



CATIA V5 制图与二次开发教程

CATIA V5 基础操作

1. 启动与界面介绍

启动 CATIA V5 通常通过双击桌面图标或从开始菜单中选择 CATIA V5。首次启动后，用户将看到 CATIA 的主界面，它由以下几个主要部分组成：

- 菜单栏：包含各种功能菜单，如文件、编辑、视图等。
- 工具栏：提供快速访问常用工具的按钮。
- 工作台：CATIA V5 的核心功能区，用户可以在此进行设计、分析、制图等工作。
- 图形窗口：显示 3D 模型或 2D 图纸的区域。
- 属性窗口：显示和编辑当前选择对象的属性。
- 任务窗格：提供特定任务的指导和设置，如创建零件、装配等。

2. 基本绘图工具使用

CATIA V5 提供了丰富的绘图工具，用于创建和编辑 2D 图纸。以下是一些基本工具的使用方法：

- 直线工具：用于绘制直线。选择工具后，在图形窗口中点击并拖动鼠标即可绘制。
- 圆工具：用于绘制圆。选择工具后，指定圆心和半径即可完成绘制。
- 矩形工具：用于绘制矩形。选择工具后，指定两个对角点即可完成绘制。
- 尺寸标注：用于添加尺寸信息到图纸上。选择尺寸标注工具，然后选择要标注的对象，输入尺寸值，即可完成尺寸标注。

2.1 示例：使用 VB 进行直线绘制

```
' 引用 CATIA V5 对象库
Dim oApp As Object
Set oApp = CreateObject("CATIA.Application")

' 创建新的零件文档
Dim oDocument As Object
Set oDocument = oApp.Documents.Add("Part")

' 获取零件设计工作台
Dim oPartDesign As Object
Set oPartDesign = oDocument.Part.PartDesign

' 创建新的草图
Dim oSketch As Object
```

```
Set oSketch = oPartDesign.HybridBodies.Item("Geometrical
Set.1").HybridBodySequences.Item(1).HybridBodySketches.Add(oPartDesign.Hybr
Set.1").HybridBodySequences.Item(1).HybridBodySketches.Item(1).Plane)
```

' 定义直线的起点和终点

```
Dim oPoint1 As Object
Set oPoint1 = oSketch.CreatePoint(0, 0, 0)
Dim oPoint2 As Object
Set oPoint2 = oSketch.CreatePoint(100, 0, 0)
```

' 创建直线

```
Dim oLine As Object
Set oLine = oSketch.CreateLine(oPoint1, oPoint2)
```

' 更新文档

```
oDocument.Update
```

3. 零件设计流程

零件设计在CATIA V5中遵循以下基本流程：

1. 创建新零件文档：从文件菜单中选择新建，然后选择零件类型。
2. 选择工作平面：在设计零件前，需要选择一个工作平面作为草图的基准。
3. 绘制草图：使用绘图工具在工作平面上绘制草图。
4. 添加特征：基于草图添加特征，如拉伸、旋转、倒角等。
5. 编辑特征：对已添加的特征进行编辑，调整尺寸、位置等。
6. 检查零件：使用分析工具检查零件的几何和物理属性。
7. 保存零件：完成设计后，保存零件文档。

4. 装配体创建与编辑

装配体设计是CATIA V5中的另一重要功能，它允许用户将多个零件组合成一个整体。装配体的创建和编辑流程如下：

1. 创建新装配体文档：从文件菜单中选择新建，然后选择装配体类型。
2. 导入零件：使用导入功能将设计好的零件导入到装配体文档中。
3. 定义约束：为零件之间的相对位置定义约束，如接触、对齐、距离等。
4. 编辑装配：调整零件的位置，添加或删除约束，以达到设计要求。
5. 检查装配体：使用分析工具检查装配体的干涉和运动。
6. 保存装配体：完成装配后，保存装配体文档。

4.1 示例：使用VB导入零件到装配体

```
' 引用CATIA V5对象库
Dim oApp As Object
Set oApp = CreateObject("CATIA.Application")

' 创建新的装配体文档
Dim oDocument As Object
Set oDocument = oApp.Documents.Add("Assembly")

' 获取装配体工作台
Dim oAssembly As Object
Set oAssembly = oDocument.Assembly

' 定义零件路径
Dim sPartPath As String
sPartPath = "C:\Path\To\Your\Part.CATPart"

' 导入零件
Dim oProduct As Object
Set oProduct = oAssembly.Products.Add(sPartPath)

' 更新文档
oDocument.Update
```

