

Содержание

Предисловие	10
Глава 1. Языки программирования и объектная модель	12
Обзор средств разработки	13
Об объектной модели AutoCAD	14
Объекты и коллекции.....	14
Интерфейсы.....	19
Свойства	19
Методы	19
События	19
Простой пример использования объектной модели	19
Особенности программирования для AutoCAD 2015.....	20
Доверенные каталоги	20
О совместимости	21
Глава 2. C++ и ObjectARX.....	24
Структура пакета ObjectARX.....	26
Интеграция справочной системы ObjectARX с Visual Studio.....	27
Настройка проекта в среде Visual Studio	27
Создание проекта в Visual Studio 2010 (AutoCAD 2013–2014)	27
Настройка свойств проекта	30
Создание проекта в Visual Studio 2012 (AutoCAD 2015)	36
Пример простого приложения	37
Мастера ObjectARX.....	43
Установка мастеров ObjectARX.....	44
Виды ARX-приложений, создаваемых с помощью мастера ObjectARX.....	46
Поддержка разных версий AutoCAD.....	52
Создание команд и функций.....	53
Структура файла acrxEntryPoint.cpp.....	53
Пример создания команды USERMSG	57
Пример создания LISP-функции callmymsg	57
Контекст.....	58
Загрузка и выгрузка ARX-приложений	58
Библиотеки	59
Поставляемые примеры	60

Использование отладчика	61
Выполнение команд AutoCAD	63
Примеры использования acedCommand и acedCmd	64
Особенности AutoCAD 2015	68
Пример использования acedCommandS	68
Пример использования acedCommandC	69
Структура resbuf и буфер результатов	70
Построение примитивов	72
Использование DXF-кодов	73
Доступ к базе чертежа	74
Использование классов примитивов	77
Передача данных между ARX и LISP	82
Геометрические вычисления	83
Работа с Windows API	88
Окно MessageBox	88
Пояснение к использованию acedGetArgs	91
Использование MFC	92
Индикатор в строке состояния	93
Модальные диалоговые окна	95
Немодальные диалоговые окна	104
Отступление о DCL-окнах	108
ObjectDBX	110
.NET и управляемый код	111
Поддержка управляемого кода	111
Глава 3. Интеллектуальные объекты	116
Особенности DBX-приложений	117
Прокси-объекты	117
Заготовка DBX-приложения с помощью мастера	120
Структура файла acrxEntryPoint.cpp	120
Заготовка класса нового объекта	121
Некоторые особенности заготовки класса объекта	124
Доработка проекта Custdbx	127
Новый вариант файла Ship.h	129
Конструкторы	131
Метод storePoints	132
Метод subWorldDraw	133
Метод subExplode	134
Метод subTransformBy	135
Метод dwgOutFields	136

Метод dwgInFields	137
Метод dxfOutFields	138
Метод dxflnFields	139
Метод subGetGripPoints	141
Метод subMoveGripPointsAt	142
Метод subList	143
Настройки Custdbx перед построением	144
ARX-проект Custarx	145
Команда CSHIP	146
Класс ShipJig	148
Глава 4. .NET и плагины (C#)	153
Платформа .NET Framework	154
.NET-приложения	155
Операции с реестром	155
Конфигурационный файл acad.exe.config	163
Документация к AutoCAD .NET API	165
Каркас приложения с использованием мастера Autodesk	167
Новый проект	167
Шаблоны команд и LISP-функций	169
Прежде чем приступить к написанию плагина «с нуля», без мастера	171
Создание проекта «с нуля» без использования мастера Autodesk	173
Создание проекта	173
Ресурсы для локализации	175
Написание кода	181
Результаты	188
Локальная и удаленная отладка	189
Общая идея	190
Настройка	191
Запуск	193
Единый код для разных версий	194
Работа с Database: с использованием транзакций и без них	197
Редактирование базы данных чертежа без использования транзакции	197
Редактирование базы данных чертежа с использованием транзакции	203
О блокировке документа	207
Интерфейс IDisposable	207
Работа со стилями	209
Размерный стиль	209

