при записи макроса цвет заливки, тип, толщину и цвет линии и т.д. Понятие события можно рассмотреть на примере создания управляющей кнопки из группы Элементы управления на ленте Разработчик. Связав созданную кнопку с

созданным макросом, при ее нажатии будет реализовываться событие Click (щелчок).

Используем сгенерированный код для написания программы, реализующей вращение и движение фигур. Текст программы приведен ниже

```
Sub Вращение()  
For i = 1 To 50  
    ActiveSheet.Shapes("AutoShape 1").Select  
    Selection.ShapeRange.IncrementRotation 58  
    ActiveSheet.Shapes("AutoShape 2").Select  
    Selection.ShapeRange.IncrementRotation -81  
    ActiveSheet.Shapes("AutoShape 1").Select  
    Selection.ShapeRange.IncrementLeft 78.75  
    Selection.ShapeRange.IncrementTop 11.25  
    ActiveSheet.Shapes("AutoShape 2").Select  
    Selection.ShapeRange.IncrementLeft 34.5  
    Selection.ShapeRange.IncrementTop 43.18  
    Pause (0.005)  
    Application.Calculate  
Next i  
End Sub  
Sub Pause(d As Single)  
    start = Timer  
Do While Timer < start + d  
Loop  
End
```

В приведенной программе операторы сгенерированного ранее макроса используются в цикле для организации вращения в цикле. Чтобы организовать вращение с заданной скоростью, разработана процедура

Pause. В данной процедуре используется встроенная функция Timer, которая фиксирует текущее время. Оператор Application.Calculate необходим для переноса определенных программно параметров вращения на рабочий лист MS Excel.

Аналогично в лабораторном практикуме и в самостоятельной работе студенты изучают другие визуальные эффекты, а также осваиваются понятия объектно-ориентированного программирования. В качестве контрольного мероприятия студентам предлагается разработка программы, моделирующей различные физические эффекты: закон сохранения энергии в механике, правило Ленца, опыт Фарадея и др.

Таким образом, на простых примерах студенты осваивают технологии визуального и объектно-ориентированного программирования, что несомненно способствует повышению мотивации к обучению.