以上教程详细介绍了CATIA V5的基础操作，包括启动与界面介绍、基本绘图工具使用、零件设计流程以及装配体创建与编辑。通过VB脚本示例，展示了如何在CATIA V5中自动化直线绘制和零件导入到装配体的过程，为用户提供了实际操作的参考。

CATIAV5VB二次开发入门

5. VB编程环境搭建

在开始CATIA V5的VB二次开发之前，首先需要搭建一个合适的编程环境。这通常涉及到以下步骤：

1. **安装Microsoft Visual Basic 6.0**：虽然CATIA V5支持多种编程语言，但VB因其易学性和与CATIA的紧密集成而被广泛使用。确保你的系统中安装了Visual Basic 6.0。
2. **配置CATIA V5**：在CATIA V5中，需要启用VBScript支持。这可以通过CATIA的“选项”菜单下的“宏”设置来完成，确保“启用VBScript”选项被勾选。
3. **添加CATIA V5对象库引用**：在Visual Basic的“工程”菜单中，选择“引用”，然后添加

CATIA V5的Type Library。这一步是关键，因为它允许VB识别CATIA的对象和方法。

4. **创建工程**：在Visual Basic中创建一个新的工程，这将成为你编写和测试CATIA V5二次开发脚本的环境。

6. CATIAV5API简介

CATIA V5的API（Application Programming Interface）是一套允许开发者通过编程语言控制CATIA功能的工具。它包括了大量类和方法，覆盖了从基本的文件操作到复杂的零件设计和装配管理。

6.1 基本概念

- **Application**：代表CATIA V5应用程序的主对象，通过它可以访问所有其他对象。
- **Document**：代表CATIA中的一个文档，可以是零件、装配或图纸。
- **Part**：代表一个零件，提供了创建和编辑零件几何体的方法。
- **Product**：代表一个装配，提供了管理装配结构和零件关系的方法。
- **Shape**：代表一个几何形状，如面、线、点等，提供了对这些形状进行操作的方法。

6.2 示例代码

下面是一个简单的VB脚本示例，用于创建一个新的CATIA V5零件文档：

```
' 声明CATIA Application对象
Dim oApp As Object
Set oApp = CreateObject("CATIA.Application")

' 显示CATIA窗口
oApp.Visible = True

' 创建一个新的零件文档
Dim oDoc As Object
Set oDoc = oApp.Documents.Add("Part")

' 创建一个立方体
Dim oPart As Object
Set oPart = oDoc.Part
Dim oHybridShapeFactory As Object
Set oHybridShapeFactory = oPart.HybridShapeFactory
Dim oBox As Object
Set oBox = oHybridShapeFactory.AddNewBox(10, 10, 10)
oPart.Bodies.Item(1).AddHybridShape oBox

' 保存文档
```

```
oDoc.SaveAs "C:\MyCube.CATPart"
```

```
' 清理对象
```

```
Set oBox = Nothing  
Set oHybridShapeFactory = Nothing  
Set oPart = Nothing  
Set oDoc = Nothing  
Set oApp = Nothing
```

这段代码首先创建了一个CATIA Application对象，并使其可见。然后，它添加了一个新的零件文档，并在该文档中创建了一个10x10x10的立方体。最后，它将文档保存到指定的路径，并清理了所有使用的对象。