Стиль мультивыносок	227
Стиль мультилиний.....	235
Стиль точек	238
Стиль таблиц	239
Текстовый стиль	266
Особенности программного запуска команд в AutoCAD 2015	271
Пример использования Command	271
Пример использования CommandAsync	271
Графический интерфейс. Windows Forms и WPF	273
Windows Forms.....	274
Модальное окно	274
Немодальное окно.....	276
«Прозрачное» добавление стандартным классам AutoCAD недостающего им функционала.....	277
Размещение плагинов. Доверенные каталоги.....	281
Программная загрузка и выгрузка плагинов AutoCAD	282
Использование AcCoreConsole	284
Глава 5. AutoLISP и DCL	285
Интерпретатор LISP.....	286
Программа и комментарии.....	288
Символы.....	288
Загрузка программы	291
Перечень основных функций	292
Зарезервированные символы	311
Примеры программ	313
Среда разработки приложений.....	314
Обзор меню.....	316
Панели инструментов Visual LISP	329
Написание и проверка синтаксиса программ в текстовом редакторе среды Visual LISP	330
Разработка приложений на языке LISP	331
Средства отладки программ.....	331
Сообщения о неверном типе	333
Проекты и FAS-приложения.....	334
VLX-приложения.....	337
Выполнение команд AutoCAD из программы пользователя.....	342
Программа рисования штампа чертежа.....	343
Особенности программного выполнения команд в AutoCAD 2015.....	346

Оформление функций пользователя в виде новых команд AutoCAD.....	347
Работа с примитивами.....	348
Доступ к табличным данным и неграфическим объектам.....	350
Расширенные данные.....	351
Другие способы хранения данных.....	354
ActiveX.....	355
Функции, реализующие ActiveX.....	355
Новые типы данных.....	356
Работа с документами и примитивами.....	357
Работа с меню.....	360
Реакторы.....	362
Применение COM.....	369
Создание документа Word.....	370
Пример формирования документа.....	378
Язык DCL.....	380
Директивы.....	381
Директива описания диалога.....	384
Элементы диалога.....	387
Типы элементов.....	387
Элементы с кнопкой выхода.....	388
Атрибуты элементов.....	388
Открытие и закрытие диалоговых окон.....	391
Функция load_dialog.....	391
Функция unload_dialog.....	391
Функция new_dialog.....	392
Функция start_dialog.....	392
Функция done_dialog.....	393
Функция term_dialog.....	393
Пример вызова диалогового окна.....	393
Функции, работающие с элементами диалога.....	394
Функция action_tile.....	394
Функция mode_tile.....	395
Функция set_tile.....	396
Функция get_tile.....	400
Функция get_attr.....	400
Функция client_data_tile.....	401
Переменные действия.....	401
Функции, работающие со списками.....	402
Функция start_list.....	402
Функция add_list.....	403

Функция end_list	403
Функции, работающие с графическими элементами	403
Функция start_image	403
Функция end_image	404
Функция fill_image	404
Функция slide_image	405
Функция vector_image	405
Функция dimx_tile	405
Функция dimy_tile	406
Дополнительные приемы программирования диалоговых окон	406
Временный выход из диалогового окна	406
Вложенные диалоговые окна	408
Применение паролей	409
Интеграция приложений	410
Глава 6. VBA	411
Выполнение отдельных выражений VBA	412
Использование ActiveX	412
Среда VBA	413
Меню	413
Панели инструментов	414
Проекты и их элементы	415
Макросы	417
Модули и процедуры	419
Формы	419
Типы данных	422
Обычные типы	422
Массивы	423
Специальные типы	423
Область видимости	424
Константы и перечисления	425
Простой макрос	425
Выполнение команд AutoCAD	428
Создание формы	431
Элементы управления	431
Отображение и скрытие диалогового окна	433
Пример приложения с модальным диалоговым окном	433
Модуль программного кода	433
Типы процедур	435
Добавление программного кода работы с формой	435

