

آموزش گام به گام

ویژوال بیسیک

جلد اول

نویسنده: اوانجلوس پتروسوس

مترجم: مهندس سامان هزارخانی



نشر گستر





سرشناسه	: پتروئوس، اوانجلوس	Petroutsos, Evangelos
عنوان و نام پدیدآور	: آموزش گام به گام ویژوال بیسیک ۲۰۱۰ نویسنده گان اوانجلوس پتروئوس؛ مترجم سامان هزارخانی.	
مشخصات نشر	: تهران: نشر گستر، ۱۳۹۰.	
مشخصات ظاهری	: ۵۳۳ص: مصور، جدول.	
شابک	: ۹۷۸۶۰۰۵۸۸۳۳۴۳ ریا: ۱۲۰۰۰۰	
یادداشت	: عنوان اصلی: Mastering Microsoft Visual Basic 2010	
موضوع	: ویژوال بیسیک مایکروسافت	
موضوع	: بیسیک (زبان برنامه نویسی کامپیوتر)	
شناسه افزوده	: هزارخانی، سامان، ۱۳۵۸ - مترجم	
شناسه افزوده	: فلاح مشفق، کیوان، ۱۳۴۵ - ویراستار	
رده بندی کنگره	: ۱۷۱۳۹۰ پ ۹ و ۱۷۳/۷۳۴ QAY۶	
رده بندی دیویی	: ۰۰۵/۲۷۴۸	
شماره کتابشناسی ملی	: ۲۲۸۲۸۱۱	



نشر گستر

نام کتاب	: آموزش گام به گام ویژوال بیسیک ۲۰۱۰ (جلد اول)
نویسندگان	: اوانجلوس پتروئوس
مترجم	: مهندس سامان هزارخانی
ویراستار	: مهندس کیوان فلاح مشفق
نوبت چاپ	: اول
تاریخ چاپ	: ۱۳۹۰
تیراژ	: ۱۵۰۰ جلد
چاپ	: پڑمان
قیمت	: ۱۲۰۰۰ تومان
ناشر	: انتشارات نشر گستر ۶۶۴۳۲۷۰۵
مرکز پخش	: انتشارات نشر گستر (۳-۶۶۵۹۱۵۶۱)

فهرست مطالب

مقدمه

۲۰	ویژگی‌های این کتاب
۲۰	این کتاب برای چه کسانی است؟
۲۲	موضوعات پیشرفته‌تر
۲۳	ساختار کتاب
۲۶	ضمیمه‌ها را از دست ندهید
۲۶	دانلود کردن کد این کتاب

بخش اول: ویژوال بیسیک، زبان برنامه‌نویسی

۶۹	فصل ۱: شروع به کار با ویژوال بیسیک ۲۰۱۰
۳۰	بررسی محیط برنامه‌نویسی یکپارچه ویژوال بیسیک
۳۰	صفحه تصویر شروع به کار ویژوال استودیو
۳۳	شروع یک پروژه جدید
۳۶	کار با محیط Windows Form Designer
۴۴	اولین برنامه ویژوال بیسیک شما
۵۰	برنامه را کاربرپسندتر کنید
۵۴	آشنایی با اجزای IDE
۵۴	منوهای IDE
۵۵	منوی File
۵۵	منوی Edit
۵۵	منوی فرعی Edit→Advanced
۵۵	منوی فرعی Edit→IntelliSense
۵۷	منوی فرعی Edit→Outlining
۵۸	منوی View (نمایش)
۵۸	منوی Project
۵۹	منوی Build (ایجاد)
۵۹	منوی Debug (اشکال‌زدایی)
۵۹	منوی Data (داده‌ها)
۵۹	منوی Format (شکل‌دهی)

۵۹	منوی Tools (ابزارها)
۶۰	منوی Window (پنجره)
۶۰	منوی Help
۶۰	پنجره Toolbox (جعبه ابزار)
۶۰	پنجره Solution Explorer (مرورگر سولوشن)
۶۱	پنجره Properties
۶۲	پنجره Output (خروجی)
۶۲	پنجره‌های Command و Immediate (فوری)
۶۳	پنجره Error List (لیست اشتباهات)
۶۴	تعیین گزینه‌های محیط کاری
۶۶	ساخت برنامه‌های کنسولی
۶۸	استفاده از اسنپیت کدها (code snippets)
۷۰	استفاده از کامپوست My
۷۳	خلاصه و تمرین

فصل ۲: مدیریت داده‌ها

۷۵	متغیرها
۷۶	معرفی کردن متغیرها
۷۷	قرار دادهای نامگذاری یک متغیر
۷۸	مقداردهی یک متغیر
۷۹	انواع متغیرها
۸۰	متغیرهای عددی (numeric)
۸۰	متغیرهای نوع عدد صحیح (integer)
۸۰	اعداد با دقت منفرد (Single) و مضاعف (Double)
۸۳	نوع دادهای Decimal (اعشاری)
۸۵	بی‌نهایت و سایر موارد عجیب
۸۷	تست برای Infinity (بی‌نهایت) و NaN (یک عدد نیست)
۸۸	متغیرهای بایتی (Byte)
۸۹	متغیرهای Boolean (منطقی)
۹۰	متغیرهای رشتهای (String)
۹۱	متغیرهای کاراکتری (Character)
۹۴	متغیرهای Date (تاریخ)
۹۶	گزینه‌های Strict, Explicit و Infer
۱۰۰	تعیین اتوماتیک نوع متغیر
۱۰۱	متغیرهای نوع Object (شیء)
۱۰۳	متغیرها به عنوان اشیاء
۱۰۴	تبدیل کردن انواع متغیرها به یکدیگر
۱۰۶	تبدیلات وسعت‌دهنده و محدودکننده
۱۰۷	فرمت (شکل‌دهی) اعداد
۱۰۹	رشته‌های فرمت عددی استاندارد
۱۱۰	رشته‌های فرمت عددی تصویری

۱۱۱	انواع داده‌ای تعریف شده توسط کاربر
۱۱۳	تعیین نوع متغیر
۱۱۴	این یک عدد است یا یک رشته یا یک تاریخ؟
۱۱۵	محدوده یک متغیر
۱۱۷	طول عمر یک متغیر
۱۱۸	ثابت‌ها
۱۱۹	آرایه‌ها
۱۱۹	معرفی آرایه
۱۲۱	دادن مقادیر اولیه به عناصر آرایه‌ها
۱۲۱	حدود آرایه‌ها
۱۲۲	آرایه‌های چند بعدی
۱۲۵	کلکسیون‌ها
۱۲۷	خلاصه و تمرین

فصل ۳: مبانی برنامه‌نویسی ویژوال بیسیک

۱۳۰	عبارت‌های کنترل جریان
۱۳۰	عبارت‌های تصمیم‌گیری
۱۳۰	عبارت If...Then
۱۳۱	عبارت If...Then...Else
۱۳۵	عبارت Select Case
۱۳۶	عبارات ارزیابی مدار کوتاه
۱۳۸	عبارات تعریف حلقه
۱۳۸	حلقه‌های For...Next
۱۴۰	حلقه‌های For Each...Next
۱۴۲	حلقه‌های DO
۱۴۴	حلقه‌های While
۱۴۶	ساختارهای کنترلی تودرتو
۱۴۸	عبارت‌های Exit (خروج) و Continue (ادامه)
۱۴۹	نوشتن پراسیجرها و استفاده از آن‌ها
۱۵۰	ساب‌روتین‌ها
۱۵۱	توابع
۱۵۴	آرگومان‌ها
۱۵۶	مکانیزم‌های ارسال آرگومان به توابع
۱۵۶	مقایسه "ارسال مقدار" با "ارسال اصل" آرگومان
۱۵۷	برگرداندن چند مقدار از یک تابع
۱۵۹	توابع از پیش تعریف شده در ویژوال بیسیک
۱۶۱	توابع تعریف شده توسط کاربر
۱۶۴	ارسال آرگومان‌ها به توابع و گرفتن مقادیر بازگشتی آنها
۱۶۴	ارسال تعداد نامعلومی آرگومان به یک تابع
۱۶۵	آرگومان‌های نام‌دار
۱۶۷	توابعی که آرایه‌ها را برمی‌گردانند

۱۶۸		سربار گذاری توابع
۱۷۰		ویزوال بیسیک ۲۰۱۰ در عمل: پروژه Overloaded Functions
۱۷۴		مستند سازی (توضیح دار کردن) توابع
۱۷۵		خلاصه و تمرین

بخش دوم: ایجاد نرم افزارهای تحت ویندوز

۱۷۹		فصل ۴: طراحی صفحه تصویر اصلی نرم افزارها و برنامه نویسی (پروژه ممر)
۱۷۹		طراحی نرم افزارهای کار کننده تحت ویندوز
۱۸۱		ایجاد یک ماشین حساب وام
۱۸۲		درک طرز کار برنامه Loan Calculator
۱۸۵		طراحی صفحه تصویر اصلی ماشین حساب
۱۸۷		تراز کردن چیدمان کنترل ها
۱۸۸		کدنویسی برنامه وام
۱۹۳		راستی آزمایی داده ها
۱۹۹		ساختن یک ماشین حساب
۱۹۹		طراحی صفحه تصویر اصلی برنامه
۲۰۲		کدنویسی برنامه Math Calculator
۲۰۵		کدنویسی عملیات ریاضی
۲۰۸		استفاده از ابزارهای اولیه دیباگ
۲۱۱		مدیریت اعتراضات (exceptions)
۲۱۵		خلاصه و تمرین

