

И.Е. ЖИГАЛОВ, М.И. ОЗЕРОВА

«Программирования в среде Delphi»

Учебное пособие

Допущено Учебно-методическим объединением вузов по университетскому политехническому образованию в качестве учебного пособия для студентов высших учебных заведений, обучающихся по направлению подготовки бакалавра 09.03.02 и магистра 09.04.02 Информационные системы и технологии 230400)

Рецензенты:

Доктор технических наук, профессор кафедры
специальной техники и информационных технологий
Владимирского юридического института Федеральной
службы исполнения наказаний

Б.Ю. Житников

Кандидат экономических наук, доцент кафедры управления
и информатики в технических и экономических системах
Владимирского государственного университета

Д.А. Градусов

Доктор технических наук, профессор, заведующий кафедрой информационных
технологий Владимирского филиала Российской академии народного хозяйства и
государственной службы при Президенте Российской Федерации

А. С. Шалумов

Озерова М. И., Жигалов И.Е.

О-46 Программирования в среде Delphi: Учебное пособие / Озерова М. И., Жигалов
И.Е. — М. : Lennex Corp, — Подготовка макета: Издательство Нобель Пресс,
2014. — 110 с.

ISBN 978-5-519-01883-8

В учебном пособии в виде практических работ представлен система-
тизированный материал по основам программирования на языке Delphi,
лабораторные работы содержат теоретический материал, практическое задание и
контрольные вопросы.

Предназначено для студентов, изучающих среду программирования Delphi в
рамках дисциплин «Основы алгоритмизации и программирования», «Технология
программирования», «Информатика», «Инструментальные средства И.С.»
обучающихся по направлению 230400.65 (бакалавр) «Информационные системы и
технологии».

Может быть, полезно студентам других специальностей, а также широкому
кругу читателей, самостоятельно осваивающих среду программирования Delphi

Рекомендовано для формирования профессиональных компетенций в
соответствии с ФГОС 3-го поколения. Библиогр.: 3 назв.

УДК 004.43
ББК 21.183.49

ISBN 978-5-519-01883-8

© Издательство Нобель Пресс, 2014
© Озерова М. И. Жигалов И.Е., 2014

ПРЕДИСЛОВИЕ

В последнее время появилось мнение, что среда Delphi на сегодняшний день уже не актуальна. Однако это далеко не так. Язык Delphi является востребованным. Это строго типизированный объектно-ориентированный язык, в основе которого лежит хорошо знакомый программистам Object Pascal. С помощью Delphi создаются приложения для операционной системы Windows. Среда Delphi легко расширяется установкой дополнительных модулей. Пользовательский интерфейс также хорошо настраиваемый, именно поэтому изучение среды программирования Delphi является одной из основных задач при подготовке кадров по направлению 230400.62 (бакалавр) «Информационные системы и технологии».

С 14 февраля 1995 года, когда Software Development Conference 95 West компания Borland презентовала первую версию Delphi и по настоящее время вышло уже 18 версий носит название Delphi. К сожалению, на версии Delphi7 история Delphi в каком смысле завершилась. В 2008 году Borland продала подразделения, занимающиеся средствами разработки и сервером баз данных InterBase, компании Embarcadero Technologies. Одной из последних версий является Delphi XE4, которая поддерживает языки программирования Delphi и C++. Данная версия разработана в соответствии с концепцией визуального программирования. Delphi XE5 ориентирована на создание native-приложения для Android и iOS.

Однако именно Turbo Delphi, анонсированная в 2006 году, была ориентирована в первую очередь на студентов, индивидуальных пользователей и программистов-любителей. Основное отличие данного продукта от других продуктов CodeGear — наличие бесплатной версии Turbo Delphi Explorer. Именно поэтому данная среда наиболее удобна для обучения.

В зависимости от версии Delphi основное окно имеет различный вид, но функциональное назначение остается аналогичным, поэтому в учебном пособии описаны алгоритмы и приведены практические примеры, которые позволяют изучить основы программирования в среде Delphi независимо от версии.

В пособии представлен учебный материал, позволяющий студентам в ходе выполнения практических заданий изучить структуру Delphi, создать и отладить простейшие линейные алгоритмы, реализованные на Delphi. Так же данное пособие можно использовать для самостоятельной работы студентов при выполнении курсовых работ по дисциплинам «Программирование компьютерной графики», «Информационные основы дизайна».

