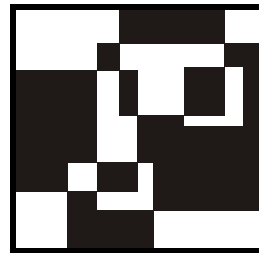


ГЛАВА 15



Программирование в Project

15.1. Основы программирования в Project Professional. Объект *Application*

Еще одно приложение в семействе Office — это Project. Он часто используется на больших предприятиях. Как правило, если на предприятии применяется Project (чаще всего вместе с Project Server), то значит, данных много и, скорее всего, для выполнения различных операций потребуется автоматизация.

Самый главный объект в Project Professional 2003, как и во всех других приложениях Office, — это объект *Application* со стандартным набором свойств и методов. Создать объект *Application* — значит запустить Project Professional:

```
Dim pj As New MSProject.Application  
pj.Visible = True  
MsgBox pj.Name & " " & pj.Version
```

Если запуск производится из другого приложения Office, то вам потребуется добавить в проект ссылку на библиотеку Microsoft Project 11.0 Object Library (меню **Tools | References**).

Если мы выполняем код из самого Project, то объект *Application* будет доступен по умолчанию. Если мы не укажем, из какого именно объекта используется свойство или метод, то компилятор VBA будет считать, что это свойство или метод объекта *Application*. Например, при вызове из VBA в Project следующие строки кода идентичны:

```
Application.ActiveProject.Visible = True  
и  
ActiveProject.Visible = True
```

В Project Professional есть замечательное средство, которого очень не хватает в Word, Excel, PowerPoint и в других приложениях Office. Это средство — глобальный корпоративный шаблон. Он единый для всего сервера Project Server, автоматически загружается при подключении пользователя к серверу и модули из него доступны всем пользователям Project Server. Фактически это корпоративное хранилище кода VBA. Конечно, этот шаблон можно использовать не только для действий, непосредственно связанных с Project, но и для других целей, например, для создания документов Word по единой форме. Немного развив эту тему, можно прийти к системе корпоративного репортинга с централизованным хранилищем шаблонов отчетов.

Однако при этом нужно быть очень внимательным: если у вас есть модуль **Module1** в глобальном корпоративном шаблоне и модуль с таким же именем в проекте, то модуль проекта со всем кодом будет заменен модулем из шаблона.

15.2. Коллекция *Projects*, объект *Project* и вложенные объекты

После того, как объект `Application` создан (а значит, открыт Project, программно или вручную), нам нужно создать или открыть в нем проект.

Создание проекта может выглядеть очень просто, при помощи метода `Add()` коллекции `Projects`:

```
Dim oProj As Project
Set oProj = Projects.Add()
```

Дополнительные необязательные параметры метода `Add()` позволяют создать новый проект на основе шаблона или открыть окно для выбора шаблонов.

Если нужно открыть существующий проект, то нам может встретиться два варианта этой задачи:

1. Нужно открыть локальный проект (из файла `mpp`).

Здесь все просто:

```
Application.FileOpen "D:\Project1.mpp"
```

А затем ловим объектную ссылку на этот проект (поскольку сам метод `FileOpen()` ее не возвращает):

```
For Each Project In Application.Projects
    If Project.Name = "Project1.mpp" Then
        Set oProj = Project
        Exit For
    End If
```

```
Next
MsgBox oProj.Name
```

2. Нужно открыть проект с сервера Project Server. Здесь все выполняется точно так же, за исключением того, что мы используем метод `FileOpen()` с параметром с хитрым синтаксисом:

```
Application.FileOpen "<>\Забор.Опубликовано"
```

"<>\Забор.Опубликовано" — это, конечно, имя проекта на сервере Project Server. К этому серверу мы должны подключиться в момент запуска Project. При этом при запуске Project просто указать сервер, к которому мы подключаемся, не получится. Придется работать с коллекцией `Profiles` объекта `Application`, который позволяет просмотреть профили, создать новый профиль, настроить параметры подключения (путь к серверу, имя и тип учетной записи), сделать его профилем по умолчанию и т. п. Например, если мы работаем из внешнего приложения, можно создать нужный профиль, сделать его профилем по умолчанию, подключиться к Project Server, а потом вернуть все обратно. Но обычно при использовании Project Server на предприятии пользователь автоматически подключается к нему при каждом запуске Project Professional (в том числе и программном), поэтому выбирать профиль нам не нужно.

Такой же синтаксис ("<>\Забор.Опубликовано") можно будет использовать и потом, при сохранении проекта на Project Server (методом `SaveAs()` объекта `Project`).

После того как мы создали проект, можно работать с его элементами: задачами, ресурсами и назначениями.

Добавление задачи в проект может выглядеть так:

```
Dim oTask As Task
Set oTask = oProj.Tasks.Add("Задача 1")
```

У объекта `Task` огромное количество свойств, которые позволяют настроить любые параметры задачи. Например, у этого объекта есть свойства `ActualStart`, `ActualFinish`, `ActualDuration` и т. п. Но догадаться, как каждое из этих свойств соотносится с нужным нам полем на графическом экране, очень сложно. Обычно намного удобнее после создания задачи использовать не ее свойства, а специальный метод `SetTaskField()` объекта `Application`. Отличие этого метода в том, что он принимает в качестве параметров имя свойства (на русском языке, как оно выглядит на графическом экране), значение и в качестве одного из необязательных параметров `Task.ID`. Например, начало и длительность созданной нами задачи можно установить так:

```
Application.SetTaskField "Начало", Date, False, False, oTask.ID
Application.SetTaskField "Длительность", "2д", False, False, oTask.ID
```

В принципе, средствами этого метода можно сразу создавать задачи, но это обычно не самый удобный способ.

Можно также после настройки нужного параметра на графическом экране сохранить проект в XML-файл и посмотреть информацию о настроенных параметрах элементов.

Другой важный элемент любого проекта — ресурсы. Работа с ресурсами включает в себя несколько задач.

Первая задача — это работа с корпоративным пулом ресурсов. Заполнение корпоративного пула, отслеживание изменений в нем и обеспечение актуальности — один из самых трудоемких компонентов работы с Project Server. Часто информацию о ресурсах нужно синхронизировать с внешними источниками данными (обычно с базами данных). Это можно сделать средствами Project Data Services, а можно — средствами VBA. Открыть корпоративный пул ресурсов можно так:

```
Application.EnterpriseResourcesOpen EUID="", OpenType:=pjReadWrite
```

Если передать этому методу значения обоих параметров (они необязательные), то глобальный корпоративный пул ресурсов будет открыт без каких-либо диалоговых окон. Далее обычным способом (указанным ранее) ловим объектную ссылку на проект, имя которого — "Извлеченные корпоративные ресурсы", создаем в нем ресурсы, а потом сохраняем и закрываем. Другая возможность — импортировать ресурсы в глобальный корпоративный пул при помощи специального метода `Application.EnterpriseResourcesImport()`, которому передается ID локального ресурса.

Другая задача — создание локальных ресурсов. Здесь все просто. В объекте Project предусмотрена коллекция Resources с методом Add(), которая состоит из объектов Resource. Единственная проблема — то, что свойств у ресурсов тоже очень много. Но на помощь нам приходит метод SetResourceField(), аналогичный уже рассмотренному:

```
Dim oRes As Resource  
Set oRes = oProj.Resources.Add("Иванов Иван")  
Application.SetResourceField "Тип", "Трудовой", False, False, oRes.ID
```

Можно, конечно, использовать и стандартные свойства объекта:

```
oRes.StandardRate = 100
```

Разновидность этой задачи — поместить в локальный проект ресурсы из корпоративного пула. Для этой цели проще всего использовать метод `Application.EnterpriseResourceGet()`, которому нужно передать ID глобального ресурса и ID локального ресурса. Предварительно ID глобального ресур-

са можно получить, пройдя по нему циклом и выбрав нужные ресурсы по значению определенных полей.

Следующая задача — произвести назначения. Это можно сделать множеством разных способов:

первый способ — воспользоваться коллекцией `Assignments` и ее метод `Add()`:

```
oTask.Assignments.Add oTask.ID, oRes.ID
```

второй способ — применить метод `Application.ResourceAssignment()`. Он хорош тем, что позволяет назначать ресурсы одновременно нескольким задачам, но эти задачи должны быть выделены (пользователем или программным способом, что снижает надежность этого метода);

третий способ — воспользоваться уже знакомым нам методом `SetTaskField()`:

```
SetTaskField Field:="Названия ресурсов", Value:="Иванов Иван[100%]"
```

Последнее, о чем нужно упомянуть, — это о назначении настраиваемых кодов структуры и корпоративных полей. Если обязательный настраиваемый корпоративный код структуры для проекта, задачи или ресурса не определен, то проект просто нельзя будет сохранить на сервере. Настраиваемые коды структуры и корпоративные поля настраиваются как обычные свойства. Если для задач и ресурсов они доступны напрямую (например, `oRes.EnterpriseText1 = "Мой текст"`), а для проекта — через свойство `SummaryTask`:

```
ActiveProject.ProjectSummaryTask.EnterpriseCost1 = "500.00"
```

Задание для самостоятельной работы 15: Программное создание проекта и его элементов

ЗАДАНИЕ:

Вам необходимо создать макрос, который программно создает проект "Забор" в файле `Забор.mpr` со следующими параметрами:

- время начала проекта — сегодняшнее число;
- задачи проекта:
 - вкапывание столбов. Исполнитель — Иванов Иван с почасовой ставкой 100 руб. в час. Время на выполнение — 2 рабочих дня (16 часов). Материальные ресурсы не используются;

- прибивание досок. Исполнитель — Петров Петр с почасовой ставкой 150 руб. в час. Время на выполнение — 2 рабочих дня (16 часов). Материальные ресурсы — пиломатериалы в размере 2 куб. м по цене 800 руб. за куб. м;
 - покраска забора. Исполнитель — Сидорова Светлана с почасовой ставкой 125 руб. в час. Время на выполнение — 1 рабочий день (8 часов). Материальные ресурсы — краска в количестве 2 банки по цене 100 руб. за банку;
- каждую следующую задачу можно начинать только по завершении предыдущей.

Ответ к заданию 15

Код макроса, который можно запустить из Project, может быть таким:

```
Public Sub CreateProject()  
    Dim oProject As Project  
    Dim oTask As Task  
    Dim oRes, oResMaterial As Resource  
    Dim sID As String  
    Dim oA As Assignment  
  
    'Создаем новый проект  
    Set oProject = Application.Projects.Add()  
  
    'Создаем первую задачу и определяем дату начала и продолжительность  
    Set oTask = oProject.Tasks.Add("Вкапывание столбов")  
    Application.SetTaskField "Начало", Date, False, False, oTask.ID  
    Application.SetTaskField "Длительность", "2д", False, False, oTask.ID  
  
    'Создаем трудовой ресурс и настраиваем его свойства  
    Set oRes = oProject.Resources.Add("Иванов Иван")  
    oRes.Type = pjResourceTypeWork  
    oRes.StandardRate = 100  
  
    'Производим назначение ресурса задаче  
    oTask.Assignments.Add oTask.ID, oRes.ID  
  
    'Сохраняем идентификатор задачи – он потребуется,  
    'чтобы потом указать эту задачу как предшественницу  
    sID = oTask.ID  
  
    'Создаем вторую задачу и второй трудовой ресурс  
    Set oTask = oProject.Tasks.Add("Прибивание досок")
```

```
oTask.Predecessors = sID
Application.SetTaskField "Длительность", "2д", False, False, oTask.ID

Set oRes = oProject.Resources.Add("Петров Петр")
oRes.Type = pjResourceTypeWork
oRes.StandardRate = 150

'Создаем материальный ресурс и настраиваем его свойства
Set oResMaterial = oProject.Resources.Add("Пиломатериалы")
oResMaterial.Type = pjResourceTypeMaterial
oResMaterial.StandardRate = 800
oResMaterial.MaterialLabel = "куб. м"

'Назначаем трудовой ресурс
oTask.Assignments.Add oTask.ID, oRes.ID

'Для материального ресурса нужно указать количество
Set oA = oTask.Assignments.Add(oTask.ID, oResMaterial.ID)
'Оно указывается ... в минутах!
'120 минут (2 часа), т. е. 2 кубометра пиломатериалов
oA.Work = 120

'Опять сохраняем идентификатор задачи
sID = oTask.ID

'Для последней задачи выполняем уже знакомые нам действия
Set oTask = oProject.Tasks.Add("Покраска забора")
oTask.Predecessors = sID
Application.SetTaskField "Длительность", "1д", False, False, oTask.ID

Set oRes = oProject.Resources.Add("Сидорова Светлана")
oRes.Type = pjResourceTypeWork
oRes.StandardRate = 125

Set oResMaterial = oProject.Resources.Add("Краска")
oResMaterial.Type = pjResourceTypeMaterial
oResMaterial.StandardRate = 100
oResMaterial.MaterialLabel = "банка"

oTask.Assignments.Add oTask.ID, oRes.ID

Set oA = oTask.Assignments.Add(oTask.ID, oResMaterial.ID)
oA.Work = 120

'Сохраняем проект в соответствии с поставленными условиями
oProject.SaveAs "C:\Забор.mpp"

End Sub
```