УДК 377.016:004.4

Денисова Ю. А., Шестаков А. П.

СОДЕРЖАНИЕ И МЕТОДИКА ОБУЧЕНИЯ ВИЗУАЛЬНОМУ ПРОГРАММИРОВАНИЮ НА C#

Юлия Алексеевна Денисова

магистр

denisova@pspu.ru

Александр Петрович Шестаков

кандидат педагогических наук, доцент

shestakov@pspu.ru

ФГБОУ ВО «Пермский государственный гуманитарно-педагогический университет»

THE CONTENTS AND METHODS OF TEACHING VISUAL PROGRAMMING C#

Yulia Alekseevna Denisova

Alexander Petrovich Shestakov

Perm State Humanitarian Pedagogical University, Russia, Perm

Аннотация. *В статье раскрывается содержание раздела «Визуальное программирование» в рамках дисциплины «Объектно-ориентированное и визуальное программирование» для системы СПО. Даются краткие методические рекомендации для изучения визуального программирования.*

Abstract. *The article describes the content of the section «Visual programming» within the discipline «Object-oriented and visual programming» for the system of Secondary Professional Education. Short methodical recommendations for the study of visual programming are presented.*

Ключевые слова: *визуальное программирование, C#, методика обучения программированию.*

Keywords: *visual programming, C#, methods of teaching programming.*

Динамичное развитие информационных технологий, цифровизация экономики обуславливают потребность в высококвалифицированных специалистах, разрабатывающих программные приложения любой сложности. Первоначальная подготовка будущих программистов осуществляется, в том числе, на базе среднего профессионального образования.

Современные программные комплексы функционируют под управлением операционных систем с графическим интерфейсом, что определяет и их пользовательский интерфейс. Все ведущие фирмы, создающие средства для программирования и конструирования, имеют системы, поддерживающие технологию визуального программирования.

На настоящий момент язык программирования C# — один из самых мощных, быстро развивающихся и востребованных языков в IT-отрасли. Сегодня на нем создаются самые различные приложения: от небольших программ до крупных порталов и сервисов, обслуживающих ежедневно миллионы пользователей.

Таким образом, существует проблема в дидактическом наполнении дисциплин среднего профессионального образования, связанных с визуальным программированием.

Для проведения лабораторных занятий по программированию, требуются задачи и задания по каждому изучаемому разделу, желательно индивидуальные. Написано много различных пособий и учебников, позволяющих изучить теоретическую часть среды разработки, но изучение программирования не останавливается только на теоретической части, в большинстве учебной литературы для среднего профессионального образования, посвященной объектно-ориентированному программированию, присутствует существенный недостаток: немногочисленны практические задания, позволяющие читателю в полной мере освоить визуальное программирование.

Авторами предлагается комплекс практических заданий для освоения визуального программирования средствами C#, предназначенный для студен-

тов среднего профессионального образования, обучающихся по технологическим и физико-математическим профилям (укрупненная группа специальностей 09). Комплекс рассчитан на следующие начальные знания обучающихся:

- базовые знания информатики и математики;
- знание основ логики;
- знание алгоритмизации и программирования С-подобных структурных языков;
- умение программировать в объектно-ориентированном стиле;
- умение разрабатывать программы в консольном режиме C#.

Комплекс разделен на восемь разделов:

- базовые компоненты;
- графика;
- файлы;
- базы данных;
- мультимедиа;
- COM-объекты;
- сетевые компоненты;
- создание собственного компонента.

Изучение учебного материала в рамках каждого раздела происходит в два этапа.

Первый этап — ознакомление с определенным разделом программирования на C#. Этот этап включает задачи с разобранным решением (алгоритмом решения) и программным кодом с подробными комментариями, рассмотрев которые, читатель сможет детально ознакомиться с подходами к решению задач указанного раздела.

Второй этап — закрепление полученных знаний. Он предполагает самостоятельное решение задач раздела.

Обсудим подробнее данный комплект дидактического обеспечения.