Учебное пособие состоит из введения, справочной информации и лабораторных работ. Каждая лабораторная работа состоит из теоретического материала, практического задания, примеров выполнения задания, комментариев об ошибках, контрольных вопросов, практических заданий и требований к содержанию отчета. Рассмотрен материал по структуре Delphi, разобраны основные элементы языка Object Pascal.

Введение

«Основные понятия объектно-ориентированного программирования. Система визуального объективно-ориентированного проектирования Delphi»

Цель работы:

Изучить основные понятия объектно-ориентированного программирования. Познакомиться с системой визуального объективно-ориентированного проектирования Delphi.

Теоретическая часть:

В объектно-ориентированном программировании главной отправной точкой при проектировании программы является не процедура, не действие, а *объект*. Такой подход достаточно естественен, поскольку в реальном мире нас окружают именно объекты, взаимодействующие друг с другом.

Объектно-ориентированное программирование базируется на трех основных принципах: *наследование, инкапсуляция и полиморфизм*. Программа, построенная по этим принципам, — это не последовательность операторов, не некий жесткий алгоритм, а совокупность объектов и способов их взаимодействия. Обмен информацией между объектами происходит посредством *сообщений*.

Объекты

Назовем объектом понятие, абстракцию или любой предмет с четко очерченными границами, имеющий смысл в контексте рассматриваемой прикладной проблемы. Все объекты можно идентифицировать, между объектами можно установить отношение тождества (два Сидоровых Ивана — это разные люди, но они студенты одной группы).

Классы объектов

Классом называют особую структуру, которая может иметь в своем составе поля, методы и свойства. Класс выступает в качестве объектного типа данных, а объект – это конкретный *экземпляр* класса. Например, два Сидоровых Ивана принадлежат одному и тому же классу объектов, они — студенты группы ТКС-206. Именно с этим связана их одинаковость (одинаковый шифр группы, одно расписание занятий и т.д.).

Каждый конкретный класс имеет свои особенности поведения и характеристики, определяющие этот класс. Например,

Геометрический объект			
Объемный	Плоский		
	С вершинами	Без вершин	
		Окружность	Эллипс

Наивысший уровень – самый общий и самый простой, каждый последующий уровень более специфический и менее общий. На самом последнем уровне можно определить цвет, стиль заполнения, величину радиуса окружности и т.п.

Если характеристика уже однажды определена для более высокого уровня, то все уровни, расположенные ниже имеют ту же характеристику (если уж определена окружность, понятно, что вершин у нее нет).

Таким образом, классы-наследники могут наследовать характеристики классов-родителей.

Свойства

Свойства – перечень параметров объекта, которые определяют внешний вид и поведение объекта, выделяют уникальные особенности каждого экземпляра. К свойствам относятся: имя, тип, значение, цвет, размер и др. *Состояние* – совокупность всех свойств данного объекта.

Методы

Метод — это некоторое действие (операция), которое можно выполнять над данным объектом. В результате этого действия в объекте что-нибудь меняется (например, местоположение, цвет и др.). Другими словами можно еще сказать, методом называется команда, которую может

выполнять объект. Для каждого класса объектов имеется свой перечень методов, которые можно к нему применить или которые он может выполнить. Например, объект можно удалить с экрана, переместить в другое место.

События

Каждый объект способен реагировать на определенные события – это разновидность свойства объекта. При возникновении события производится его обработка.

События – сигналы, формируемые внешней средой, на которые объект должен отреагировать соответствующим образом.

Средой взаимодействия объектов являются *сообщения*, генерируемые в результате наступления различных *событий*.

События наступают в результате действий пользователя – перемещение курсора пользователя, нажатия кнопок мыши или клавиш на клавиатуре, а также в результате работы самих объектов. Для каждого объекта определено множество событий, на которые он может реагировать. Для конкретных экземпляров объекта могут быть определены *обработчики* каких-то из этих событий, которые и определяют реакцию данного экземпляра объекта.

Объект можно определить как совокупность свойств и методов, а также событий, на которые он может реагировать.

Внешнее управление объектом осуществляется через обработчик событий. Эти обработчики обращаются к методам и свойствам объекта.