Сжатие приложения.....	436
Защита приложения	436
Автоматическая загрузка приложений.....	439
Обработка ошибок.....	439
Оператор On Error	440
Объект Err	440
Отладка	441
Передача данных между VBA и LISP-средой	443
Работа с документами	444
Имена открытых чертежей.....	444
Доступ к неактивному документу.....	445
Расширенные данные и словари	446
Работа с меню	446
Получение наименований меню	447
Динамическое изменение контекстного меню.....	448
Применение Windows API.....	449
Литература.....	451
Алфавитный указатель	453

Предисловие

Тема программирования в системе AutoCAD не нова, и ей посвящено в мире два-три десятка книг. Для того чтобы создать программу, которая запускается в среде AutoCAD и корректно работает в ней, необходимы не только знания обычного программиста, но и знания опытного пользователя, помноженные на владение графическим редактором и объектной моделью AutoCAD, нюансами интерфейса, контекстами приложения, документа.

Автор написал по этой теме восемь книг, из них семь вышли на русском языке. Настоящая книга создана как обзор систем программирования, применяемых для приложений в среде AutoCAD, и является обновлением книги «AutoCAD: разработка приложений, настройка и адаптация», напечатанной в 2006 г. издательством «БХВ-Петербург». С момента выхода упомянутой книги прошло восемь лет и выпущено девять новых версий AutoCAD, причем версии 2008–2015, помимо 32-разрядных выпусков, имеют еще и 64-разрядные. Каждая версия приносит разработчику определенные проблемы, вызываемые несовместимостью или неполной совместимостью новых версий с приложениями, работавшими в более ранних версиях AutoCAD. Если это для читателя критично, то он должен с особой тщательностью подойти к выбору языка разработки. А иначе с каждой новой версией AutoCAD придется выпускать новую версию своего приложения.

В данной книге упор сделан на три последние версии системы AutoCAD: 2013, 2014 и 2015 (самая новая в этом списке версия 2015 появилась в марте 2014 г.). Книга имеет 6 глав. Практические примеры к книге выложены на сайте издательства «ДМК Пресс».

Глава 1 является обзорной. В ней приведены основные схемы работы приложений в среде AutoCAD и разбирается объектная модель AutoCAD. Здесь же рассмотрены вопросы совместимости приложений.

Центральной можно назвать главу 2, в которой рассмотрен процесс разработки приложений на C++ с помощью библиотек ObjectARX. Это самый мощный инструмент для приложений под AutoCAD. На нем написана большая часть ядра системы.

Следом идет глава 3, которая дополняет материал главы 2 сведениями об интересной теме – написании собственных примитивов для AutoCAD и проектировании их «интеллекта». Это можно сделать только на C++.

В главе 4 изучается процесс написания управляемых приложений с помощью библиотек .NET Framework. Языков, которые используют .NET, много, и все они применимы, но мы ведем изложение на примере языка C#. В этой же главе рассмотрены такие интересные вопросы, как создание, размещение и совместимость плагинов, локальная и удаленная отладка, поддержка единого кода для разных версий.

Глава 5 посвящена сведениям о языках AutoLISP и DCL, а также о среде разработки Visual LISP. Кратко разбираются некоторые приемы написания LISP-приложений.

В главе 6 описана методика разработки приложений на языке Visual Basic for Applications (VBA), с учетом разрядности AutoCAD.

Базовой для описания является русская версия системы AutoCAD, однако часть материала демонстрируется на базе английской версии. То же самое относится к описываемым версиям среды разработки Microsoft Visual Studio.

Автор выражает свою благодарность Андрею Андреевичу Бушману (Санкт-Петербург) за предоставленные материалы (для главы 4), а также ценные советы и замечания.