Первый раздел — базовые компоненты. В данном разделе демонстрируются примеры оформления интерфейса, внешнего вида программы, они предназначены для изучения свойств компонентов, обеспечивающих ввод–вывод информации. Задачи сопровождаются изображениями и таблицами позволяющие наглядно продемонстрировать, варианты оформления интерфейса, что обеспечивает изучение компонентов одновременно, показывает их взаимодействие друг с другом.

Изучение следует начинать с рассмотрения задачи с использованием стандартных элементов управления, которая приведена с кодом решения, пояснением и комментариями основных действий, что обеспечивает наглядность изучения строения интегрированной среды разработки. В ней рассматриваются элементы и их свойства, события и методы. Это обеспечит закрепление знаний основных свойств объектно-ориентированного программирования, таких как инкапсуляция (свойства и методы компонента объединены), полиморфизм (свойства, и методы имеют общее имя, но выполняются по-разному), наследственность (обнаружение общих свойств и методов компонента).

Для закрепления полученных навыков необходимо перейти задачам для самостоятельной работы. Самостоятельная работа является одним из эффективных средств развития и активизации творческой, познавательной и интеллектуальной деятельности студентов. Решение задач позволяет закрепить знания, полученные при разборе задач на первом этапе, и развить навыки по работе с элементами управления.

Существует и другой подход к изучению данной темы. Это решение задач, разделенных на группы по конкретным темам, т. е. рассматривается первая задача на компоненты Label, Edit, Form, Button, которая приведена с подробным решением. Следующие задачи — с частичным решением, потом нерешенные задачи. У такого способа изучения присутствует недостаток, нельзя в полной мере рассмотреть взаимное применение компонентов, т.е. разработать удобный, красивый интерфейс программы.

На этом изучение основных компонентов не заканчивается, они в дальнейшем будут использоваться во всех остальных задачах.

Следующим разделом является графика. В любых современных программных приложениях уделяется большое внимание внешнему виду, графическому дизайну. Одним из способов представления, может быть создание на его форме каких-либо рисунков с помощью анимации. Изучение данного раздела начинается с примитивов графики: предлагается нарисовать флаг олимпийских игр. Фрагменты флага — пять колец, которые выводятся на элемент PictureBox при использовании функции DrawEllipse.

Очередная задача — вывод на форму графика с помощью примитивов. Приводится код программы с автоматическим подбором масштаба (также автоматически подбирается цена деления осей).

Еще одна рассматриваемая задача — использование компонента Chart для вывода диаграммы, также здесь впервые используется элемент DataGridView, он отображает строки и столбцы данных в сетке, это необходимо для ввода и хранения данных, которые используются для построения диаграммы.

Однако, C# не ограничивается только графическими примитивами. C# предоставляет большую возможность для создания различной анимации. Например, задача на создание мультипликационных эффектов. Идея заключается в следующем: создается простой рисунок с помощью графических примитивов, по команде стирается и рисуется уже в другом месте, тем самым создается иллюзия движения рисунка.

Также C# предоставляет возможность любому пользователю создать свой рисунок. Существует два режима: первый — произвольное рисование, второй — создание какой-либо фигуры. При произвольном рисовании, если пользователь нажал кнопку мыши, каждый раз вычисляются координаты и соединяются с предыдущей координатой линией. Если используется режим рисования объектов, то при нажатии запоминается начальная точка, при повторном нажатии — конечная точка, между ними рисуется необходимый объект.

Изучение необходимо начать с рассмотрения задач на линейную графику, что способствует формированию способов построения изображения. Далее изучается работа на построение графиков и диаграмм. После, необходимо познакомить студентов со способами построения графического редактора, с возможностями рисования различными инструментами и созданием простой анимации. Особое внимание здесь необходимо обратить на способ построения иллюзии движения, а также использование и создание градиентной заливки. Также применяются диалоговые окна для открытия, сохранения файлов.

Раздел файловая система и исключения — один из наиболее важных в программировании, так как рано или поздно придется столкнуться с необходимостью долговременного хранения полученной информации. Сохранение информации происходит в файлах. Этот раздел посвящен работе с файлами, каталогами и обработке исключительных ситуаций.