Инкапсуляция

С одной стороны объект, обладает определенными свойствами, которые характеризуют его состояние в данный момент. С другой стороны над объектами возможны операции, которые приводят к изменению этих свойств. Доступ к изменению свойств осуществляется только с помощью методов, присущих данному классу объектов. Есть метод, данное свойство данного объекта можно изменить, нет метода – нельзя. Методы как бы «окружают» свойства объекта, говорят, что свойства «инкапсулированы» в объект. Для обеспечения инкапсуляции класс не должен позволять прямого доступа к своим данным. *Инкапсуляция* — механизм скрытия всех внутренних деталей объекта, не влияющих на его поведение.

Наследование

Классы-наследники могут наследовать характеристики классов-родителей. Т.е. один объект приобретает свойства другого объекта, добавляя к ним свойства, характерные только для него.

Наследование определяет отношение между классами: объекты класса-наследник обладают всеми свойствами и методами объектов класса-родитель и не должны их повторно реализовывать.

Класс Точка (родитель)		Класс Окружность (наследник)	
Свойства	Методы	Свойства	Методы
Координаты (x,y)	Перемещение	Координаты центра (x, y)	Перемещение
Цвет	Изменение цвета	Цвет	Изменение цвета
		Радиус	Изменение радиуса

Полиморфизм (имеющий много форм)

К объектам разных классов можно применять один и тот же метод, вот только действовать этот метод будет по-разному. Например, к большинству объектов в Windows&Office можно применять одни и те же методы: копирование, перемещение, переименование, удаление и т.п. Однако, механизмы реализации этих методов для разных классов (файл в Windows и документ Word) неодинаковы.

Полиморфизм – возможность использования одних и тех методов для объектов разных классов, только реализация этих методов будет *индивидуальной* для каждого класса.

Система визуального объективно-ориентированного проектирования Delphi

Delphi – это среда разработки программ, ориентированных на работу в Windows. В основе идеологии Delphi лежат методология объектно-ориентированного программирования и технология визуального проектирования.

Работа производится в Интегрированной среде разработки (ИСР) Delphi, которая предоставляет пользователю формы, где размещаются с помощью мыши необходимые компоненты, имеющиеся в библиотеке Delphi. С

помощью простых манипуляций мышью можно изменять размеры и расположение этих компонент. При этом в процессе проектирования можно постоянно видеть результат – изображение формы и расположенных на ней компонент. А самое главное заключается в том, что во время проектирования формы редактор кода Delphi автоматически генерирует код программы, включая в нее соответствующие фрагменты, описывающие данный компонент. В соответствующих диалоговых окнах можно изменить заданные по умолчанию свойства компонент и, при необходимости, написать обработчики событий.

Компоненты могут быть визуальными, видимыми при работе приложения, и невидимыми, выполняющим некоторые служебные функции, они отображаются в виде значка в процессе проектирования и не видны при работе приложения.

Система визуального объективно-ориентированного проектирования Delphi позволяет:

- создавать законченные приложения для Windows самой различной направленности, от чисто вычислительных и логических, до использующих графику и мультимедиа;

- быстро создавать профессионально выглядящий оконный интерфейс для любых приложений, написанных на любом языке; интерфейс удовлетворяет всем требованиям Windows и автоматически настраивается на ту систему, которая установлена на компьютере пользователя, поскольку использует функции, процедуры и библиотеки Windows;

- создавать свои динамически присоединяемые библиотеки (Dll) компонент, форм, функций, которые затем можно использовать из других языков программирования;

- создавать мощные системы работы с локальными и удаленными базами данных любых типов;

- формировать и печатать сложные отчеты, включающие таблицы, графики и т.п.;

- создавать справочные системы (файлы.hlp), как для своих приложений, так и для любых других, с которыми можно работать не только из приложений, но и просто из Windows;

- создавать профессиональные программы установки для приложений Windows, учитывающие всю специфику и все требования операционной системы.

Включать объекты в свою программу пользователь может вручную, используя соответствующие операторы, или путем визуального программирования, используя заготовки – компоненты.

Контрольные вопросы:

1. В чем заключается основное отличие процедурного и объектно-ориентированного программирования?
2. Назовите основные принципы объектно-ориентированного программирования
3. Что такое объект?
4. Что такое класс?
5. В чем заключается наследование?
6. Что такое инкапсуляция и для чего она применяется?
7. В чем заключается полиморфизм операций?
8. Что такое метод?
9. Что такое свойство объекта?
10. Что такое события?
11. Каково назначение сообщений?
12. В чем заключается преимущества визуального программирования интерфейса?

Содержание отчета:

1. Название, цели лабораторной работы
2. Формулировка практической части работы
3. Выводы по работе
4. Ответы на контрольные вопросы