Дополнительные сведения о других книгах автора можно найти на сайте <http://poleshchuk.spb.ru/cad> (на русском языке), <http://poleshchuk.spb.ru/cad/eng.html> и <http://partnerproducts.autodesk.com/popups/company.asp?rdid=DERU1314> (на английском языке).

Языки программирования и объектная модель

Обзор средств разработки	13
Об объектной модели AutoCAD	14
Особенности программирования для AutoCAD 2015	20
Доверенные каталоги	20
О совместимости	21

Система AutoCAD предоставляет программистам и опытным пользователям обширные возможности создавать свои приложения и строить на базе графического процессора новые системы автоматизированного проектирования. Примерами таких систем могут служить специализированные пакеты, разработанные самой фирмой Autodesk (AutoCAD Mechanical, AutoCAD Architecture, AutoCAD MEP, AutoCAD Civil 3D и т. д.) и другими компаниями.

Данная глава является справочной и содержит краткое описание средств программирования для AutoCAD и информацию об объектной модели AutoCAD.

Обзор средств разработки

Вместе с системой AutoCAD всех версий поставляются среда разработки на языке LISP (Visual LISP). Для AutoCAD 2014–2015 выпущены специальные плагины (32-разрядные и 64-разрядные) со средой разработки на языке Basic (VBA, или Visual Basic for Applications).

Для работы с другими языками обычно требуются приобретение и установка среды разработки Microsoft Visual Studio (версия Visual Studio зависит от версии AutoCAD).

Отдельно от системы AutoCAD распространяется пакет ObjectARX, который содержит необходимые описания и библиотеки для программирования на языке C++ (при этом требуется среда Microsoft Visual Studio). Пакет ObjectARX выпускается под каждую версию AutoCAD. В главе 2 настоящей книги рассмотрены основы разработки приложений с применением ObjectARX (они называются ARX-приложениями). Глава 3 продолжает рассмотрение, делая акцент на DBX-приложениях, которые являются подмножеством ARX-приложений.

.NET Framework за последние годы стала самой популярной платформой, разработанной фирмой Microsoft для пользовательских приложений. На .NET-языках можно писать плагины для AutoCAD в формате управляемых DLL (см. главу 4). Для загрузки таких DLL есть специальная команда NETLOAD.

Сохраняется поддержка в AutoCAD языка AutoLISP (вариант языка LISP) и его расширения – Visual LISP. В главе 5 разбираются вопросы программирования на языке LISP, создания FAS- и VLX-приложений. AutoLISP по-прежнему остается самым простым языком для написания приложений, работающих в AutoCAD. С ним достаточно легко справляются пользователи, не имеющие базовых знаний программирования. Для написания простых диалоговых окон AutoLISP использует специальный язык DCL (Dialog Control Language).

В главе 6 изучаются основы применения в среде AutoCAD языка Basic (VBA) и создания DVB-приложений. VBA в основном работает через объектную модель AutoCAD.

В версии 2014 появилась поддержка языка JavaScript (JS). По смыслу она нужна в первую очередь для облачных версий AutoCAD (AutoCAD WS, AutoCAD 360). Для загрузки JS-приложений используется специальная команда WEBLOAD. Но

пока уровень возможностей JavaScript невысок, он в данной книге не рассматривается.

Пользователь может работать с любыми перечисленными средствами для создания своих приложений, однако в любом случае желательно изучить язык Visual LISP (не обязательно полностью), поскольку в командной строке и в меню системы AutoCAD допускаются только команды с опциями и выражения на языке LISP (они должны начинаться и заканчиваться круглыми скобками). Возможно создание смешанных приложений, части которых созданы в разных языках. Самыми лучшими возможностями обладает язык C++, а если нужна связка языков, то лучше всего себя зарекомендовала пара C++ – LISP.