Изучению файлов предшествует тема, связанная с обработкой исключительных ситуаций. Для этого студентам предложена задача, которая помещена в отдельный раздел, потому что последующие задачи должны сопровождаться перехватом исключительных ситуаций, возникающих при написании программ.

После полного освоения способов перехвата исключительных ситуаций, можно приступить к непосредственной работе с каталогами. Наиболее частыми действиями над каталогами является создание, удаление, переход в каталог и указание текущей папки.

Далее рассматриваются основные действия над файлами, такие как поиск файла, создание и запись во временный файл, копирование файла.

Поиск файла осуществляется рекурсивно, функция просматривает иерархию папок, начиная с той, полный путь к которой передан ей, обходит все вложенные подпапки и заполняет элемент ListBox списком файлов, которые соответствуют маске поиска.

При работе с временным файлом необходимо показать создание, заполнение и расположение файла. А также генерацию имени, которое не будет совпадать с другими файлами. Заполнять файл можно вручную, а также с помощью генерации текста, который подключается из Интернет-ресурса, подробно данная функция рассмотрена в одной из задач.

Следующим шагом будет рассмотрение одной из наиболее часто применяемых операций над файлами — копирование. В C# существует несколько способов организации копирования файла, в данном разделе было описано два типа: первый — с помощью встроенной функции C# CopyFile; и второй — создание собственного метода копирования посредством файловых потоков.

При решении задач раздела «Файловая система и исключения» могут допускаться ошибки записи, чтения файла, существования файла, поэтому первоначально необходимо полностью изучить соответствующие исключительные ситуации, что позволит в дальнейшем уменьшить вероятность некорректного завершения приложения. Особое внимание здесь следует обратить на такие задачи, как поиск файла, копирование файла и работа с каталогами.

Следующий раздел посвящен работе с базами данных. Хранение информации в файлах можно организовать по–разному, одним из способов является хранение в виде баз данных. C# предоставляет широкие возможности для создания и работы с базами данных. В изучении данного раздела, задачи разделены на следующие группы:

- основные компоненты для работы с базами данных;
- основы языка SQL;
- создание базы данных MySQL;
- организация работы с базами данных посредствам C#.

Чтобы начать изучение работы с базами данных, необходимо ее создать. Для начала создается реляционная база данных MySQL, которая будет использоваться во всех последующих задачах данного раздела.

Изначально рассматривается возможность подключения и изменения созданной базы данных. Важно подчеркнуть внимание не только на отображение данных в приложении, но и сохранение всех действий над ней в реальной базе данных.

Следующая задача — фильтрация данных по определенным диапазонам. Для быстрого доступа к данным их необходимо представить в порядочном виде, по какому-либо параметру. Создается специальная панель, которая фильтрует данные с помощью SQL запросов — И, ИЛИ.

Далее рассматриваются задачи на основные действия с базами данных: поиск, сортировка, вставка и удаление записей.

В заключении раздела рассмотрена работа со связными таблицами, которая показывает связь отношений в базе данных.

Изучение данного раздела рекомендуется строго по выделенным блокам, т. к. организация базы данных сопровождается сложными программными действиями, и для качественного овладения, нужно изучить сначала один из блоков, а потом приступать к изучению другого по аналогии с предыдущим, что позволит качественно и быстро изучить объектно-ориентированный подход к программированию баз данных.

Следующий раздел рассматривает работу с мультимедиа. Иногда возникает необходимость разнообразить программы, сделать их более привлекательными. Один из способов — это мультимедиа, анимационные эффекты, сопровождаемые или не сопровождаемые звуком. В разделе мультимедиа рассматривается два типа анимации: простая анимация, не сопровождаемая звуком и анимационные эффекты со звуком (AVI клипы). А также работа только со звуком (MP3 плеер). Соответственно, рассмотрены такие компоненты:

- Метод ImageAnimator (анимация без звука);
- WMP (Windows Media Player) (подключаемый элемент для просмотра видео);
- Winmm (DLL библиотека для работы с аудиофайлами).