۲۱۷		فصل ۵: کنترل های اصلی ویدئو
۲۱۸		کنترل TextBox (مستطیل متن)
۲۱۹		ویژگی های اساسی
۲۱۹		TextAlign (تراز متن)
۲۱۹		MultiLine (چندخطی)
۲۱۹		MaxLength (حداکثر طول)
۲۱۹		ScrollBars (میله های پیمایش)
۲۲۰		WordWrap (پیچش متن)
۲۲۱		AcceptsReturn (قبول کلید اینتر) و AcceptsTab (قبول کلید تب)
۲۲۱		CharacterCasing (بزرگ یا کوچک بودن حروف انگلیسی)
۲۲۱		PasswordChar (کاراکتر رمز)
۲۲۱		ReadOnly (فقط خواندنی) و Locked (قفل شده)
۲۲۲		ویژگی های مربوط به کار روی متن
۲۲۲		Text (متن)
۲۲۴		Lines (خطوط)
۲۲۴		ویژگی های انتخاب متن
۲۲۴		SelectedText (متن انتخاب شده)
۲۲۴		SelectionStart (مکان شروع متن انتخاب شده) و

۲۲۴	SelectionLength (طول متن انتخاب شده)
۲۲۶	HideSelection (انتخاب مخفی شود)
۲۲۶	لغو (Undo) کردن تغییرات
۲۲۶	ویژوال بیسیک ۲۰۱۰ در عمل: پروژه TextPad
۲۲۹	دستورات منوی Edit (ویرایش)
۲۳۱	منوهای Process (پردازش) و Format (قالب‌بندی)
۲۳۱	عملیات یافتن و جایگزینی
۲۳۴	دستورات Undo/Redo
۲۳۵	گرفتن زده شدن کلیدهای کیبورد
۲۳۷	گرفتن زده شدن کلیدهای تابع
۲۳۸	ویژگی‌های Autocomplete (تکمیل خودکار)
۲۴۳	کنترل‌های ListBox، CheckedListBox و ComboBox
۲۴۴	ویژگی‌های اصلی
۲۴۴	IntegralHeight (ارتفاع کامل)
۲۴۴	Items
۲۴۴	MultiColumn (چند ستونی)
۲۴۵	SelectionMode (مود انتخاب)
۲۴۵	Sorted (مرتب شده)
۲۴۵	Text
۲۴۶	کار روی کلکسیون Items
۲۴۷	متد Add (اضافه شود)
۲۴۸	متد Insert (وارد شود)
۲۴۸	متد Clear (پاک شود)
۲۴۸	ویژگی Count (تعداد)
۲۴۸	متد CopyTo (کپی شود در)
۲۴۸	متدهای Remove (حذف شود) و RemoveAt (در ... حذف شود)
۲۴۹	متد Contains (محتوی)
۲۴۹	انتخاب کردن آیتم‌ها
۲۵۰	ویژوال بیسیک ۲۰۱۰ در عمل: پروژه ListBox Demo
۲۵۱	طراحی دکمه‌های Add Item (اضافه کردن آیتم)
۲۵۱	حذف آیتم‌ها از دو لیست
۲۵۲	انتقال آیتم‌ها از یک لیست به لیست دیگر
۲۵۳	جستجو در ListBox
۲۵۳	ویژوال بیسیک در عمل: برنامه ListBoxFind
۲۵۵	کنترل ComboBox (مستطیل لیست‌دار)
۲۵۸	اضافه کردن آیتم‌ها به ComboBox در Runtime (زمان اجرا)
۲۵۹	ویژوال بیسیک ۲۰۱۰ در عمل: پروژه FlexCombo
۲۶۰	کنترل‌های ScrollBar (میله پیمایش) و TrackBar (نوار خط‌کش)
۲۶۱	کنترل ScrollBar
۲۶۱	ویژوال بیسیک ۲۰۱۰ در عمل: پروژه Colors
۲۶۲	رویندهای کنترل ScrollBar
۲۶۲	مدیریت رویندها در برنامه Colors

۲۶۳	کنترل TrackBar
۲۶۴	ویزوال بیسیک ۲۰۱۰ در عمل: پروژه Inches
۲۶۵	خلاصه و تمرین
۳۶۷	فصل ۴: کار با فرم‌ها
۲۶۸	ظاهر فرم‌ها
۲۶۹	ویژگی‌های شی Form
۲۶۹	AcceptButton (دکمه لغو) CancelButton (دکمه قبول)
۲۶۹	AutoScaleMode (مود مقیاس اتوماتیک)
۲۶۹	AutoScroll (پیمایش اتوماتیک)
۲۷۰	AutoScrollPosition (موقعیت میله‌های پیمایش اتوماتیک)
۲۷۱	AutoScrollMargin (حاشیه پیمایش اتوماتیک)
۲۷۱	AutoScrollMinSize (حداقل اندازه برای پیمایش اتوماتیک)
۲۷۲	FormBorderStyle (شیوه کناره‌های فرم)
۲۷۲	ControlBox (نماد کنترل)
۲۷۲	MinimizeBox (دکمه حداقل) و MaximizeBox (دکمه حداکثر)
۲۷۳	MinimumSize (اندازه حداقل) MaximumSize (اندازه حداکثر)
۲۷۳	استفاده از کلاس SystemInformation برای خواندن اطلاعات سیستم
۲۷۳	KeyPreview (پیش‌نمایش کلید زده شده)
۲۷۴	SizeGripStyle (اندازه دستگیره تغییر اندازه)
۲۷۴	StartPosition (موقعیت شروع) Location (مکان)
۲۷۴	TopMost (روی همه)
۲۷۵	Size (اندازه)
۲۷۵	قرار دادن کنترل‌ها روی فرم
۲۷۷	تنظیم ویژگی TabIndex (ترتیب ضربات کلید تب)
۲۷۹	ویزوال بیسیک ۲۰۱۰ در عمل: پروژه Contacts
۲۸۱	مدیریت کلیدهای زده شده
۲۸۳	لنگر کردن (anchoring) و استقرار (docking)
۲۸۳	لنگر کردن (anchoring) کنترل‌ها به اضلاع فرم
۲۸۵	مستقر کردن (docking) کنترل‌ها در فرم
۲۸۷	تقسیم کردن فرم‌ها به چند پانل
۲۹۰	رویدادهای فرم‌ها
۲۹۰	رویدادهای Activated (فعال شده) و Deactivated (غیرفعال شده)
۲۹۰	رویدادهای FormClosing (بستن فرم) و FormClosed (فرم بسته شده)
۲۹۰	رویدادهای Resize (تغییر اندازه)، ResizeBegin (شروع تغییر اندازه) و ResizeEnd (پایان تغییر اندازه)
۲۹۱	اندازه
۲۹۲	رویداد Scroll (میله پیمایش)
۲۹۲	رویداد Paint (نقاشی)
۲۹۳	لود کردن و نمایش دادن فرم‌ها
۲۹۴	فرم آغازین (Startup Form)
۲۹۴	کنترل کردن یک فرم از درون فرم دیگر

۲۹۴	اشتراک‌گذاری متغیرها بین فرم‌ها
۲۹۶	مقایسه فرم با دیالوگ‌باکس
۳۰۰	ویژوال بیسیک در عمل: پروژه MultipleForms (چندفرمی)
۳۰۴	ایجاد فرم‌های دینامیک در Runtime
۳۰۴	کلکسیون Controls
	ویژوال بیسیک ۲۰۱۰ در عمل: پروژه ShowControls (تعیین و نمایش نوع کنترل‌های داخل یک فرم)
۳۰۵	
۳۰۸	ویژوال بیسیک در عمل: پروژه DynamicForm (فرم تغییرپذیر)
۳۱۳	ایجاد مدیران رویدادها در Runtime
۳۱۵	طراحی منوها
۳۱۵	ادیتور (ویرایشگر) منو
۳۱۹	ویژگی‌های ToolStripMenuItem (نوار ابزار آیتم منو)
۳۲۰	برنامه‌نویسی دستورات منو
۳۲۱	استفاده از کلیدهای دسترسی و میانبر
۳۲۱	کلیدهای دسترسی به فرم
۳۲۲	کلیدهای میانبر (shortcut keys)
۳۲۳	کار روی منوها در Runtime
۳۲۳	ایجاد دو نوع کوتاه و بلند برای یک منو
۳۲۴	اضافه و حذف کردن دستورات یک منو در Runtime
۳۲۶	ایجاد منوهای کلیک راست (Context Menu)
۳۲۶	خلاصه و تمرین