Об объектной модели AutoCAD

Система AutoCAD разработана с учетом концепции COM (COM – Component Object Model, модель компонентных объектов) фирмы Microsoft, что позволяет обращаться к объектам AutoCAD из других систем и их приложений и, наоборот, к объектам других систем из системы AutoCAD. COM – это спецификация метода создания компонентов, из которых можно строить приложения.

Благодаря COM система AutoCAD поддерживает **технология ActiveX** (точнее, ActiveX Automation), которая реализует идеологию объектно-ориентированного программирования и позволяет оперировать не байтами, числами, а объектами прикладной среды и работать с ними на языке, подобном языку обычных манипуляций. В технологии ActiveX объектная модель работающего приложения представляется как совокупность объектов, коллекций, свойств, методов и событий.

Описание объектной модели для последней установленной на вашем компьютере версии AutoCAD можно найти в файле C:\Program Files\Common Files\Autodesk Shared\acadauto.chm.

Объекты и коллекции

Объекты (objects) ActiveX в системе AutoCAD рассматриваются как иерархия, содержащая не только примитивы, но и прочие элементы (символьные таблицы, словари и т. д.). Однотипные объекты объединяются в коллекции (семейства, collections). Приводимые далее наименования элементов иерархии используются в системах программирования.

Корневым элементом такой иерархии является объект Application (или Acad-Application в VBA), остальные находятся на более низких уровнях. Application является родительским объектом для объектов Preferences (настройки) и VBE (среда VBA-приложений, которая тоже оформлена как объект), а также следующих коллекций:

- Documents (документы);
- MenuBar (строка меню);
- MenuGroups (группы адаптации).

Объект Preferences, в свою очередь, является родительским для объектов различных видов настроек системы AutoCAD.

Коллекция Documents включает объекты типа Document – открытые документы (чертежи). В коллекцию MenuGroups входят объекты типа MenuGroup (группа адаптации), а в коллекцию MenuBar – объекты типа PopupMenu (падающие меню, загруженные в строку меню).

Эта иерархия раскрывается дальше. Объект MenuGroup содержит коллекции PopupMenus (падающие и контекстные меню) и Toolbars (панели инструментов). Затем следуют цепочки:

- PopupMenus – PopupMenu (падающее или контекстное меню) – PopupMenuItem (пункт падающего или контекстного меню);
- Toolbars – Toolbar (панель инструментов) – ToolbarItem (пункт, или кнопка панели инструментов).

Самым сложным и интересным является объект Document (в VBA Acad-Document). Он включает в себя такие коллекции:

- Blocks (блоки);
- ModelSpace (пространство модели);
- PaperSpace (пространство листа);
- Dictionaries (словари);
- DimStyles (размерные стили);
- Groups (группы);
- Layers (слои);
- Layouts (вкладки листов);
- Linetypes (типы линий);
- PlotConfigurations (конфигурации печати);
- RegisteredApplications (зарегистрированные приложения);
- SelectionSets (наборы выбора);
- TextStyles (текстовые стили);
- Viewports (видовые экраны);
- Views (виды).

Кроме того, объект Document порождает такие объекты: Preferences (настройки чертежа), Plot (печать), Utility (утилита), SummaryInfo (свойства чертежа).

В коллекции Blocks, ModelSpace и PaperSpace входят объекты, которые соответствуют графическим примитивам.

В табл. 1.1 приведен перечень объектов и коллекций.

Таблица 1.1. Объекты и коллекции

Имя объекта	Описание
3DFace	3D-грань
3DPolyline	3D-полилиния
3DSolid	Тело
AcadState	Состояние системы AutoCAD как приложения
AcCmColor	Точный цвет
Application	Приложение

Таблица 1.1 (продолжение)