Первый компонент рассчитан на простые анимационные ролики, которые выводят анимированное изображение на экран. Изображение создается из анимированного GIF-файла, находящегося в той же папке, что и приложение или в ресурсах проекта. Такие анимации не сопровождаются звуком. Они могут украшать приложение при запуске (например, заставка программы), вызове справки и т. п.

Подключаемый элемент Windows Media Player. С его помощью можно проигрывать файлы любых форматов, поддерживаемых данной системой.

Существует единый набор команд для воспроизведения аудиофайлов с помощью любой звуковой карты. В операционной системе Windows функции, предназначенные для выполнения мультимедийных операций, находятся в библиотеке WinMM.dll. Соответственно, чтобы MP3-плеер мог воспроизводить аудиофайлы, необходимо подключить данную библиотеку и работать с ней.

Раздел следует начинать с работы метода ImageAnimator, который очень прост в обращении, что позволит, во-первых, понять, как устроены программы, работающие с анимацией, во-вторых, заинтересовать студентов данной темой, так как результат получить очень просто и быстро. После изучения, можно переходить к Windows MediaPlayer и библиотеке Winmm. Таким образом, можно добиться хороших программных результатов.

Написать все приложения, удовлетворяющее требованию пользователя, невозможно, иногда хочется использовать сторонние приложения в собственной разработке. C# предоставляет такую возможность благодаря COM-объектам.

Объектная модель программных компонентов (Component Object Model, COM) проектировалась с целью дать возможность создавать компоненты на любом языке/платформе, обладающем поддержкой этой модели, и использовать их в любом языке/платформе (другом), также обладающем поддержкой COM. Платформа.NET не исключение и позволяет легко использовать сторонние COM-объекты и экспортировать типы.NET в виде COM-объектов.

COM-объекты используется для обеспечения межпроцессного взаимодействия и создания динамических объектов на широком спектре языков программирования. В C# есть свой набор стандартных COM-объектов. В данном разделе показано, как подключать, использовать и взаимодействовать с такими объектами. Все задачи направлены на изучение использования сторонних объектов.

Чтобы подключить какой-либо объект, необходимо в обозревателе решений добавить ссылку на него, после добавить уже сам компонент (при необходимости) на панель элементов.

Изначально рассматривается работа с PDF Adobe Reader, в ней детально акцентировано внимание на подключение COM-объектов. Вторая и третья задача рассматривает работу с компонентами Microsoft Excel и Word.

Один из самых интересных и увлекательных разделов, так с его помощью при небольших затратах, можно получить быстро качественный продукт. В разделе присутствует небольшая теоретическая информация, которая облегчит работу не только со следующими задачами, но и работу с C# и интегрированной средой разработки в общем. Особое внимание надо уделить подключению объекта.

Компьютер не представлял бы большой ценности, если не было бы возможности обмена информации между компьютерами. Данную проблему можно легко решить, используя сетевые программные продукты. В следующем разделе показано, как быстро и легко создать небольшие сетевые приложения.

Изучение раздела начинается с описания протокола SMTP, который осуществляет отправку почты в среде Интернет. Данный протокол указывает, как почтовые сервера взаимодействуют при передаче электронной почты.

Следующая задача рассматривает работу с очередью сообщений, которая является компонентом Windows. Администратор домена создает очередь сообщений, и пользователи могут прислать и получать сообщения используя эту очередь. Особенность заключается в том, что пользователю не нужно быть

в сети в тот момент, когда ему отправляется сообщение, эти сообщения будут хранится на сервере и когда пользователь подключится к домену он сможет их прочитать.

Также в разделе рассмотрена работа с сокетами TCP, дана интересная задача обмена сообщения между клиентом и сервером, которые зашифрованы с помощью технологии DES (Data Encryption Standard).

Данный раздел необходимо начинать с теоретической части, так как он содержит много специфических подразделов. Если студенты не могут овладеть с помощью справочной подборки и подробно разобранных задач, которые даны в комплекте, необходимо использовать дополнительную литературу. Рекомендуется на практическом занятии подробно разобрать вместе со студентами задачу посвященую работе с сокетами TCP, которая представлена с полным решением. После этого можно дать на самостоятельное решение задачи подобной тематики, что позволит студентам понять принцип работы программы. После этого можно переходить на другие задачи, так как все задачи данного раздела разобраны, то порядок их изучения особо не важен.