7. 第一个CATIA V5 VB脚本

编写你的第一个CATIA V5 VB脚本是一个激动人心的时刻，它标志着你开始掌握如何通过编程控制CATIA。以下是一个简单的脚本，用于在CATIA中创建一个圆：

```
' 声明CATIA Application对象
```

```
Dim oApp As Object  
Set oApp = CreateObject("CATIA.Application")
```

```
' 显示CATIA窗口
```

```
oApp.Visible = True
```

```
' 创建一个新的零件文档
```

```
Dim oDoc As Object  
Set oDoc = oApp.Documents.Add("Part")
```

```
' 创建一个圆
```

```
Dim oPart As Object  
Set oPart = oDoc.Part  
Dim oHybridShapeFactory As Object  
Set oHybridShapeFactory = oPart.HybridShapeFactory  
Dim oCircle As Object  
Set oCircle = oHybridShapeFactory.AddNewCircle(0, 0, 0, 0, 0, 1, 10)  
oPart.Bodies.Item(1).AddHybridShape oCircle
```

```
' 保存文档
```

```
oDoc.SaveAs "C:\MyCircle.CATPart"
```

```
' 清理对象
```

```
Set oCircle = Nothing  
Set oHybridShapeFactory = Nothing
```

```
Set oPart = Nothing
Set oDoc = Nothing
Set oApp = Nothing
```

在这个脚本中，我们使用了AddNewCircle方法来创建一个半径为10的圆，其位置在原点，且垂直于Z轴。

8. 错误处理与调试

在开发CATIA V5的VB脚本时，错误处理和调试是必不可少的步骤，以确保脚本的稳定性和可靠性。

8.1 错误处理

使用On Error GoTo语句可以捕获和处理运行时错误。例如：

```
On Error GoTo ErrorHandler

' 创建CATIA Application对象
Dim oApp As Object
Set oApp = CreateObject("CATIA.Application")

' 如果出错，跳转到错误处理部分
ErrorHandler:
If Err.Number <> 0 Then
    MsgBox "An error occurred: " & Err.Description
    Err.Clear
End If
```

8.2 调试技巧

- 使用断点：在代码中设置断点，然后逐行执行脚本，观察变量的值和程序的执行流程。
- 输出调试信息：使用Debug.Print或MsgBox来输出变量的值或程序状态，帮助理解代码的执行情况。
- 检查对象属性和方法：在脚本中使用对象时，确保你了解该对象的所有属性和方法，这可以通过CATIA的帮助文档或对象浏览器来实现。

通过这些步骤，你可以有效地处理和调试CATIA V5 VB脚本中的错误，确保你的二次开发项目顺利进行。

零件设计自动化

9. 参数化设计实现

参数化设计是CATIA V5 VB二次开发中的核心概念之一，它允许设计者通过定义参数和约束来控制零件的几何形状和尺寸。这种设计方式极大地提高了设计的灵活性和效率，因为一旦参数或约束发生变化，整个零件模型会自动更新，无需手动调整。

9.1 原理

参数化设计基于参数和约束的关联性。在CATIA中，每个零件的几何特征和尺寸都可以被定义为参数，这些参数可以是数值、表达式或与其他参数的关联。约束则定义了零件中不同特征之间的关系，如平行、垂直、相切等。通过VB脚本，可以创建和修改这些参数和约束，从而实现零件设计的自动化。

9.2 示例代码

以下是一个使用VBScript在CATIA V5中创建参数化圆柱体的示例：

```
' 创建新的零件文档
Dim aProductDocument1 As ProductDocument1
Set aProductDocument1 = CATIA.ActiveDocument

' 获取零件
Dim aPart As Part
Set aPart = aProductDocument1.Part

' 创建工作平面
Dim aWorkPlane As WorkPlane
Set aWorkPlane = aPart.WorkPlanes.CreateByOffset(aPart.MainWorkPlane,
    0, CATBaseKinds.catPlaneKind)

' 创建参数
Dim aParameter As Parameter
Set aParameter = aPart.Parameters.Create("CylinderRadius", 10, "mm",
    "圆柱半径")

' 创建圆柱体
Dim aCylinder As Body
Set aCylinder = aPart.Bodies.CreateCylinder(aWorkPlane,
    aParameter.Value, 50, "mm", True)

' 修改参数
```

```
aParameter.Value = 15 ' 改变圆柱半径为15mm
```

9.3 描述

此代码首先创建一个新的零件文档，并在其中定义一个工作平面。接着，通过 `aPart.Parameters.Create` 方法创建一个名为 `CylinderRadius` 的参数，其初始值为 `10mm`。然后，使用这个参数的值来创建一个圆柱体。最后，通过修改参数的值，可以自动更新圆柱体的尺寸，无需重新创建圆柱体。

10. 特征创建与编辑自动化

在CATIA V5中，特征是指构成零件几何形状的基本元素，如孔、凸台、凹槽等。通过VB脚本，可以自动化创建和编辑这些特征，从而节省设计时间并减少错误。

10.1 原理

特征创建与编辑自动化依赖于CATIA的API，这些API提供了创建和修改特征的方法。例如，可以使用 `Bodies.CreateCylinder`、`Bodies.CreateHole` 等方法来创建圆柱体和孔。编辑特征则可以通过访问和修改特征的属性来实现。

10.2 示例代码

下面的代码示例展示了如何在CATIA V5中自动化创建一个孔，并修改其尺寸：

```
' 创建孔特征
Dim aHole As Body
Set aHole = aPart.Bodies.CreateHole(aWorkPlane, 20, "mm", 50, "mm",
    0, 0, 0, True)

' 修改孔的直径
Dim aHoleDiameter As Parameter
Set aHoleDiameter = aHole.Parameters.Item("Diameter")
aHoleDiameter.Value = 25 ' 改变孔直径为25mm
```

10.3 描述

这段代码首先使用 `Bodies.CreateHole` 方法在工作平面上创建一个直径为 `20mm`、深度为 `50mm` 的孔。然后，通过访问孔特征的参数，找到名为 `Diameter` 的参数，并将其值修改为 `25mm`，从而自动更新孔的尺寸。

11. 零件库的创建与管理

零件库是CATIA V5中用于存储和管理标准零件和组件的工具。通过VB脚本，可以自动化创建和更新零件库，提高设计的标准化和一致性。

11.1 原理

零件库的创建与管理主要涉及零件的分类、存储和检索。在CATIA中，可以使用VB脚本来创建新的零件库、添加零件到库中、修改零件属性以及检索和使用库中的零件。这通常涉及到对CATIA的Product和ProductDocument对象的操作。

11.2 示例代码

以下代码示例展示了如何在CATIA V5中自动化创建一个零件，并将其添加到零件库中：

```
' 创建新的零件文档
Dim aProductDocument1 As ProductDocument1
Set aProductDocument1 = CATIA.Documents.Add("Part")

' 获取零件
Dim aPart As Part
Set aPart = aProductDocument1.Part

' 创建圆柱体
Dim aCylinder As Body
Set aCylinder = aPart.Bodies.CreateCylinder(aPart.MainWorkPlane, 10,
    "mm", 50, "mm", True)

' 保存零件
aProductDocument1.SaveAs "C:\PartsLibrary\MyCylinder.CATPart", 1

' 将零件添加到库中
Dim aProduct As Product
Set aProduct = CATIA.ActiveDocument.Product
aProduct.Products.AddFromFile "C:\PartsLibrary\MyCylinder.CATPart"
```

11.3 描述

这段代码首先创建一个新的零件文档，并在其中创建一个圆柱体。然后，将这个零件保存到指定的路径下，作为零件库的一部分。最后，通过aProduct.Products.AddFromFile方法，将保存的零件添加到当前的零件库中，使得在未来的装配设计中可以方便地检索和使用这个零件。

通过上述示例，我们可以看到CATIA V5 VB二次开发在零件设计自动化中的强大功能，它不仅能够创建和编辑零件特征，还能管理零件库，极大地提高了设计效率和标准化水平。

装配自动化与宏命令

12. 装配序列自动化

在CATIA V5的二次开发中，装配序列自动化是提高设计效率的关键技术之一。通过VB宏命令，可以实现装配体的自动创建、零件的自动定位与约束、以及装配顺序的自动优化。这一过程不仅减少了人工操作的错误，还大大缩短了产品设计周期。

12.1 原理

装配序列自动化基于CATIA V5的装配设计模块，利用VB宏命令对装配体的创建过程进行编程控制。宏命令可以调用CATIA V5的API函数，实现对装配体的参数化控制，包括零件的添加、位置的调整、约束的设定等。

12.2 内容

1. 零件自动添加：通过宏命令，可以自动将零件添加到装配体中，避免手动操作。
2. 自动定位与约束：宏命令可以自动设定零件之间的定位和约束关系，确保装配体的正确性。
3. 装配顺序优化：宏命令可以分析装配体的结构，自动优化装配顺序，提高装配效率。

12.3 示例代码

```
Sub AutoAssemble()  
    Dim CATIA As Object  
    Dim ProductDocument1 As Object  
    Dim Product1 As Object  
    Dim Part1 As Object  
    Dim Part2 As Object  
    Dim ProductStructure1 As Object  
    Dim ProductStructureNode1 As Object  
    Dim ProductStructureNode2 As Object  
    Dim ProductStructureNode3 As Object  
    Dim ProductStructureNode4 As Object  
    Dim ProductStructureNode5 As Object  
    Dim ProductStructureNode6 As Object  
    Dim ProductStructureNode7 As Object  
    Dim ProductStructureNode8 As Object  
    Dim ProductStructureNode9 As Object  
    Dim ProductStructureNode10 As Object  
    Dim ProductStructureNode11 As Object  
    Dim ProductStructureNode12 As Object  
    Dim ProductStructureNode13 As Object  
    Dim ProductStructureNode14 As Object
```

```
Dim ProductStructureNode15 As Object
Dim ProductStructureNode16 As Object
Dim ProductStructureNode17 As Object
Dim ProductStructureNode18 As Object
Dim ProductStructureNode19 As Object
Dim ProductStructureNode20 As Object
Dim ProductStructureNode21 As Object
Dim ProductStructureNode22 As Object
Dim ProductStructureNode23 As Object
Dim ProductStructureNode24 As Object
Dim ProductStructureNode25 As Object
Dim ProductStructureNode26 As Object
Dim ProductStructureNode27 As Object
Dim ProductStructureNode28 As Object
Dim ProductStructureNode29 As Object
Dim ProductStructureNode30 As Object
Dim ProductStructureNode31 As Object
Dim ProductStructureNode32 As Object
Dim ProductStructureNode33 As Object
Dim ProductStructureNode34 As Object
Dim ProductStructureNode35 As Object
Dim ProductStructureNode36 As Object
Dim ProductStructureNode37 As Object
Dim ProductStructureNode38 As Object
Dim ProductStructureNode39 As Object
Dim ProductStructureNode40 As Object
Dim ProductStructureNode41 As Object
Dim ProductStructureNode42 As Object
Dim ProductStructureNode43 As Object
Dim ProductStructureNode44 As Object
Dim ProductStructureNode45 As Object
Dim ProductStructureNode46 As Object
Dim ProductStructureNode47 As Object
Dim ProductStructureNode48 As Object
Dim ProductStructureNode49 As Object
Dim ProductStructureNode50 As Object
```

' 创建CATIA对象

```
Set CATIA = CreateObject("CATIA.Application")
Set ProductDocument1 = CATIA.Documents.Add("Product")
Set Product1 = ProductDocument1.Product
```

' 加载零件

```
Set Part1 = CATIA.Documents.Open("C:\Part1.CATPart")
Set Part2 = CATIA.Documents.Open("C:\Part2.CATPart")
```

```

' 将零件添加到装配体中
Set ProductStructure1 = Product1.ProductStructure
Set ProductStructureNode1 =
ProductStructure1.CreateProductNode (Part1.Product)
Set ProductStructureNode2 =
ProductStructure1.CreateProductNode (Part2.Product)

' 设置零件之间的约束
ProductStructureNode1.SetProductRelation "Constraint1",
"Part1_Face1", "Part2_Face1", "Coincident"

' 保存并关闭文档
ProductDocument1.Save
ProductDocument1.Close
End Sub

```

13. 宏命令的创建与应用

宏命令是CATIA V5二次开发中的一种重要工具，它允许用户记录一系列操作，然后通过VB脚本进行调用和自动化执行。宏命令可以用于重复性高的任务，如零件的批量修改、装配体的创建、图纸的生成等。

13.1 原理

宏命令本质上是CATIA V5操作的脚本记录，通过VB宏命令，可以将这些记录转换为可编程的脚本，实现自动化操作。宏命令的创建通常包括记录操作、编辑宏脚本、保存宏命令等步骤。

13.2 内容

1. 宏命令的记录：在CATIA V5中，可以使用宏记录器记录用户操作。
2. 宏命令的编辑：记录的宏命令需要通过VB编辑器进行编辑，以适应自动化需求。
3. 宏命令的调用：编辑后的宏命令可以通过VB脚本进行调用，实现自动化操作。

13.3 示例代码

```

Sub CreateMacro ()
' 开始记录宏
CATIA.MacroRecorder.Start "MyMacro", "MyMacro Description"

' 执行操作
CATIA.ActiveDocument.Part.CreatePad
CATIA.ActiveDocument.Part.Bodies.Item(1).Faces.Item(1), 10, 0, 0

```

```
' 停止记录宏
```

```
CATIA.MacroRecorder.Stop
```

```
' 保存宏
```

```
CATIA.MacroRecorder.Save "C:\MyMacro.vbs"
```

```
End Sub
```

14. 自动化装配检查

在装配设计中，自动化装配检查可以确保装配体的正确性和可行性。通过VB宏命令，可以自动执行装配体的干涉检查、间隙检查、以及装配顺序的合理性检查。

14.1 原理

自动化装配检查基于CATIA V5的装配分析模块，利用VB宏命令调用相应的分析函数，对装配体进行自动检查。这些检查包括零件之间的干涉分析、间隙分析、以及装配顺序的合理性分析。

14.2 内容

1. 干涉检查：自动检测装配体中零件之间的干涉情况。
2. 间隙检查：自动检测装配体中零件之间的间隙大小，确保装配的可行性。
3. 装配顺序合理性检查：自动分析装配体的装配顺序，确保装配过程的合理性。

14.3 示例代码

```
Sub AutoCheckAssembly()  
    Dim CATIA As Object  
    Dim ProductDocument1 As Object  
    Dim Product1 As Object  
    Dim Analysis1 As Object  
    Dim InterferenceAnalysis1 As Object  
  
    ' 创建CATIA对象  
    Set CATIA = CreateObject("CATIA.Application")  
    Set ProductDocument1 =  
CATIA.Documents.Open("C:\Assembly.CATProduct")  
    Set Product1 = ProductDocument1.Product  
  
    ' 创建干涉分析  
    Set Analysis1 = Product1.Analyses.Add("InterferenceAnalysis")  
    Set InterferenceAnalysis1 = Analysis1.Item(1)  
  
    ' 执行干涉分析  
    InterferenceAnalysis1.Execute
```

```
' 输出分析结果
MsgBox InterferenceAnalysis1.Results.Count & " Interferences
found."

' 保存并关闭文档
ProductDocument1.Save
ProductDocument1.Close
End Sub
```

以上示例展示了如何使用VB宏命令在CATIA V5中实现装配序列自动化、创建与应用宏命令、以及自动化装配检查。通过这些技术，可以显著提高设计效率和装配体的准确性。

图纸生成与定制

15. 图纸模板的创建

在CATIA V5的二次开发中，创建图纸模板是实现自动化制图和定制化标注的基础步骤。通过VB（Visual Basic）编程，可以精确控制图纸的布局、尺寸标注样式、标题栏信息等，从而提高图纸制作的效率和一致性。

15.1 原理

CATIA V5提供了丰富的API（Application Programming Interface）来访问和控制其内部功能，包括图纸模板的创建。这些API允许开发者通过编程方式设置图纸的属性，如纸张大小、比例、图层、标注样式等。

15.2 内容

1. 选择纸张大小和方向：在创建模板时，首先需要选择合适的纸张大小和方向，如A4、A3等，以及是横向还是纵向。
2. 设置比例：根据需要，可以设置图纸的整体比例，以适应不同的设计需求。
3. 定义图层：图层管理是图纸组织的关键，可以定义多个图层，每个图层用于不同的绘图目的，如轮廓线、中心线、尺寸标注等。
4. 创建标题栏：标题栏是图纸的重要组成部分，用于记录设计者、日期、零件名称等信息。通过编程，可以自定义标题栏的样式和内容。
5. 设置标注样式：尺寸标注的样式直接影响图纸的可读性和专业性。可以设置标注的字体、大小、精度等参数。

15.3 示例代码

```
' 创建图纸模板的示例代码
Sub CreateDrawingTemplate()
    Dim aSession As Session
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/076113002114010201>