Имя объекта	Описание
Arc	Дуга
Attribute	Описание атрибута
AttributeReference	Атрибут
Block	Описание блока
BlockRef	Вхождение блока
Blocks	Блоки (коллекция)
Circle	Круг
Database	Графическая база рисунка
DatabasePreferences	Настройки чертежа
DgnUnderlay	DGN-подложка
Dictionaries	Словари (коллекция)
Dictionary	Словарь
Dim3PointAngular	Угловой размер по трем точкам
DimAligned	Выровненный размер
DimAngular	Угловой размер
DimArcLength	Размер длины дуги
DimDiametric	Диаметральный размер
DimOrdinate	Ординатный размер
DimRadial	Радиальный размер
DimRadialLarge	Радиальный размер со сломом размерной линии
DimRotated	Повернутый размер
DimStyle	Размерный стиль
DimStyles	Размерные стили (коллекция)
Document	Документ (чертеж)
Documents	Документы (коллекция)
DwfUnderlay	DWF-подложка
DynamicBlockReferenceProperty	Свойство вхождения динамического блока
Ellipse	Эллипс
ExternalReference	Внешняя ссылка
ExtrudedSurface	Поверхность, построенная методом выдавливания
FileDependencies	Внешние зависимости (коллекция)
FileDependency	Элемент из списка файловых зависимостей
Group	Группа
Groups	Группы (коллекция)
Hatch	Штриховка
Helix	Спираль
Hyperlink	Гиперссылка
Hyperlinks	Гиперссылки (коллекция)
IDPair	Специальный объект метода CopyObjects
Layer	Слой
Layers	Слои (коллекция)
LayerStateManager	Диспетчер свойств слоев
Layout	Вкладка листа
Layouts	Вкладки (коллекция)

Таблица 1.1 (продолжение)

Имя объекта	Описание
Leader	Выноска
LightweightPolyline	Компактная двумерная полилиния
Line	Отрезок
Linetype	Тип линий
Linetypes	Типы линий (коллекция)
LoftedSurface	Поверхность, созданная методом лфтинга
Material	Материал
Materials	Материалы (коллекция)
MenuBar	Строка меню (коллекция)
MenuGroup	Группа адаптации
MenuGroups	Группы адаптации (коллекция)
MinsertBlock	Массив блоков
Mline	Мультилиния
ModelSpace	Пространство модели (коллекция)
MText	Мультитекст
Ole	Контейнер OLE-объекта
PaperSpace	Пространство листа (коллекция)
PlaneSurface	Плоская поверхность
Plot	Набор свойств печати листа
PlotConfiguration	Настройка печати
PlotConfigurations	Настройки печати (коллекция)
Point	Точка
PolyfaceMesh	Многогранная сеть (сеть из полиграней)
PolygonMesh	Полигональная сеть
Polyline	Подробная двумерная полилиния
PopupMenu	Падающее или контекстное меню
PopupMenuItem	Пункт падающего или контекстного меню
PopUpMenus	Падающие и контекстные меню (коллекция)
Preferences	Настройки системы AutoCAD
PreferencesDisplay	Экранные настройки
PreferencesDrafting	Настройки рисования
PreferencesFiles	Файловые настройки
PreferencesOpenSave	Настройки открытия и сохранения
PreferencesOutput	Настройки вывода
PreferencesProfiles	Настройки профилей
PreferencesSelection	Настройки выбора
PreferencesSystem	Системные настройки
PreferencesUser	Пользовательские настройки
PViewport	Видовой экран пространства листа
Raster	Растровое изображение
Ray	Луч
Region	Область
RegisteredApplication	Имя зарегистрированного приложения

Таблица 1.1 (окончание)