При программировании в визуальных средах можно встретиться с такой проблемой, как нехватка какого-либо компонента. Поэтому возникает потребность разработать свой компонент. Для этого существует специальный раздел «собственные компоненты». В нем подробно рассматривается, как создается компонент, с чего необходимо начать.

Раздел разбит на две темы — создание визуального компонента и создание невизуального компонента.

Собственные элементы управления предоставляют средства для создания и повторного использования настраиваемых графических интерфейсов. Собственный элемент управления — это компонент, имеющий визуальное представление. Таким образом, он может состоять из одного или нескольких элементов управления, компонентов или блоков кода Windows Forms, позволяющих расширить функциональные возможности за счет проверки введенных пользователем данных, изменения свойств отображения или выполнения

других предусмотренных разработчиком действий. Собственные элементы управления можно вставлять в Windows Forms точно так же, как другие элементы управления.

Данная тема очень важна, так как она определяет алгоритм создания компонента или класса.

Визуальный компонент разрабатывается на основе уже существующего класса, функциональность которого специализируется или расширяется путем изменения имеющихся и добавления новых свойств и методов. Программирование компонента, в отличие от визуального конструирования классического приложения, сводится в основном к написанию исходного кода, для чего совершенно необходимо более тесное знакомство с объектно-ориентированным программированием [2].

Следующая задача посвящена созданию невизуального компонента, назначение которых — инкапсулировать некоторый часто используемый код.

На этапе проектирования приложения предоставляется доступ до некоторых функций, методов, полей, которые объединены общим целым. Добавляется сборка, которая содержит компонент. После добавления она становится доступной, и можно объявить какую-либо переменную или поле класса этого компонента. Также необходимо получить все необходимые доступы до методов и полей этого компонента. Основная задача, выполняемая невизуальными компонентами — инкапсуляция участков кода, выполняющих какие-либо функции и имеющих набор входных параметров, влияющий на их поведение.

В пределах темы построения невизуальных компонентов рассматривается пример определение треугольника по трем сторонам, по двум сторонам и углу между ними, либо по одной стороне и двум прилежащим углам, а также его использование в тестовом приложении.

Из всех представленных разделов этот самый сложный для понимания, но в тоже время является ключевым в объектно-ориентированной методологии программирования. Изучение должно начинаться исключительно с озна-

комительной части, вся теория должна сопровождаться практическими примерами, что позволит обеспечить наглядность в обучении. После полного освоения можно приступать к практическим заданиям, под четким руководством преподавателя, так как у студентов могут возникать ошибки, это связанно с тем, что большую часть кода они должны прописывать самостоятельно. Когда студенты усвоят простые задачи, то можно переходить к более сложным. Порядок изучения можно начинать с любой задачи. При изучении данного раздела необходим контроль преподавателя, после успешного выполнения нескольких заданий контроль можно ослабить.

Таким образом, в разработанном дидактическом комплекте в поддержку изучения визуального программирования средствами C# насчитывается 18 задач с разобранным решением и 185 задач для самостоятельной работы, что позволяет в достаточной мере познакомиться с основными возможностями визуального программирования.

Список литературы

1. ФГОС СПО 09.02.07 Информационные системы и программирование [Электронный ресурс]. – Введ. 09.12.2016 г. – Режим доступа: www.edu.ru.

2. Создание Windows-приложений на основе Visual C#: Лекция [Электронный ресурс] / НОУ «ИНТУИТ» – URL: http://www.intuit.ru/studies/courses/106/106/info (дата обращения 16.03.2018).

3. Павловская, Т. А. C#. Программирование на языке высокого уровня: учебник для вузов / Т. А. Павловская. – Санкт-Петербург : Питер, 2009. – 432 с.

4. Программирование на C# : лабораторный практикум [Электронный ресурс] / сост. И. П. Половина ; Перм. гос. гуманит.-пед ун-т. – Пермь, 2013. – 1 электрон. опт. диск (CD ROM).