فصل ۷: سایر کنترل‌های ویژوال

۳۳۰	کنترل‌های دیالوگ‌باکسی مشترک در همه نرم‌افزارها
۳۳۲	کار با کنترل‌های دیالوگ‌باکسی از پیش تعریف شده
۳۳۳	کنترل ColorDialog (دیالوگ‌باکس [Color])
۳۳۴	AllowFullOpen (همه دیالوگ باز شود)
۳۳۴	AnyColor (همه رنگ‌ها)
۳۳۴	Color
۳۳۴	CustomColors (رنگ‌های دلخواه)
۳۳۵	SolidColorOnly (فقط رنگ‌های یکدست)
۳۳۵	کنترل FontDialog (دیالوگ‌باکس Font)
۳۳۶	AllowScriptChange (امکان تغییر ناحیه Script داده شود)
۳۳۷	AllowVerticalFonts (فونتهای عمودی نشان داده شوند)
۳۳۷	Color, ShowColor
۳۳۷	FixedPitchOnly (فقط فونت‌های با عرض ثابت)
۳۳۷	Font
۳۳۷	FontMustExist (فونت باید وجود داشته باشد)
۳۳۸	MinSize و MaxSize (حداکثر و حداقل اندازه فونت‌ها)
۳۳۸	ShowApply (دکمه Apply نشان داده شود)
۳۳۸	ShowEffects (جلوه‌ها نمایش داده شوند)

۲۳۹	کنترل‌های OpenFileDialog و SaveDialog (دیالوگ‌باکس‌های Open و Save)
۲۴۰	AddExtension (پسوند به طور اتوماتیک اضافه شود)
۲۴۰	CheckFileExists (وجود فایل بررسی شود)
۲۴۰	CheckPathExists (وجود مسیر بررسی شود)
۲۴۰	DefaultExt (پسوند پیش فرض)
۲۴۰	DereferenceLinks (ارتباط با مرجع لازم نیست)
۲۴۱	FileName
۲۴۱	Filter (صافی)
۲۴۲	FilterIndex
۲۴۲	InitialDirectory (پوشه اولیه)
۲۴۲	RestoreDirectory (بازیابی پوشه)
۲۴۲	FileNames
۲۴۳	MultiSelect (چند انتخاب)
۲۴۳	ReadOnlyChecked (گزینه Read Only تیک دارد) و ShowReadOnly
۲۴۳	متدهای OpenFile و SaveFile
۲۴۴	ویژوال بیسیک در عمل: انتخاب چند فایل
۲۴۵	کنترل FolderBrowserDialog (دیالوگ‌باکس مرورگر پوشه)
۲۴۵	RootFolder (پوشه ریشه)
۲۴۶	SelectedPath (مسیر انتخاب شده)
۲۴۶	ShowNewFolderButton (دکمه New نمایش داده شود)
۲۴۶	ویژوال بیسیک در عمل: پروژه Folder Browsing Demo
۲۴۹	کنترل RichTextBox
۲۵۰	زبان RTF (متن فرمت شده)
۲۵۱	ویژگی‌های مربوط به کار روی متن و قالب‌بندی آن
۲۵۱	SelectedText (متن انتخاب شده)
۲۵۲	ویژگی‌های SelectionStart (مکان شروع انتخاب) و SelectionLength (طول انتخاب)
۲۵۲	ویژگی SelectionAlignment (تراز متن)
	ویژگی‌های SelectionIndent ، SelectionRightIndent و SelectionHangingIndent
۲۵۲	SelectionBullet (بولت متن انتخاب شده) و BulletIndent (تورفتگی بولت‌دار)
۲۵۳	SelectionTabs (تب‌های متن انتخاب شده)
۲۵۴	متمدهای کار روی کنترل RichTextBox
۲۵۴	متد SaveFile
۲۵۵	متد LoadFile
۲۵۵	Select (انتخاب شود) و SelectAll (همه انتخاب شوند)
۲۵۶	ویژگی‌های ویرایش پیشرفته
۲۵۶	CanRedo و CanUndo
۲۵۶	RedoActionName و UndoActionName
۲۵۷	Redo و Undo
۲۵۷	کپی، برش (Cut) و چسباندن (Paste)
۲۵۸	ویژوال بیسیک ۲۰۱۰ در عمل: پروژه RTFPad
۲۶۰	منوی File برنامه RTFPad

۳۶۲		منوی Edit برنامه RTFPad
۳۶۴		دیالوگ باکس Search & Replace (یافتن و تعویض)
۳۶۷		کنترل های TreeView و ListView
۳۶۸		ساختارهای درختی و لیستی
۳۷۱		کنترل TreeView
۳۷۲		اضافه کردن گره ها در زمان طراحی
۳۷۴		اضافه کردن گره ها در Runtime
۳۷۴		اضافه کردن گره های جدید
۳۷۵		کنترل ListView
۳۷۸		کلکسیون Columns (ستون ها)
۳۷۹		وارد کردن آیتم های اصلی و فرعی به کنترل ListView
۳۸۰		کلکسیون Items
۳۸۱		کلکسیون SubItems (آیتم های فرعی)
۳۸۲		کار روی آیتم های انتخاب شده
۳۸۳		ویژوال بیسیک ۲۰۱۰ در عمل: پروژه Custom Explorer
۳۸۴		خلاصه و تمرین

بخش سوم: کار با کلاس ها و کنترل های خودتان

۳۸۹		فصل ۸: کار با اشیاء
۳۹۰		کلاس ها و اشیاء
۳۹۱		کلاس چیست؟
۳۹۳		کلاس ها کد را با داده ها ترکیب می کنند
۳۹۴		ایجاد کلاس Minimal (کمترین)
۳۹۸		اضافه کردن کد به کلاس Minimal
۴۰۰		استفاده از پراپرتی های Property
۴۰۲		تولید اعتراض ها (exceptions)
۴۰۴		پیاده سازی ویژگی های فقط خواندنی (read-only)
۴۰۵		محاسبه مقدار ویژگی ها در لحظه درخواست
۴۰۶		ویژگی های خود پیاده ساز (Auto-Implemented Properties)
۴۰۷		اختصاصی سازی اعضای پیش فرض
۴۰۷		اختصاصی سازی متد ToString (تبدیل به رشته)
۴۰۸		اختصاصی سازی متد Equals (مساوی است با ...)
۴۰۹		بدانید چه چیزهایی را دارید مقایسه می کنید
۴۱۰		عملگر IS
۴۱۰		انواع شمارشی اختصاصی ((custom enumerations
۴۱۲		ویژوال بیسیک ۲۰۱۰ در عمل: پروژه Contacts (تماس ها)
۴۱۷		اضافه کردن، امکان ویرایش و حذف تماس ها
۴۲۰		سازندگان اشیاء
۴۲۲		استفاده از کلاس SimpleClass در سایر پروژه ها
۴۲۴		راه انداختن رویدادها

۴۳۸	متدهای همراه نسخه (instance methods) و متدهای اشتراکی (shared methods).....
۴۳۲	یک کلاس "واقعی".....
۴۳۴	تودرتو کردن کلاس‌ها.....
۴۳۸	سربرار گذاری عملگرها (operator overloading).....
۴۴۲	پیاده‌سازی عملگرهای یکتایی (unary operators).....
۴۴۲	مدیریت واریانت‌ها (variants).....
۴۴۴	خلاصه و تمرین.....

فصل ۱۹ ایجاد کنترل‌های جدید برای برنامه‌های تحت ویندوز..... ۴۴۵

۴۴۶	طراحی کنترل برای برنامه‌های تحت ویندوز.....
۴۴۷	توسعه کنترل‌های موجود.....
۴۴۸	ایجاد کنترل FocusedTextBox.....
۴۵۱	اضافه کردن عملگرها به کنترل اختصاصی شما.....
۴۵۴	تست کردن کنترل FocusedTextBox.....
۴۵۵	طبقه‌بندی ویژگی‌های یک کنترل.....
۴۵۷	ایجاد کنترل‌های مرکب.....
۴۵۸	ویژوال بیسیک ۲۰۱۰ در عمل: کنترل ColorEdit.....
۴۶۱	تست کردن کنترل مرکب ColorEdit.....
۴۶۱	ایجاد کنترل‌های رسم شده توسط کارم.....
۴۶۲	ویژوال بیسیک در عمل: کنترل Label3D.....
۴۶۳	مشخصات کنترل Label3D.....
۴۶۳	طراحی کنترل جدید Label3D.....
۴۶۷	تست کردن کنترل جدید شما.....
۴۶۹	رویدادهای تغییر یافته.....
۴۷۱	تعریف رویدادهای اختصاصی.....
۴۷۲	استفاده از کنترل اختصاصی Label3D در سایر پروژه‌ها.....
۴۷۳	طراحی کنترل‌هایی با اشکال غیرمستطیلی.....
۴۷۷	اختصاصی کردن کنترل‌های لیست‌دار.....
۴۷۸	طراحی کنترل‌های ListBox ترسیم دلخواه.....
۴۸۱	خلاصه و تمرین.....

فصل ۲۰ برنامه‌نویسی شی‌گرای کاربردی..... ۴۸۳

۴۸۴	نکاتی درباره برنامه‌نویسی شی‌گرا.....
۴۸۴	مقایسه کلاس با شی.....
۴۸۵	مقایسه شی با متغیر شی.....
۴۸۷	متغیرهایی که مقدار اولیه نمی‌خواهند و متغیرهای هیچ‌پذیر.....
۴۸۹	بررسی نوع مقادیر (value types).....
۴۸۹	بررسی انواع رجوعی.....
۴۹۲	مقایسه ویژگی با فیلد.....
۴۹۳	مقایسه اعضای اشتراکی (shared members) با.....
۴۹۳	اعضای همراه نسخه (instance members).....