Имя объекта	Описание
RegisteredApplications	Имена зарегистрированных приложений (коллекция)
RevolvedSurface	Поверхность, созданная методом вращения
Section	Секущая плоскость
SectionManager	Менеджер коллекции секущих плоскостей
SectionSettings	Настройки секущей плоскости
SectionTypeSettings	Настройки элементов динамического сечения
SecurityParams	Настройки безопасности
SelectionSet	Набор выбора
SelectionSets	Наборы (коллекция)
ShadowDisplay	Свойства теней
Shape	Форма
Solid	Фигура
SortentsTable	Таблица управления порядком прорисовки
Spline	Сплайн
SubEntity	Подобъект
SubEntSolidEdge	Кромка тела
SubEntSolidNode	Узел тела
SubEntSolidVertex	Вершина тела
SummaryInfo	Свойства чертежа (автор, заголовок и т. п.)
Surface	Поверхность
SweptSurface	Поверхность, построенная методом сдвига
Table	Таблица
TableStyle	Стиль таблиц
TableStyles	Стили таблиц (коллекция)
Text	Текст (однострочный)
TextStyle	Текстовый стиль
TextStyles	Текстовые стили (коллекция)
Tolerance	Допуск
Toolbar	Панель инструментов
ToolbarItem	Элемент панели инструментов
Toolbars	Панели инструментов (коллекция)
Trace	Полоса
UCS	ПСК
UCSs	Коллекция ПСК
Utility	Утилита
VBE	Среда VBA
View	Вид
Viewport	Видовой экран
Viewports	Видовые экраны (коллекция)
Views	Виды (коллекция)
Wipeout	Маска
XLine	Прямая
XRecord	X-запись

В таблицу не внесены объекты и семейства, связанные с подшивками и операциями связи с базами данных.

Работа с объектами в технологии ActiveX, как правило, ведется на уровне указателей, которые указывают на начало структур данных этих объектов.

Интерфейсы

Для реализации базовых инструментов объектной технологии в модели предусмотрены компоненты, называемые интерфейсами. Каждый класс реализуется с помощью своего интерфейса, являющегося потомком соответствующего базового интерфейса. В базовом интерфейсе описываются все типичные свойства и методы. Например, объект Line, который является представителем класса AcDbLine, создан с помощью интерфейса IAcadLine, который, в свою очередь, является потомком интерфейса IAcadEntity – интерфейса всех графических примитивов.

Свойства

Каждый объект в иерархии, используемой технологией ActiveX, имеет свои данные, или **свойства** (properties), которые можно читать или изменять с помощью соответствующих функций.

Коллекции тоже обладают свойствами. Такие компоненты объектной модели, как интерфейсы, не все обладают свойствами.

Методы

Каждый объект в технологии ActiveX обычно имеет специфические функции, или **методы** (methods), которые можно применять, совершая операции над объектами. Практически любая команда системы AutoCAD соответствует некоторому методу.

К коллекциям и интерфейсам применимы их собственные методы.

События

Программа может отслеживать **события** (events), о которых сообщает система AutoCAD. Они возникают в результате действий пользователя, и программа может реагировать на них. События тоже являются элементами объектной модели.

Простой пример использования объектной модели

Все разбираемые в следующих главах системы программирования дают возможность использовать рассмотренную объектную модель с помощью COM (ActiveX). Приведем три варианта примера (три фрагмента программ), как прочесть указатель объекта падающего меню, находящегося в строке меню на позиции с заданным номером.

В Visual LISP:

```
(vla-get-item (vla-get-menubar (vlax-get-acad-object)) i)
```

В Visual Basic for Applications:

```
ThisDrawing.Application.MenuBar.Item (i);
```

Здесь ThisDrawing – стандартный указатель активного документа.

В ObjectARX (C++):

```
pPopupMenu = pMenuBar->Item(_variant_t(i));
```

Особенности программирования для AutoCAD 2015

В системе AutoCAD 2015 прекращена поддержка fibers (волокон потоков Windows). Это задает дополнительную работу для программистов, переносящих свои приложения с версии 2014 (или более ранней) на версию 2015.

Изменения коснулись функций, выполняющих команды AutoCAD и имитирующих ввод в командной строке. Теперь стало два типа инструментов разработчика: для синхронного и для асинхронного выполнения команды. Под синхронным выполнением имеется в виду вызов команды с полным и заранее определенным набором параметров (ответов пользователя) вплоть до завершения команды. Остальные случаи следует рассматривать как асинхронные. В частности, к асинхронным следует отнести варианты с паузой для ожидания дополнительного ввода пользователя.

