

ویژوال بیسیک پیشرفته

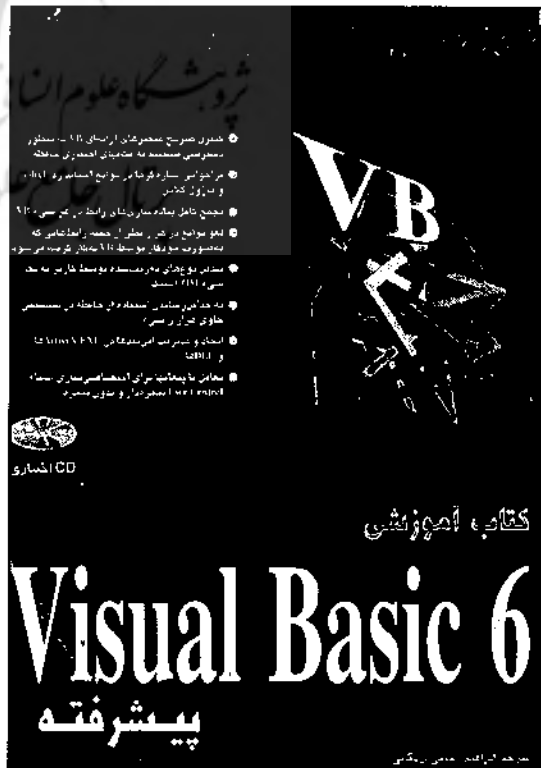
ویژوال بیسیک در سطح بندی زبانهای برنامه نویسی، جزو زبانهای سطح بالا می باشد. این نرم افزار شامل ابزارها و خصوصیات تقریباً کاملی برای تولید سریع و آسان برنامه های کاربردی با خطای کمتر نسبت به دیگر زبانهای برنامه نویسی می باشد و به همین دلیل مورد علاقه بسیاری از برنامه نویسان حرفه ای می باشد. تنها موردی که این برنامه نویسان را نگران می کند این است که پروژه نتواند در ویژوال بیسیک کامل شود، زیرا این نرم افزار به آنها کنترل سطح پایینی روی برنامه نمی دهد تا بتوانند بطور کامل سیستم را کنترل کنند. ویژوال بیسیک برای ایجاد اشیاء تا مدیریت آرایه ها و رشته ها با Com و خودکارسازی OLE به میزان زیادی کار می کند. Com مبتنی بر یک استاندارد دو دویی می باشد، بنابراین طرح حافظه تمام انواع Com به خوبی مستندسازی شده و به راحتی تکرار می شود. محورهای اصلی مطرح شده در کتاب ویژوال بیسیک ۶ پیشرفته را نوع های Com و همچنین اینکه چگونه به ویژوال بیسیک یک Boost بدهید بطوریکه ویژوال بیسیک بهتر اجرا شود و قابلیت هایی که موجود نیستند را اجرا نماید، تشکیل می دهند.

کنترل صریح متغیرهای آرایه ای ویژوال بیسیک به منظور دسترسی مستقیم به مکانهای اختیاری حافظه، فراخوانی اشاره گرها در توابع استاندارد، cdecl و مازول کلاس، تجمع کامل پیاده سازی های رابط در هر شی و ویژوال بیسیک، لغو توابع در هر رابطی از جمله رابط هایی که بصورت خودکار توسط ویژوال بیسیک بکار گرفته می شوند، تبدیل نوع های تعریف شده توسط کاربر به یک شی Com سبک، به حداقل رساندن استفاده از حافظه در سیستمی حاوی هزاران شی، ایجاد مدیریت امن بندها در ActiveX EXE ها و Dll ها و تعامل برای اختصاصی سازی اشیاء User Control پنجره دار و بدون پنجره خلاصه ای از موضوعاتی هستند که در این کتاب مورد بحث قرار گرفته شده اند.

○ مهندس محمد حقانی

○ میتو کرلند (Matthew curland).

ویژوال بیسیک ۶ پیشرفته، ترجمه ابراهیم رحیمی لردگانی، تهران: شایگان، چاپ دوم، ۱۳۸۰، ۴۰۰ صفحه، فارسی، وزیری، شمیم





Basic Visual

متغیر آرایه علاوه بر بخش داده، دسترسی مستقیم به حافظه را نیز می‌توانید داشته باشید.

در فصل سوم نویسنده به توضیح رابط کمیت نامعلوم پرداخته است. VB اشیاء Com را ایجاد و مورد استفاده قرار می‌دهد. تمام اشیاء Com از رابط کمیت نامعلوم مشتق می‌شوند.

کمیت نامعلوم هیچ نوع بومی متناظری در ویژوال بیسیک ندارد. این واقعیت که ویژوال بیسیک اجازه تعریف یک متغیر با یک نوع کمیت نامعلوم بومی را نمی‌دهد هنگامی که عملیاتی روی شی سطح پایین انجام می‌دهد مانند ارجاع ضعیف و روش‌های مجتمع‌سازی که در این کتاب توضیح داده می‌شوند، باعث بروز پیچیدگیها و مشکلاتی خواهد شد. این فصل نگاهی دقیق به استفاده ویژوال بیسیک از رابط کمیت نامعلوم و چگونگی ارسال تعریف و فراخوانی رابط کمیت نامعلوم در ویژوال بیسیک می‌اندازد در فصل بعد مقید کردن توابع به اشیاء مورد بحث قرار گرفته شده است. اگر نتوانید با یک شی هر کاری که می‌خواهید انجام دهید چندان مناسب نخواهد بود برای فراخوان یک تابع عضو در یک شی، VB باید یک عملیات که تحت عنوان مقیدسازی (binding) نامیده می‌شود را انجام دهد. تمام انواع مقیدسازی‌ها از داده ثابت فرمان کامپایل به همراه مقدار پویایی زمان اجرای اشاره‌گر شیء برای یافتن تابعی که باید فراخوانی شود، استفاده می‌کنند. در این فصل هر دو دسته از مقیدسازی‌ها (Vtable و IDispatch) بطور مفصل شرح داده شده‌اند.

در فصل پنجم نویسنده به ساختارهای طراحی شده شی پرداخته است. معماری کد و طراحی پایه‌های اصلی، موفقیت هر پروژه نرم‌افزاری می‌باشند. اگر طراحی ضعیفی داشته باشید در پایان چرخه تولید حجم زیادی از کد باید بنویسید که بتوانید سیستمی را که به طور ذاتی ناپایدار است پایدار کنید. این

نویسنده اهداف اصلی تألیف این کتاب برای خوانندگان را چنین بیان کرده است: یاد بگیرند که چگونه استاندارد ویژوال بیسیک بهتری بنویسند، یاد بگیرند از مجموعه جدیدی از تکنیک‌های پیشرفته برنامه‌نویسی ویژوال بیسیک استفاده کنند، و با پیروی از راهنمایی‌های ویژوال بیسیک، سعی کنند تکنولوژی سطح پایین را در این کتاب به صورت یک بسته درآورند بدون اینکه لازم باشد بدانند در پشت پرده چه اتفاقی می‌افتد.

کتاب شامل یک مقدمه، ۱۶ فصل و یک ضمیمه شامل مرجع VBoost می‌باشد.

در فصل اول به مبانی بلوکهای ساختمانی در ویژوال بیسیک اشاره شده است. ایده اساسی در پشت تمام توسعه‌های ویژوال بیسیک یک زبان مبتنی بر Com می‌باشد. انواع داده‌های ویژوال بیسیک، همان انواع داده‌ای Com هستند که بصورت مناسبی مستندسازی و معرفی شده‌اند. این کتاب به میزان زیادی روی دستکاری، آرایه حافظه ویژوال بیسیک و انواع شی تأکید می‌کند و برای دسترسی به چنین داده‌های سطح پایینی نیازمند دسترسی مستقیم به حافظه که به عنوان دستکاری اشاره‌گری شناخته شده است می‌باشد. VB پشتیبانی صریحی برای عملیات اشاره‌گری فراهم نمی‌سازد. به اندازه‌ای که بتوان بعضی از عملیات ساده را انجام داد را شامل می‌شود. در فصل بعد نویسنده به کاربرد آرایه‌ها پرداخته است. آرایه‌ها ابزاری ضروری برای هر برنامه‌ریزی جدی می‌باشند. آرایه‌ها برنامه را قادر می‌سازند یک جهش سریع از عمل کردن روی یک آیتم تا انجام عملیات یکسان روی تعداد دلخواهی آیتم را انجام دهد. آرایه‌ها همچنین شما را قادر می‌سازند یک ترتیب روی چندین آیتم اعمال کنید.

VB به طور خودکار به شما اجازه نوشتن در قسمت داده یک آرایه را می‌دهد ولی با خواندن و نوشتن در توصیف‌گر و

فصل نگاهی به روشهای VB برای قوانین سعی و تصدیق^۲ اجرای نصب شدنی، مجددسازی و استفاده مجدد کد می‌اندازد. ارجاعات چرخه‌ای در فصل بعد مورد بررسی قرار گرفته است. تعداد ارجاعات زمان حیات هر شی Com را کنترل می‌کند وقتی از یک دستور Set برای انتساب یک عبارت شیء به یک متغیر شیء استفاده می‌کنید، متغیر صاحب یک ارجاع به شیء می‌شود. ولی دارا بودن یک ارجاع به یک شیء

به این معنا نیست که آن متغیر مالک شیء می‌باشد. اگر نمونه مشابهی از یک شیء به دو متغیر در یک زمان انتساب داده شود، هر متغیر دارای یک شماره ارجاع روی شیء خواهد بود هیچ تمایزی بین متغیرها قائل نمی‌شود هر ارجاع دارای وزن یکسان است.

در فصل هفتم نویسنده به ایجاد شیء خارجی پرداخته است. اشیاء عمومی به میزان وسیع در Com استفاده می‌شوند یکی از وظایف شما به عنوان برنامه‌نویس ویژوال بیسیک این است که مجموعه‌ای از اشیاء عمومی فراهم آورید که خود و دیگر اعضای گروه بتوانید از آنها برای استفاده در ایجاد برنامه کاربردی کمک بگیرید. درک چگونگی ایجاد اشیاء خارجی یک گام مهم برای استفاده از آنها

می‌باشد این فصل به بررسی چگونگی ایجاد اشیاء خارجی با استفاده از New و Creat object می‌پردازد و جایگزین‌های مبتنی بر API برای ایجاد شیء را بررسی می‌کند.

در فصل بعد اشیاء Com کم‌وزن مورد بحث قرار گرفته شده است. زمان حیات یک شیء Com با یک مکانیزم شمارش ارجاع کنترل می‌شود و این امر اجازه می‌دهد در یک زمان چندین ارجاع بر یک شیء داشته باشید.

VB دارای یک مکانیزم درونی برای ساختارهای ارجاعی نمی‌باشد (این ساختارها نوع‌های تعریف شده توسط کاربر نیز نامیده می‌شوند) زیرا هیچ مکانیزم درونی برای شمردن ارجاع یک حافظه ساختار وجود ندارد در این فصل چگونگی «ایجاد اشیاء کم‌وزن» (آنها را به عنوان ساختارهایی که اشیاء Com هستند تعریف می‌کنیم) را خواهید دید.

در فصل نهم سیستم‌های بزرگی از اشیاء توضیح داده شده است. ماژول‌های VB برای ایجاد تعداد کمی اشیاء با پیچیدگی زیاد استفاده می‌شوند ولی بالانسری سرباری هر شیء آن را به منظور ایجاد تعداد زیادی اشیاء ساده به مقدار بازدارنده‌ای

پرهزینه می‌سازد. طیف الگوریتم‌ها از ابزار زمان‌بندی تا کامپایلرها اغلب بجای تعداد کمی شیء نیازمند تعداد زیادی شیء می‌باشند. در این فصل خواهیم دید چگونه از یک مدیر حافظه سفارشی برای تخصیص اشیاء کم‌وزن بدون تن دادن به تجربه کدنویسی که از این اشیاء استفاده می‌کند، را به کار بگیرید.

در فصل بعد نویسنده به اشیاء ویژوال بیسیک و جدول شیء

اجرائی پرداخته است. جدول شیء اجرائی (مخفف آن ROT) می‌باشد یک جدول سیستم سراسری است که یک نام را به یک شیء اجرائی فعلی مرتبط می‌سازد جدول شیء اجرائی یک انبار مرکزی است که به اشیاء امکان می‌دهد بدون نیاز به فرآیندهایی برای تعریف پروتکل‌های ارتباطی سفارشی مابین محدوده‌های فرآیندها، مکان‌یابی شوند. وقتی یک فرآیند یک شیء خود را در جدول شیء اجرائی ثبت می‌کند، اعلام می‌کند که آماده است آن شیء را با هر فرآیندی که نام شیء را می‌داند به اشتراک بگذارد اگر فرآیند دیگری نام شیء را نمی‌داند، نمی‌تواند شیء را مورد استفاده قرار دهد. اگر دو شیء با یک نام در جدول شیء اجرائی ثبت شوند،



جدول شیء اجرائی تعریف نکرده است که به کدام شیء می‌توانید دسترسی بیابید. در فصل یازدهم نویسنده به فراخوانی اشاره‌گرهای تابع پرداخته است.

در فصل یازده نشان داده می‌شود که چگونه یک نمونه شیء را با یک اشاره‌گر تابع مرتبط بسازید (به میزان زیادی عملیاتهای استاندارد مثل زیرکلاس‌سازی پنجره‌ها را ساده می‌سازد) و چگونه توابع CDeCL را احضار کنید که شما را قادر می‌سازد نقاط ورود در MSVCRT و دیگر DLL‌های CDeCL احضار کنید. نویسنده در فصل بعد پیشگیری از وقوع توابع را شرح داده است. کلمه کلیدی implements ویژوال بیسیک روش مبتنی بر رابط را برای برنامه‌نویسی پشتیبانی می‌کند. در طول کتاب بر طراحی با رابط‌ها تاکید شده است. زیرا یک مدل عالی برای انتزاع و استفاده مجدد از فراخوانی کد می‌باشد. اشیاء مجتمع‌سازها VBoost به خوبی امکان استفاده مجدد از پیاده‌سازی‌های رابط کامل را می‌دهند. این فصل به شما نشان می‌دهد که چگونه توابع را با استفاده از اشاره‌گرهای تابع برای راهنمایی مجدد فراخوانی‌ها از کلاس

پایه یا با سفارشی کردن تکنیک‌های تجمع‌سازی برای اشیاء مجزا به منظور لغو توابع بدون هیچ‌گونه مشارکتی از کلاس پایه لغو کنید.

در فصل سیزدهم نویسنده به بندها در ویژوال بیسیک پرداخته است. ویژوال بیسیک ایجاد EXE ها، DLL ها، OXC های چندبندی را پشتیبانی می‌کند. پشتیبانی از چندین بند به چندین معناست قابلیت اجرای همزمان چندین بند کد در یک فرآیند، قابلیت فراخوانی کدی که در بند دیگری در حال اجراست، یا قابلیت اجرای همزمان چند بند روی مجموعه یکسانی از داده‌ها، ویژوال بیسیک اجرای کد چندبندی را پشتیبانی می‌کند ولی ذاتاً بندکشی آزاد را که قابلیت اجرای چند بند روی مجموعه یکسانی از داده می‌باشد را پشتیبانی نمی‌کند. VB می‌تواند قطعات دیگری که بندکشی آزاد را پشتیبانی می‌کند، فراخوانی نماید. این فصل سعی می‌کند اصطلاح چندبندی را توصیف کند و به شما نشان می‌دهد چگونه از قابلیت‌های بندکشی ویژوال بیسیک بطور کامل استفاده کنید. در فصل بعد رشته‌ها در ویژوال بیسیک شرح داده شده‌اند. تاکنون به یک برنامه ویژوال بیسیک که از رشته‌ها استفاده نکرده باشد برخورد نکرده‌ایم.

برنامه‌ها به منظور فعل و انفعال به کاربران یا تولید چندین خروجی طراحی شده‌اند که هر دو حالت، نیازمند متن می‌باشند در برنامه‌های زیادی که برای اینترنت نوشته می‌شود، رشته‌ها هر روز نقش بزرگتری را بازی می‌کنند. HTML و XML و قالب‌های اینترنتی دیگر بسیار وابسته به رشته می‌باشند. و اغلب نیازمند ایجاد رشته‌های بزرگی هستند این باعث می‌شود کارایی عملیاتی رشته‌های نقش اصلی در کارایی برنامه کاربردی را ایفا کنند. این فصل در پی این است که با هدف افزایش کارایی رشته‌ها، نگاهی به چگونگی تعریف و تعامل رشته‌ها توسط ویژوال بیسیک بیندازد. هرچند چندین کلاس کمکی مفید معرفی خواهد شد در این فصل در پی فراهم آوردن کتابخانه کاملی از توابع رشته‌ای نمی‌باشد. در مقابل این فصل ابزاری برای نوشتن روتین‌های رشته‌ای با حداکثر بهینه‌سازی را ارائه می‌دهد.

در ادامه فصل قبل، فصل پانزدهم به کتابخانه‌های نوع ویژوال بیسیک پرداخته است. کتابخانه‌های نوع زبان توصیف بایندی را می‌سازند که VB برای مصرف اشیاء و تشریح اشیایی که ایجاد می‌کند از آن استفاده می‌کند. قبل از اینکه بتوانید اشیاء موجود در پروژه خود را کامپایل کنید، پروژه باید یک کتابخانه نوع را که یک قطعه اکتیوایکس ایجاد می‌کند، ویژوال بیسیک یک کتابخانه نوع تولید می‌کند. به طور خلاصه، بدون کتابخانه‌های نوع حتی قادر به نوشتن اولین خط از کد در یا تولید قطعات اکتیوایکس سفارشی نخواهید بود. این فصل به شما نشان می‌دهد چگونه type Lib های مورد استفاده را ایجاد کنید.

در فصل آخر پنجره‌های کنترلی شرح داده شده‌اند اغلب برنامه‌های سرویس‌گیرنده ویژوال بیسیک حول پنجره‌ها ایجاد می‌شوند در واقع ویژوال بیسیک اولین موفقیتش را در دنیای برنامه‌نویسی به دلیل ساده کردن ایجاد برنامه‌های کاربردی ویندوز با دور نگه داشتن شما از پیچیدگی‌های گفتگوهای مستقیم با اشیاء پنجره به دست آورده است، اما این لایه حفاظتی در صورتی که بخواهید در سطح ایجاد و رویه پنجره با ویندوز گفتگو کنید، می‌تواند مقداری ناامیدکننده باشد، این فصل مفاهیم اشاره‌گر تابع، دسترسی مستقیم به حافظه اشیاء سبک و تکنیک‌های لغو از فصل‌های پیشین را به اشیاء پنجره بطور عمومی و کنترل‌های سفارشی بطور خاص اعمال می‌کند. با فراگیری زیرکلاس‌سازی، به سمت ایجاد پنجره سفارشی و کنترل‌های بدون پنجره که دقیقاً مانند کنترل‌های پنجره‌دار عمل می‌کنند، حرکت خواهیم کرد.

ضمیمه کتاب شامل مرجع VBoost می‌باشد. VBoost کتابخانه کوچکی از توابع می‌باشد که سناریوی تکنولوژی پایه موردنیاز تمام تکنیک‌های این کتاب را فراهم می‌سازد. کتاب برای برنامه‌نویسان پیشرفته و حرفه‌ای ویژوال بیسیک نوشته شده است. به همین دلیل دیگر موارد مقدماتی ویژوال بیسیک را شامل نمی‌شود. کتاب‌های ترجمه شده موجود در بازار به ویژگی‌هایی مانند ارتباط با بانک‌های اطلاعاتی و یا برنامه‌نویسی اینترنت با ویژوال بیسیک پرداخته‌اند و این تقریباً اولین کتابی است که به موارد پیشرفته این نرم‌افزار پرداخته است. از جمله این موارد اشیاء Com می‌باشد که باعث دسترسی به ویژگی‌هایی است که در درون VB یافت نمی‌شود. کتاب کمک شایانی به برنامه‌نویسان حرفه‌ای ویژوال بیسیک می‌نماید و تا اندازه‌ای به دانش و تجربه عملی قبلی در مورد Com نیازمند است. مرجع VBoost که در ضمیمه انتهایی کتاب آمده است، بسیار سودمند است زیرا VBoost شامل اشیاء و توابعی است که قابلیت‌هایی را فراهم می‌آورد که در ویژوال بیسیک موجود نمی‌باشد از جمله این قابلیت‌ها مدیریت حافظه و مجموعه‌ای از توابع حسابی است که به میزان زیادی دستکاری اشاره‌گرها در ویژوال بیسیک را ساده می‌کند. علاوه بر کد، VBoost شامل مجموعه‌ای از typelib می‌باشد که براساس عملکرد طبقه‌بندی شده‌اند. این typelib ها باعث کامپایل سریع تمامی فایل‌های موجود در دایرکتوری Code در سی دی می‌شوند. از دیگر مزایای این کتاب می‌توان به نمونه برنامه‌ها و کدهای ارائه شده در میان فصل‌ها اشاره کرد که بخوبی مطالب ارائه شده را پوشش می‌دهند.

پاورقی:

- ۱- Un Known
- ۲- Triedand True