۴۹۵	 قالب‌ریزی نوع (Type Casting)
۴۹۷	 مقایسه مقید کردن (binding) زود هنگام و دیر هنگام متغیرها
۴۹۸	 پیدا کردن نوع یک متغیر
۴۹۹	 ارث‌بری
۵۰۰	 نحوه اجرای ارث‌بری
۵۰۱	 طراحی با استفاده از ارث‌بری
۵۰۷	 متدهای اضافه شونده (extension methods)
۵۱۲	 چندشکل‌پذیری (Polymorfism)
۵۱۴	 ایجاد کلاس Shape
۵۱۷	 تست کلاس Shape
۵۱۹	 چه کسی می‌تواند چه چیزی را به ارث ببرد
۵۲۰	 کلمات کلیدی مخصوص کلاس‌های پدر
۵۲۰	 کلمات کلیدی مختص کلاس‌های فرزند
۵۲۱	 کلمات کلیدی مختص اعضای کلاس‌های پدر
۵۲۲	 کلمات کلیدی مختص اعضای کلاس‌های مشتق شده (فرزند)
۵۲۲	 ویرژوال بیسیک ۲۰۱۰ در عمل: پروژه InheritanceKeywords
۵۲۴	 کلمات کلید MyBase و MyClass
۵۲۵	 به کارگیری وراثت
۵۳۲	 محیط Class Diagram Designer (طراح دیاگرام کلاس‌ها)
۵۳۳	 خلاصه و تمرین

www.ketab.ir

مقدمه

به مایکروسافت ویژوال بیسیک ۲۰۱۰، نسخه دیگری از محبوب‌ترین زبان دنیا برای ایجاد نرم‌افزارهای تحت ویندوز و وب^۱ خوش آمدید البته در نرم‌افزارسازی مدرن زبان فقط یکی از اجزائی است که ما برای نرم‌افزارسازی مورد استفاده قرار می‌دهیم. مهم‌ترین جزء نرم‌افزارسازی بستر دات‌نت^۲ است که جزء لاینفک هر نرم‌افزاری است. در واقع، بستر دات‌نت از خود زبان مهم‌تر است. شما می‌توانید بستر دات‌نت را همچون کلکسیون بزرگی از توابع درباره هر کار برنامه‌نویسی در نظر بگیرید. با وجود بستر دات‌نت دیگر زبان برنامه‌نویسی شما بزرگ نمی‌شود و برای هر کاری مجموعه دستورات مربوط به آن کار را در برنامه خود قرار می‌دهید و از مجموعه دستورات دیگری که به دردتان نمی‌خورد استفاده نمی‌کنید.

برای مثال، همه متدهای مربوط به ترسیم در کلاسی به نام `System.Drawing` (ترسیم سیستم) قرار دارند. بنابراین برای رسم یک مستطیل، شما متد `DrawRectangle` (رسم مستطیل) از کلاس^۳ `System.Drawing` را از بستر دات‌نت فراخوانی کرده و آرگومان‌های آن (مقادیری که برای رسم مستطیل نیاز دارد) را به آن ارسال می‌کنید و این متد مستطیل شما را رسم می‌کند؛ یا برای ایجاد یک پوشه جدید شما متد `CreateDirectory` (ساختن دایرکتوری^۴) از کلاس `Directory` را در برنامه خود فراخوانی می‌کنید و یا برای کار بر روی فایل‌های درون یک پوشه متد `GetFiles` (گرفتن فایل‌ها) از همین کلاس را فراخوانی می‌کنید.

"بستر دات‌نت" حاوی کل دستورات لازم برای کار با "سیستم عامل" است و آنها را از طریق "متدها"^۵ در اختیار

^۱ Web یا همان World Wide Web یا WWW و W3 مجموعه‌ای از میلیاردها صفحه اطلاعات است که از طریق اینترنت قابل دسترسی هستند. این صفحات از طریق "لینک" قابل دسترسی هستند.

^۲ NET Framework

^۳ Class یک کلاس یک "بسته" است که هم محتوی داده‌ها و هم محتوی برنامه‌های کار روی این داده‌ها (متدها) است. مثلاً کلاس `System.Drawing` (ترسیم سیستم) هم دارای یک برنامه اصلی برای رسم شکل‌های مختلف است و هم چندین متد مانند `DrawRectangle` دارد که انواع شکل‌های متداول را جداگانه رسم می‌کنند. هر کدام از این شکل‌ها رنگ و ضخامت خط و انواع ویژگی‌ها را دارند که اینها نیز در داخل همین کلاس قابل تعیین هستند. شما یک کلاس را در برنامه خود فراخوانی کرده و با دادن مقدار به متدها و ویژگی‌های داخل این کلاس کار مورد نظر خود را انجام می‌دهید.

^۴ در اینجا دایرکتوری همان پوشه است.

^۵ Methods متد یک قطعه برنامه مستقل است که یک یا چند مقدار را به عنوان آرگومان می‌پذیرد و بر طبق آنها کارهای خود را انجام می‌دهد و سپس یک یا چند مقدار را به برنامه‌ای که آن را احضار کرده است برمی‌گرداند.

برنامه‌هایی که شما می‌نویسید قرار می‌دهد. متدها بسیار شبیه به توابع^۱ هستند. بستر داتانت کلکسیون بزرگی از چنین متدهایی است که بر اساس نقش خود در واحدهایی^۲ به نام "کلاس" گروه‌بندی شده‌اند تا به این ترتیب پیدا کردن متد مربوط به کاری که می‌خواهید انجام دهید بسیار راحت شود. "زبان برنامه‌نویسی" و "بستر داتانت" دو جزء کاملاً ضروری "برنامه‌نویسی" برای ایجاد "نرم‌افزارهای تحت ویندوز" هستند. برنامه‌نویسی با هر یک از این‌ها به تنهایی هم امکان‌پذیر است اما فرآیند برنامه‌نویسی به شدت کند خواهد بود.

برای ایجاد یک نرم‌افزار به ابزارهای متعددی نیاز است که باعث می‌شوند کدنویسی^۳ ضروری شود. عنصر سوم طراحی یک نرم‌افزار وجود یک "محیط یکپارچه" (IE) است که خانه همه این ابزارها باشد و به شما امکان دهد بدون خارج شدن از یک محیط و رفتن به یک محیط دیگر همه کارهای مربوط به طراحی یک نرم‌افزار را از طریق ماوس و کلیک در "یک پنجره" انجام دهید. در واقع IE محیطی است که شما می‌توانید در آن "فرم‌های" خود را با ابزارهای ویزوال^۴ طراحی کنید و کد^۵ (تکه برنامه‌های) مورد نظران را نیز بنویسید.

این محیط که توسط ویزوال استودیو فراهم می‌شود "محیط یکپارچه برنامه‌نویسی" یا IDE نامیده می‌شود. شما از امکانات فراهم شده توسط ابزارهای ویزوال حیرت خواهید کرد: شما می‌توانید یک نرم‌افزار "داده محور" را بدون نوشتن حتی یک خط کد طراحی کنید و باز هم می‌توانید بدون نوشتن حتی یک خط کد، با استفاده از همان ابزارهای برنامه قبلی، در همین محیط (یا در واقع "پنجره برنامه ویزوال استودیو") یک "صفحه وب داده محور" زیبا طراحی کنید.

ویزوال استودیو حتی ابزارهایی را برای طراحی "بانک‌های اطلاعاتی"^۶ به شما می‌دهد. البته شما درک می‌کنید

^۱ Functions شما می‌توانید توابع را مانند فرمول‌ها در نظر بگیرید که به آن‌ها یک یا چند مقدار می‌دهید و تابع این مقادیر را در فرمول خود گذاشته و یک یا چند مقدار دیگر به شما می‌دهد. توجه کنید که فرق تابع با متد این است که متد یک برنامه مستقل است که ممکن است از یک یا چند تابع استفاده کند.

units^۲

^۲ جلوتر که برویم متوجه می‌شوید که برای طراحی یک نرم‌افزار هم نوشتن برنامه مهم است (که به آن کدنویسی می‌گویند) و هم استفاده از ابزارهای زبان برنامه نویسی. مثلاً لازم نیست که شما برای نرم‌افزارتان منو طراحی کنید. ابزار ساختن منو در زبان‌های ویزوال وجود دارد و شما فقط باید آن را احضار کنید. به این ترتیب به راحتی یک منو در نرم‌افزارتان می‌گذارید و داخل آن نام دستورآتان را تایپ می‌کنید. اما اینکه هر دستور چه کاری باید انجام دهد را باید کدنویسی کنید.

^۳ Integrated Environment در این محیط هم امکان طراحی برنامه وجود دارد و هم امکان اشکال‌زدایی آن و هم امکان کامپایل کردن آن.

^۴ Forms منظور از "فرم" همان پنجره اصلی و هر پنجره مستقل یک نرم‌افزار است. پس پنجره هر دستور یا پنجره هر دکمه تب هم یک فرم است.

^۵ "ویزوال" به معنی "چشمی" و "بصری" است؛ یعنی نیازی به برنامه نویسی نیست. مثلاً اگر می‌خواهید یک دکمه در جایی از فرم خود قرار دهید کافیست روی نماد دکمه در یک نوار ابزار کلیک کنید تا این نماد انتخاب شود و سپس در جایی از فرم خود کلیک کنید تا دکمه در آنجا ظاهر شود و بعد با ماوس آن را جابجا کنید و در محل دقیق آن قرار دهید و یا گوشه آن را با ماوس بگیرید و کوچک و بزرگ کنید. خلاصه همه کارها با چشم و ماوس انجام می‌شود.

^۶ Code در این کتاب منظور از کد، بیشتر قطعات کوچک برنامه است که باید تایپ شوند.

^۷ Integrated Development Environment

^۸ Data-driven "داده‌گرا" یعنی نرم‌افزاری که از شما داده می‌گیرد و بر اساس آنچه گرفته است کار می‌کند و یک عملکرد ثابت ندارد. نمونه جالب این نوع نرم‌افزارها بازی‌های کامپیوتری هستند.

^۹ database بانک‌های اطلاعاتی یا "پایگاه‌های داده‌ها" به برنامه‌هایی گفته می‌شود که اطلاعات را ذخیره کرده و روی آنها کار انجام می‌دهند. مثلاً یک نرم‌افزار فروشگاه یک بانک اطلاعاتی دارد که موجودی انبار در آن است و هرچه که فروخته می‌شود یا خریده می‌شود در آن ثبت می‌شود. در اکثر مواقع یک بانک اطلاعاتی یک جدول است که هر ردیف آن یک "رکورد" و هر ستون آن یک "فیلد" نامیده می‌شود.

که "ویژوال استودیو"^۱ یک وسیله برنامه‌سازی بدون کدنویسی نیست؛ بلکه فقط انجام کارهای خاصی را از طریق "ویزاردها"^۲ ساده می‌کند و معمولاً ما در طراحی دخالت می‌کنیم و برای قسمت‌های مختلف و ویژه برنامه‌مان کدهای خاصی را می‌نویسیم. اما ویژوال استودیو ابزارهای متعددی برای طراحی نرم‌افزار در اختیار شما می‌گذارد؛ از ابزارهای اشکال‌زدایی^۳ که همه انواع اشکالات کدهایی که شما نوشته‌اید را ردگیری و رفع می‌کنند تا ابزارهای طراحی بانک‌های اطلاعاتی و ویزاردهای انتشار^۴ که فرایند انتشار نرم‌افزار را فراهم می‌کنند.

این کتاب چگونگی استفاده از ویژوال استودیوی ۲۰۱۰ و ویژوال بیسیک ۲۰۱۰ را برای طراحی نرم‌افزارهای حرفه‌ای کارکننده تحت ویندوز و وب به شما نشان می‌دهد. ما در ابتدا با ابزارهای ویژوال موجود در ویژوال استودیو شروع خواهیم کرد و سپس به ویژوال بیسیک و بستر دات‌نت خواهیم پرداخت. یک برنامه تحت ویندوز عبارت است از "یک صفحه تصویر اصلی و تعدادی کد که در پشت عناصر این صفحه تصویر هستند" (کدها به نرم‌افزار می‌گویند که هر زمان روی یکی از اجزای صفحه تصویر-از قبیل: یک دکمه، یک آیتم، یک منو و غیره- کلیک شد نرم‌افزار چه کار باید بکند).

شما از ابزارهای^۵ ویژوال استودیو برای ایجاد اجزای "صفحه تصویر اصلی نرم‌افزار"^۶ که گرافیک است استفاده کرده و سپس اینکه هر عنصر از این صفحه تصویر چه کار باید بکند را در ویژوال بیسیک برنامه‌نویسی (کدنویسی) خواهید کرد. برای هر پردازش مهم-از قبیل: کار روی فایل‌ها و پوشه‌ها، ذخیره کردن داده‌ها و ... - شما از "کلاس‌های"^۷ مناسب موجود در "بستر دات‌نت" استفاده خواهید کرد.

بخش عمده این کتاب به مفیدترین اجزای بستر دات‌نت می‌پردازد. همچنین، ما بانک‌های اطلاعاتی و "برنامه‌های داده محور" را نیز بررسی خواهیم کرد که رایج‌ترین نوع نرم‌افزارهای تجاری هستند (نظیر برنامه‌های حسابداری و انبارداری که از شما داده‌ها را می‌گیرند و روی آنها کار انجام می‌دهند و نهایتاً نتیجه کار خود را به شما نشان می‌دهند). در نهایت، ما مبانی طراحی نرم‌افزار برای اینترنت را مورد بحث قرار خواهیم داد و شما نحوه طراحی سرویس^۸ برای نرم‌افزارهای تحت وب (اینترنت) را فرا خواهید گرفت.

^۱ ویژوال استودیو یعنی "استودیوی چشمی". این یعنی این که شما هر چه را که می‌خواهید می‌توانید با چشم و ماوس و کلیک کردن و کشیدن با ماوس انجام دهید. اما این فقط نام این برنامه است و فقط بخشی از کارهای آن را بیان می‌کند.

^۲ wizard ویزارد یعنی "جادوگر". اولین بار با ظهور ویندوز این اصطلاح باب شد و منظور از آن پنجره‌های پشت سر هم بودند که باز می‌شدند و هر دفعه چیزی را از شما می‌خواستند تا یک کار را انجام دهند. مثلاً ویزارد نصب یک برنامه یا ویزارد وصل شدن به اینترنت و ... و.

^۳ Debugging کلمه باگ به معنی سوسک است. وقتی کامپیوترها اندازه یک اتاق بودند یک بار یکی از آنها از کار افتاد و پس از باز کردن آن معلوم شد که یک سوسک در آن مرده. از آن به بعد به اشکال کامپیوتری "باگ" و به رفع کردن آن "دیب‌باگ" گفته شد.

^۴ Publishing آخرین مرحله طراحی یک نرم‌افزار "انتشار" است که به معنی آماده‌سازی نرم‌افزار برای روانه شدن به بازار است.

^۵ مثلاً یک ابزار منو و گزینه‌های داخل آن را می‌سازد، یک ابزار دیالوگ‌باکس (پنجره دستورات) را می‌سازد، یک ابزار لیست گزینه‌ها را می‌سازد و ...

^۶ User Interface یا UI به معنی: "رابط کاربر" است. یعنی جایی که برنامه با کاربر ارتباط برقرار می‌کند. این همان

صفحه تصویر اصلی برنامه است.

^۷ Classes

^۸ Web Service در سال‌های اخیر استفاده از سرویس‌های وب در برنامه‌نویسی صفحات وب و نرم‌افزارهای کارکننده در اینترنت بسیار متداول شده است. دلیل بسیار مهم این امر آن است که این سرویس‌ها به زبان برنامه‌نویسی و به سیستم عامل کاری ندارند. این سرویس‌ها معمولاً با XML نوشته می‌شوند و به همین دلیل مستقل هستند و در همه جا کار

ویژگی‌های این کتاب

این کتاب از سری کتاب‌های Mastering (کسب استادی) انتشارات Sybex است. این کتاب برای خوانندگانی که در کار با کامپیوتر متوسط و پیشرفته هستند بسیار مناسب است و برای کسانی که تصمیم دارند به کارشناسان حرفه‌ای برنامه‌نویسی تبدیل شوند امکانات مناسب را فراهم می‌کند. هر کتابی که از سری Mastering است شامل موارد زیر است:

- سناریوهای دنیای واقعی: موردکاوی‌ها و مثال‌هایی که نحوه کاربرد ابزارها، تکنیک‌ها و دانش‌های ارائه شده را در عمل نشان می‌دهد.
- آموزش مبتنی بر مهارت: فصل‌های آن حول کارهای واقعی سازماندهی شده‌اند؛ نه مفاهیم و موضوعات انتزاعی.
- سوالات خودآزمایی: تا شما بتوانید از یادگیری خود اطمینان حاصل کنید.

این کتاب برای چه کسانی است؟

شما برای مطالعه این کتاب نیازی به داشتن دانش سطح بالا از ویرژوال بیسیک ندارید. شما باید معنای "متغیر" و "تابع" و طرز کار دستورات ساده‌ای نظیر If... Then (اگر ... آنگاه) را بدانید. این کتاب برای برنامه‌نویسان معمولی است که می‌خواهند حداکثر استفاده را از ویرژوال بیسیک ببرند.

موضوعات مورد بحث در این کتاب مورد نیاز اکثر برنامه‌نویسان ویرژوال بیسیک است و آموزش‌ها به طور عمیق صورت می‌گیرد. "ویرژوال بیسیک ۲۰۱۰" و "بستر دات‌نت" نسخه ۴/۰ دو ابزار برنامه‌نویسی بسیار غنی هستند و من مجبور بودم از بین دو گزینه: پوشش دادن سطحی تعداد زیادی از موضوعات و پوشش دادن عمیق موضوعاتی کمتر، دست به انتخاب بزنم.

برای این که موضوعات بیشتری را ارائه کنم، بسیاری از مطالب مرجع و کدهای طولانی را از قلم انداختم. برای مثال، شما کد کامل پروژه‌ها و توضیحات فرم‌ها را در متن کتاب نخواهید یافت. فرض من این است که شما خودتان خواهید توانست چند "کنترل" را روی فرم‌ها قرار دهید و "ویژگی‌های" آن‌ها را تعیین کنید و نیازی به توضیحات طولانی درباره نحوه تعیین "ویژگی‌های" هر کنترل ندارید (نحوه طراحی یک "فرم" را در فصل‌های اول و دوم خواهید آموخت). همچنین، من فرض می‌کنم که شما علاقه‌ای به خواندن بخش‌های ساده و نه چندان مهم هر برنامه ندارید. در عوض، لیست‌های کد ارائه شده در این کتاب بر روی بخش‌های مهم و اساسی هر کد تمرکز دارند: یعنی "پراسیجرهایی" که موضوع مربوطه را توصیف می‌کنند.

در این کتاب اصول و تکنیک‌های مورد نیاز برای برنامه‌سازی با ویرژوال بیسیک ارائه می‌شود. کتاب ما درباره

می‌کنند. شما از طریق سرویس‌های وب می‌توانید با برنامه‌ها و سایت‌های کارکننده در اینترنت ارتباط برقرار کرده و امکانات بسیاری را به سایت‌های خود اضافه کنید.

۱ Control به هر جزء صفحه تصویر (مانند: دکمه، منو، گزینه و ...) یک کنترل می‌گویند.

۲ Properties

۳ Procedures یک پراسیجر یا "رویه" یک تکه برنامه است که مستقل است و یک یا چند آرگومان می‌گیرد و یک یا چند آرگومان به برنامه‌ای که آن را احضار یا "قراخوانی" کرده است برمی‌گرداند.

"کلمات کلیدی" و "توابع" جدید نگارش ۲۰۱۰ نیست. موضوعاتی انتخاب شده‌اند که به عقیده من هر برنامه‌نویسی باید در یک زبان برنامه‌نویسی یاد بگیرد تا بر زبان مسلط شود. مثال‌های ارائه شده مفید و کاربردی هستند. البته همه موضوعات به یک اندازه مهم یا جالب نیستند. من امیدوارم که همگان در این کتاب برای انجام کارهای روزمره برنامه‌نویسی خود مطالب جالب و ارزشمندی پیدا کنند؛ خواه آنچه پیدا می‌کنند برنامه‌های باشند که پوشه‌ها و فایل‌های یک درایو را در یک کنترل TreeView (نمایش درختی) قرار داده و نمایش می‌دهد یا برنامه‌ای باشد که داده‌ها را به صورت جدولی چاپ می‌کند یا یک برنامه داده محور باشد که برای کار بر روی اطلاعات مشتریان یا محصولات کاربرد دارد و یا برنامه‌ای که مجموعه‌ای از اشیاء را در یک فایل ذخیره می‌کند.

بسیاری از کتاب‌ها برای انجام یک کار تعداد زیادی مرحله ارائه می‌کنند. پیروی از دستورالعمل‌ها کارهای خاص را ساده می‌کند اما برنامه‌نویسی پیروی از دستورالعمل‌ها نیست؛ برنامه‌نویسی "خلاق" بودن است و درباره درک این مطلب است که چگونه باید از تکنیک‌های یکسان در موقعیت‌های کاربردی مختلف استفاده کرد. راه استفاده خلاق از قدرت زبان‌هایی مانند ویژوال بیسیک ۲۰۱۰، درک اصول و "مدل برنامه‌نویسی" آن است.

در موارد بسیاری، من برای انجام کارهایی نظیر طراحی یک منو، یک فرآیند مفصل و گام به گام ارائه خواهم کرد. اما همه کارها به سادگی طراحی منوها نیستند. من توضیح می‌دهم که چرا هر کاری باید با یک روش مشخص انجام شود. روش‌های جایگزین انجام آن کار را هم ارائه می‌کنم و سعی می‌کنم موضوعات جدید را با موضوعات فصل‌های قبلی مرتبط کنم. ربط دادن دانش جدید با مطالبی که شما قبلاً بر آن‌ها مسلط شده‌اید بازخورد مثبت و درک عمیق‌تری از زبان را فراهم می‌کند.

این کتاب درباره داغ‌ترین ویژگی‌های زبان ویژوال بیسیک هم نیست؛ بلکه درباره تکنیک‌های مستحکم برنامه‌نویسی و مثال‌های کاربردی است. بعد از تسلط بر مبانی کدنویسی که برای طراحی نرم‌افزارهای تحت ویندوز به کار می‌رود و احساس راحتی با مثال‌های پیشرفته‌تر کتاب، کار با موضوعات این کتاب برای شما آسان خواهد شد.

^۱ Keywords به کلماتی مانند: نام دستورات و توابع و متغیرهای از پیش تعریف شده و ... و به طور کلی کلماتی که یک زبان از آن استفاده تخصصی می‌کند "کلمات کلیدی" یا "کلیدواژه" می‌گویند. از اصلاح کلید واژه بیشتر در هنگام جستجو در اینترنت استفاده می‌شود.

موضوعات پیشرفته‌تر

برخی از موضوعات بحث شده در این کتاب ساده نیستند و چند موضوع هم هستند که پیشرفته محسوب می‌شوند. ایجاد "کلکسیون‌هایی" از اشیای اختصاصی^۱ و "پرس و جو"^۲ در میان آن‌ها و قرار دادن برخی امکانات در نرم‌افزارها به شکل "سرویس‌های وب" موضوعات ساده‌ای نیستند. اما این‌ها ابزارهایی هستند که امکان حداکثر بهره‌برداری از ویرژوال استودیو را فراهم می‌کنند.

همچنین ممکن است شما برخی مثال‌ها را دشوارتر از حد انتظار خود بیابید. من سعی کرده‌ام که متن و مثال‌ها خوانا و آسان فهم باشند؛ اما خیلی هم ساده نباشند! به عنوان مثال، درک توابع اصلی کار بر روی فایل‌ها و پوشه‌ها دشوار نیست. با این وجود، برای کسب حداکثر استفاده از این توابع، شما باید نحوه بررسی فایل‌های یک پوشه (از جمله فایل‌های درون پوشه‌های فرعی آن) با تکنیکی موسوم به "بازگشت"^۳ را درک کنید. برای کسب حداکثر استفاده ممکن از هر فصل، من مثال‌های دشوارتری را هم ارائه کرده‌ام که درک بهتری از موضوعات را فراهم می‌کنند. به علاوه، شما می‌توانید بسیاری از این مثال‌ها را در برنامه‌های خود بگنجانید!

به عنوان مثال، شما می‌توانید با اندکی برنامه‌نویسی، کارهای زیادی با کنترل TreeView (مشاهده ساختار درختی) انجام دهید. اما برای حداکثر استفاده از این کنترل، باید آماده برنامه‌نویسی پیشرفته باشید که خیلی پیچیده نیست، اما زیاد هم ساده نیست. برنامه‌نویسی اکثر عملیات‌ها برای کنترل TreeView چندان پیچیده نیست اما اگر برنامه شما مستلزم پر کردن یک کنترل TreeView با تعداد نامعلومی شاخه باشد^۴ (مثلاً جاسازی یک ساختار پوشه‌ای سلسله مراتبی در یک کنترل TreeView)، کد می‌تواند پیچیده شود. همین امر در مورد چاپ هم صدق می‌کند؛ نوشتن برنامه‌ای که یک متن را چاپ کند نسبتاً سراسر است. اما چاپ گزارش‌های دارای جدول نیاز به تلاش کدنویسی قابل توجهی دارد.

من به این دلیل مثال‌های پیشرفته‌تر را در نظر گرفته‌ام که فصل‌ها بدون آن‌ها ناقص بودند. اگر مطالبی را می‌یابید که برای مطالعه شما در بار اول سنگین هستند، می‌توانید از آن‌ها صرف‌نظر کنید و بعد از این که بر سایر جوانب زبان مسلط شدید، مجدداً به سراغ آن‌ها بروید. اکثر تکنیک‌ها در حد یک برنامه‌نویس متوسط هستند. چند موضوع پیشرفته هم برای خوانندگان بلندپروازی در نظر گرفته شده است که مایلند آن گام تکمیلی را بردارند و با استفاده از جدیدترین ابزارها و تکنیک‌ها برنامه‌های پیشرفته بسازند.

یک دلیل خوب دیگر هم برای ارائه موضوعات پیشرفته‌تر وجود دارد. توضیح دادن یک موضوع ساده-مثل نحوه پر کردن یک "کلکسیون" با آیتم‌ها^۵- بسیار ساده است. اما پر کردن یک کلکسیون چه فایده‌ای دارد اگر شما نحوه ذخیره کردن آن در دیسک و بازخوانی آیتم‌های آن در دفعات بعدی را بلد نباشید؟ به طور مشابه، یادگیری

^۱ Collection به هر مجموعه از اشیای یک کلکسیون گفته می‌شود.

^۲ Query کوئری به سوالات سازمان یافته گفته می‌شود. مثلاً فرض کنید یک بانک اطلاعاتی دارید که اطلاعات مشتریان را ذخیره کرده است. به مجموعه سوالاتی نظیر: "مشتریانی که سنشان از ۳۵ سال کمتر است و خریدشان بیشتر از ۵۰۰۰۰ تومان است و در یک ماه گذشته خرید کرده‌اند اما لباس نخریده‌اند کدامند؟" یک کوئری گفته می‌شود.

^۳ Recursion بازگشت در واقع یعنی "تکرار خود از طریق بازگشت به خود". مانند دو آینه مقابل هم که یک تصویر را بارها و به اعداد بی‌نهایت تکرار می‌کنند. در برنامه نویسی به توابعی بازگشتی گفته میشود که در تعریف تابع به خود آن رجوع شده باشد. در واقع در کامپیوتر بهتر است این اصطلاح "خودارجاعی" ترجمه شود. یک مثال خوب از تابع خودارجاعی تابع محاسبه فاکتوریل است (... ۱*۲*۳*۴*۵).

^۴ یعنی قرار دادن تعداد زیادی پوشه فرعی داخل یک پوشه.

^۵ Items یک آیتم یک "چیز" است که هر چیزی ممکن است باشد.

نحوه چاپ فایل‌هایی که فقط محتوی فقط متن باشند چه فایده‌ای دارد؟ در محیط‌های تجاری، به احتمال قوی از شما خواسته می‌شود که یک گزارش محتوی انواع جداول را چاپ کنید که کار بسیار دشوارتری نسبت به چاپ کردن متن است.

در نوشتن این کتاب یکی از اهداف من، توضیح کامل همه موضوعات مربوط به یک کار عملی و ارائه کل اطلاعات مورد نیاز شما برای انجام آن کار است. مثلاً نه تنها نحوه ایجاد کلکسیون‌ها را گفته‌ام، بلکه نحوه ذخیره کردن آن‌ها در فایل‌ها را هم بیان کرده‌ام؛ یا نه تنها نحوه نوشتن در یک فایل را گفته‌ام، بلکه نحوه احضار یک فایل را هم ارائه کرده‌ام (این کار را از طریق همان دیالوگ‌باکسی که همه نرم‌افزارهای تحت ویندوز مورد استفاده قرار می‌دهند انجام داده‌ام)؛ یا مثلاً نه تنها نحوه چاپ یک چیز، بلکه نحوه ایجاد پیش‌نمایش از خروجی چاپ را هم ذکر کرده‌ام.

خلاصه من سعی کرده‌ام که همه مطالب عملی مورد نیاز برای طراحی یک نرم‌افزار را در اختیار شما قرار دهم.

ساختار کتاب

این کتاب برای خواندن روزنامه‌وار از صفحه اول تا صفحه آخر نیست (من می‌دانم بسیاری از افراد کتاب‌های کامپیوتر را اینگونه نمی‌خوانند) و هرچند همه فصل‌ها حاوی ارجاعاتی به سایر فصل‌ها هستند، اما هر فصل مستقل از سایر فصل‌ها است؛ هر موضوع به طور عمیق بررسی می‌شود. با این وجود، من هیچ فرضی درباره میزان دانش خواننده نکرده‌ام که در نتیجه، ممکن است بخش‌های ابتدای هر فصل از نظر شما بسیار ساده باشد. موضوعات به تدریج پیشرفته‌تر می‌شوند و در اکثر فصل‌ها حتی برنامه‌نویسان مجرب نیز اطلاعات جدیدی را خواهند یافت. حتی اگر شما با موضوعات یک فصل آشنا هستید باز هم نگاهی به مثال‌ها بیاندازید. من سعی کرده‌ام که بسیاری از موضوعات پیشرفته را ساده کنم و آن‌ها را با مثال‌های واضح و کاربردی نشان دهم.

این کتاب سعی می‌کند از طریق مثال‌ها آموزش دهد. موضوعات به صورت جدا از هم و با مثال‌های کوتاه نشان داده می‌شوند و در انتهای بسیاری از فصل‌ها، شما یک برنامه بزرگ و کاربردی (یک برنامه دنیای واقعی) خواهید ساخت که موضوعات و تکنیک‌های بحث شده در کل فصل را "کنار هم جمع می‌کند". ممکن است شما برخی از برنامه‌های پیشرفته‌تر را اندکی دشوار ببابید اما نباید از آن‌ها صرف‌نظر کنید؛ ارائه برنامه‌های ساده‌تر کار مرا راحت‌تر می‌کردند اما به این ترتیب کتاب ارزشمندی نوشته نمی‌شد و دانش شما از ویژوال بیسیک کامل نمی‌گردید.

کتاب با اصول و مبانی ویژوال بیسیک شروع می‌شود که فقط بخش کوچکی از این اصول مختص نسخه ۲۰۱۰ است و اکثر آنها در همه نسخه‌های ویژوال بیسیک کاربرد دارد. در ابتدا شما نحوه طراحی ویژوال یک صفحه تصویر از طریق اشاره و کلیک با ماوس و نحوه برنامه‌نویسی چند رویداد^۱ ساده از قبیل برنامه‌نویسی کاری که با کلیک کردن ماوس روی یک دکمه باید انجام شود- را یاد خواهید گرفت. بعد از خواندن دو فصل اول، ساختار نرم‌افزارهای کارکننده تحت ویندوز را درک خواهید کرد. سپس عناصر صفحه تصویر اصلی یک نرم‌افزار (کنترل‌های اصلی ویندوز) و نحوه کدنویسی کاری که آن‌ها باید انجام دهند را فرا خواهید گرفت. همچنین،

^۱ Event یک رویداد یک کار است که کاربر در برنامه انجام می‌دهد. مثلاً کلیک کردن با ماوس یا زدن یک کلید از صفحه کلید و ... یک رویداد محسوب می‌شود.

مطالبی را درباره شیء My و قطعات کد آماده فرا خواهید گرفت: دو ویژگی که ویژوال بیسیک را ساده تر و مفرح تر می سازند. این دو شیء کل فرآیند یادگیری و نیز یادگیری ویژگی های زبان را بسیار ساده تر می کنند.

در بخش دوم اجزای اصلی نرم افزارهای تحت ویندوز به طور مفصل بررسی خواهند شد. من متداول ترین کنترل های مورد استفاده برای ایجاد فرم های ویندوز و نحوه کار با فرم ها را توضیح خواهم داد: نحوه طراحی فرم ها، نحوه طراحی منوها برای فرم ها، نحوه ایجاد برنامه هایی با چند فرم و غیره. شما بحث های مفصل بسیاری را در مورد کنترل های ویندوز و نحوه استفاده از دیالوگ باکس های از پیش تعریف شده در ویژوال بیسیک - از قبیل دیالوگ باکس های: Font و Color - در برنامه های خود، در این کتاب خواهید یافت.

ویژوال بیسیک ۲۰۱۰ یک زبان کاملاً شیء گرا^۱ است و اشیاء موضوع تکراری همه فصول هستند. بخش سوم

^۱ Object-oriented بدون دانستن این مفهوم نمی توان با زبان هایی مانند ویژوال بیسیک و ++C کار کرد. در ابتدا زبان های برنامه نویسی "کد اسپاگتی" نامیده می شدند. مثلاً اگر می خواستید در بیسیک یا فورتن برنامه نویسی کنید باید شماره خط می گذاشتید و برنامه را خط به خط می نوشتید و با عباراتی نظیر Go To و If کنترل برنامه را به جاهای دیگر می بردید. در این حالت اگر قرار بود جاهایی که کنترل برنامه به آنها می رفت را با خط روی کاغذ برنامه بکشید حاشیه کاغذ برنامه پر از خطوط درهم برهم می شد که مانند ماکارونی بود و به همین دلیل به این شیوه برنامه نویسی کد اسپاگتی گفته می شود.

در مرحله بعد زبان های "ساخت یافته" یا Structured عرضه شدند. در این زبان ها دیگر از عبارت های Go To خبری نبود؛ در عوض برنامه به صورت تکه - تکه درآمد به این معنی که برنامه یک جسم یا بدنه اصلی داشت و در کنار آن ده ها قطعه برنامه مستقل از هم بودند که هر وقت به آنها نیاز داشتید می توانستید آنها را احضار کرده، به آنها "آرگومان های" (مقادیر) مورد نیازشان را بفرستید و این قطعه برنامه ها - که به هر کدام یک "ماژول" گفته می شد - روی این آرگومان ها کار می کردند و در نهایت حاصل کار خود را به برنامه اصلی باز می گردانند از این نوع زبان ها می توان به C و پاسکال اشاره کرد.

در مرحله بعد زبان های "شیء گرا" یا Object-Oriented عرضه شدند. در زبان های ساخت یافته ماژول ها تا حدودی مستقل بودند اما در نسل بعدی زبان ها به هر تکه برنامه یک "شیء" گفته شد یک شیء دیگر کاملاً مستقل شد؛ آنقدر مستقل که مانند پیچ و مهره می توانید آنها را از یک برنامه برداشته و در برنامه دیگر مورد استفاده قرار دهید. توجه کنید که زحمت برنامه نویسی چقدر کم شد از طرف دیگر اشیاء خواص بسیار جالبی پیدا کردند: ارث بری، چندشکل پذیری و کپسوله شدن. با ارث بری دیگر نیازی به برنامه نویسی مجدد نبود. مثلاً اگر برنامه ای برای توصیف یک چرخ دوچرخه نوشته باشید، برای نوشتن برنامه یک چرخ ماشین یا تراکتور دیگر نیازی به نوشتن از ابتدا نیست. شما تعریف می کنید که شیء جدید فرزند شیء قبلی است و به این ترتیب فقط خواصی که مختص ماشین یا تراکتور است را برنامه نویسی می کنید. با چندشکل پذیری می توان یک متد تعریف کرد و بعد از آنرا کمی دستکاری کرد و از آن استفاده دیگری نمود و با کپسوله کردن هم یک شیء هم محتوی داده شد و هم متدهایی که باید روی این داده ها کار کنند. به این ترتیب برنامه ها بسیار کوچکتر و انعطاف پذیرتر شدند و در ضمن از قطعات یک برنامه می شود در برنامه های دیگر استفاده کرد.

زبان های دلفی و ++C از زبان های شیء گرا هستند.

بعد زبان های ویژوال عرضه شدند. اینها همه خواص شیء گرایی را داشتند و در ضمن در آنها خیلی از کارها با ماوس انجام می شد و نیازی به برنامه نویسی نداشت. مثلاً هر نرم افزار تازه ای که بخواهید بسازید یک پنجره اصلی دارد که به محض باز شدن نرم افزار دیده می شود. حالا دیگر نیازی به نوشتن برنامه برای رسم چارگوش دور، نوار عنوان بالای پنجره، نوار منوها و ... نیست. با یک دستور یک منو در پنجره نرم افزار قرار می گیرد و شما فقط دستوراتی که می خواهید منو داشته باشد را درون آن تایپ می کنید. بعضی از این دستورات در همه نرم افزارها مشترکند (مانند: Open و Save) و بعضی دیگر مختص همین نرم افزار هستند که باید برای آنها برنامه نویسی یا "کدنویسی" کنید. بدیهی است که به این ترتیب طراحی نرم افزار بسیار ساده تر می شود. از مشهورترین زبان های ویژوال شیء گرا می توان به ویژوال بیسیک، ویژوال ++C و دلفی های نگارش های اخیر اشاره کرد.

در مرحله بعد زبان ها هم ویژوال و هم شیء گرا بودند اما یک خصلت دیگر هم به آنها اضافه شد و آن استفاده از امکانات "بستر داتنت" بود. مزایای بستر داتنت در ابتدای کتاب گفته شد. از زبان هایی که در این بستر کار می کنند می توان به ویژوال بیسیک و #C (سی شارپ) اشاره کرد.

کتاب (فصل‌های ۸، ۹ و ۱۰) حاوی یک نگرش رسمی‌تر و سیستماتیک‌تر به اشیاء است. شما نحوه ایجاد کنترل‌ها و کلاس‌های خودتان را یاد خواهید گرفت که در درک بهتر برنامه‌نویسی شیء‌گرا به شما کمک خواهند کرد. همچنین درباره "ارث‌بری" خواهید آموخت و خواهید دید که اضافه کردن یک عملکرد دلخواه به کلاس‌های موجود، از طریق وراثت چقدر آسان‌تر از برنامه‌نویسی مجدد است.

بخش چهارم با برخی از کلاس‌های متداول بستر دانت سر و کار دارد. بستر دانت قلب برنامه‌نویسی ویندوز است؛ این بستر دروازه ورود به عملکرد خود سیستم عامل و استفاده از امکانات آن است. اولین فصل در این بخش از کتاب، مقدمه‌ای بر بستر دانت به طور کلی است و نحوه استفاده از کلاس‌های اصلی این بستر برای کار بر روی فایل‌ها و پوشه‌ها، نحوه کار با داده‌ها و زمان، نحوه کار با بازه‌های زمانی، نحوه ایجاد گرافیک و خروجی‌های جذاب چاپی و سایر امکانات بستر دانت را به شما نشان خواهد داد. در فصل بعد شما نحوه استفاده از کلکسیون‌ها در کد خود را یاد خواهید گرفت و سپس یک فصل درباره XML و یک فصل درباره LINQ خواهید یافت. شما خواهید دید که از طریق این زبان جدید که در داخل ویژوال بیسیک از پیش تعریف شده است نوشتن کدهای XML و استفاده از آنها در کدهای ویژوال بیسیک و در کوئری کردن از کلکسیون‌ها و بانک‌های اطلاعاتی چقدر آسان شده است. این زبان جدید "زبان کوئری" (پرس و جوی) یکپارچه (LINQ) است. LINQ داغ‌ترین فناوری جدیدی است که به شما امکان می‌دهد تا در داده‌هایی با ساختارهای بسیار متفاوت و داده‌های بدست آمده از منابع مختلف با روشی یکسان پرس و جو کنید.

چهارده فصل اول به مبانی زبان و نرم‌افزارهای تحت ویندوز می‌پردازند. بعد از این فصل‌ها مروری بر ابزارهای دسترسی به داده‌ها خواهید داشت (فرض من این است که اکثر شما در نهایت یک برنامه داده محور خواهید ساخت). در بخش پنجم تأکید روی کار با ابزارهای ویژوال است و شما نحوه پرس و جو در بانک‌های اطلاعاتی و روش‌های ارائه داده‌ها به کاربر را خواهید آموخت. همچنین، شما اطلاعات مربوط به برنامه‌نویسی اشیاء اصلی ADO.NET و نوشتن برنامه‌های داده محور را خواهید یافت.

چند فصل آخر این کتاب درباره: مشخصات نرم‌افزارهای کارکننده در وب (اینترنت)، مبانی ASP.NET 4، نحوه ایجاد برنامه‌های کارکننده در وب و نحوه نوشتن "سرویس‌های وب" است. با توجه به این که من نمی‌توانستم طراحی هر دو نوع برنامه‌های کارکننده تحت ویندوز و تحت وب را با جزئیات یکسان مورد بحث قرار دهم، تصمیم گرفتم که ابتدا روی برنامه‌های کارکننده تحت ویندوز تمرکز کنم و در چند فصل آخر (بخش ششم) نحوه به کارگیری دانش شما در طراحی برنامه‌های کارکننده تحت وب را نشان دهم. در برنامه‌های کارکننده تحت وب، در حالی که صفحه تصویر اصلی برنامه کاملاً متفاوت است، اما کدهای مورد نیاز همان کدهایی هستند که در برنامه‌های کارکننده تحت ویندوز به کار می‌روند.

^۱ در بانک‌های اطلاعاتی شما اطلاعات وارد می‌کنید ولی این اطلاعات باید تجزیه و تحلیل شوند و از آنها نتایج مورد نظر به دست آید. این تجزیه و تحلیل از طریق پرسش از بانک اطلاعاتی انجام می‌گیرد. مثلاً در بانک اطلاعاتی یک اداره شما ممکن است تعداد افراد بالای پنجاه سال یا آنهایی که بیش از دو فرزند دارند را بخواهید. به این نوع پرسش‌ها "کوئری" (Query) گفته می‌شود.

ضمائم را از دست ندهید

در انتهای کتاب چند ضمیمه قرار دارد که عبارتند از:

- حداکثر استفاده از کنترل‌های ListView (نمایش لیستی) و TreeView (نمایش درختی)
- چاپ در ویژوال بیسیک
- دسترسی به فایل‌ها و پوشه‌ها
- ایجاد گرافیک در ویژوال بیسیک ۲۰۱۰

این فصل‌ها بازمانده از نسخه ویژوال بیسیک ۲۰۰۸ این کتاب هستند. در سراسر این کتاب، ارجاعاتی به این ضمیمه‌ها خواهید یافت. این‌ها فصل‌های کاملی هستند که دارای پروژه‌های نمونه و همان ساختار فصل‌های کتاب هستند.

دانلود کردن این کتاب

کد مثال‌ها و پروژه‌های این کتاب را می‌توانید از وب سایت www.sybex.com دانلود کنید. در صفحه تصویر اصلی این سایت، ابتدا صفحه وب این کتاب را با جستجوی نام مؤلف، عنوان یا ISBN (9780470187425) پیدا کنید و سپس در نتایج جستجو، روی لینک این کتاب کلیک کنید. در صفحه وب کتاب، روی لینک Download کلیک کنید تا به صفحه دانلود بروید. یا می‌توانید مستقیماً به صفحه وب کتاب به آدرس www.sybex.com/go/masteringvb2010 بروید. فایلی که دانلود می‌شود یک فایل ZIP است که شما می‌توانید با برنامه WinZip آن را از حالت فشرده خارج کنید (در DVD همراه این کتاب این فایل‌ها وجود دارند).

نمونه دسترسی به مؤلف

علیرغم اینکه ما حداکثر سعی خود را کرده‌ایم، به احتمال قوی کتابی با این حجم حاوی خطاهایی است. اگرچه به روز رسانی یک رسانه چاپی به آسانی یک وب سایت نیست، اما من تمام تلاش خود را به کار می‌برم تا هر مشکلی را که شما گزارش می‌دهید (یا خودم پیدا می‌کنم)، رفع کنم. برنامه‌های تجدید نظر شده به همراه سایر مطالبی که به نظر من باید مورد استفاده خوانندگان این کتاب قرار گیرند در سایت Sybex منتشر خواهد شد. اگر شما با هر مشکلی در متن یا برنامه‌های این کتاب مواجه شدید، می‌توانید مستقیماً از طریق ایمیل:

pevangelos@yahoo.com

با من تماس بگیرید.

هرچند من نمی‌توانم قول بدهم که به تک تک سؤالات پاسخ می‌دهم، اما همه مشکلات مثال‌ها را رفع و نسخه‌های به روز را فراهم خواهم کرد.

همچنین، مایل به نظرات و توضیحات شما درباره کتاب، موضوعاتی از کتاب که دوست داشتید یا نداشتید و میزان مفید بودن مثال‌ها را بدانم. توضیحات شما در چاپ‌های بعدی کتاب با دقت کامل مد نظر قرار خواهند گرفت.