В C++ вызовы функций `acedCommand` и `acedCmd` следует заменить на `acedCommandS` и `acedCmdS` (синхронные) или на `acedCommandC` и `acedCmdC` (асинхронные).

В C# в классе `Autodesk.AutoCAD.EditorInput.Editor` появились методы `Command` (синхронное выполнение) и `CommandAsync` (асинхронное выполнение). Это распространяется на все языки на платформе .NET.

Изменения коснулись такого статичного языка, как AutoLISP. Теперь функция `command` используется для асинхронного выполнения команд. А для синхронного используется новая функция `command-s`.

См. также соответствующие разделы в главах, посвященных языкам C++, C# и AutoLISP.

Доверенные каталоги

Начиная с версии 2014, разработчики AutoCAD повысили уровень защиты от потенциально опасных приложений. Если пользователь пытается загрузить исполняемый файл из несанкционированного источника, то система AutoCAD выводит предупредительное сообщение (рис. 1.1).

Пользователь должен ответить, готов ли он загрузить указанный в предупреждении файл.

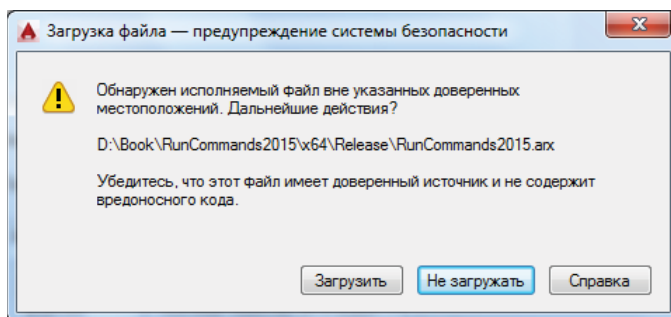


Рис. 1.1. Предупреждение системы безопасности

Уровень безопасности при загрузке контролируется значением системной переменной `SECURELOAD`, которая хранится в реестре Windows. Она может принимать целочисленные значения, от 0 до 2 (по умолчанию – 1):

- 0 – загружаются любые файлы;
- 1 – загружаются файлы из доверенных источников, а при попытке загрузки из других мест выводится предупреждение (см. рис. 1.1);
- 2 – загружаются файлы только из доверенных источников, остальные игнорируются.

Перечень доверенных папок хранится в системной переменной `TRUSTED-PATHS`. Стандартные папки установки системы AutoCAD и папка размещения плагинов (`%ProgramData%\Autodesk\ApplicationPlugins`) автоматически считаются доверенными. Если помимо указанной папки (например, «D:\RSHIP») необходимо, чтобы доверенными были и все вложенные в нее подпапки, то после полного имени папки следует вставить «\...» (например, «D:\RSHIP\...»).

Изменять список доверенных каталогов можно также через параметр **Доверенные местоположения** (Trusted Locations) вкладки **Файлы** (Files) диалогового окна **Параметры** (Options) (рис. 1.2).

На рис. 1.2 задан путь к доверенному местоположению приложений (а также к вложенным папкам).

Поскольку программы на языке JavaScript должны в первую очередь выполняться прямо со страниц Интернета, то список доверенных доменов необходимо перечислить в системной переменной `TRUSTEDDOMAINS`. Страницы типа `*.autodesk.com`, `*.autocad360.com` считаются заведомо доверенными.

В AutoCAD 2013 SP 1.1 появились системные переменные `AUTOLOADPATH` (хранит пути, разрешенные для автозагрузки), `AUTOLOAD` (может запретить автоматическую загрузку).

О СОВМЕСТИМОСТИ

Разрабатывая приложение под какую-то версию AutoCAD определенной разрядности, автор хотел бы знать, сможет ли его приложение без переделки пра-

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru