# 项目1

# 门票销售系统

### 《 项目描述

到青岛极地海洋世界看海底动物,首先要去售票窗口买票。本项目将开 发一个门票销售系统应用程序。

## \*\*\*任务分解

本项目分解为2个任务,分别为"欢迎界面"和"购票主界面"。

## 单元 1

## .NET 简介和 C # 概述

Unit 1

🔨 工作任务

本单元完成"欢迎界面"任务。

#### 

- 了解.NET 框架和 C # 语言
- 熟悉 Visual Studio 2012 集成开发环境
- 掌握创建和运行控制台应用程序的方法和步骤(重点)
- 掌握创建和运行 Windows 窗体程序的方法和步骤(重点)
- 掌握窗体、标签、按钮和文本框的基本属性、方法和事件
- 掌握模式窗体和非模式窗体的区别和调用方式

#### 🖬 知识要点

- Microsoft .NET 简介
- C # 概述
- Visual Studio 2012 集成开发环境
- 编程初体验——编写控制台应用程序和窗体应用程序
- C # 源程序的基本结构
- 窗体对象
- 三种常用输入/输出控件

#### \_\_\_\_\_\_典型案例

- 简单控制台应用程序
- 简单 Windows 窗体应用程序
- Form 控件使用
- 常用输入/输出控件使用

## 知识点 1 Microsoft .NET 简介

微软公司总裁兼首席执行官 Steve Ballmer 给.NET 下的定义为: ".NET 代表一个 集合或一个环境,可以作为平台支持下一代 Internet 的可编程结构。"

Microsoft.NET(简称.NET)是微软公司推出的面向网络的一套完整的开发平台。 从程序员的角度看,.NET 是一组用于生成 Web 服务器应用程序、Web 应用程序、 Windows 应用程序和移动应用程序的软件组件,.NET 能支持多种应用程序的开发。本 书采用的是控制台应用程序和 Windows 应用程序,其中控制台程序一般是字符界面,可 以编译为独立的可执行文件,通过命令行运行,在字符界面上输入/输出;Windows 应用 程序是基于 Windows 窗体的应用程序,是一种基于图形用户界面的应用程序。

如图 1.1 所示,.NET 体系结构的核心是.NET 框架(.NET Framework),其在操作系统之上为程序员提供了一个编写各种应用程序的高效工具和环境。.NET 体系结构的顶层是用各种语言编写的应用程序,这些应用程序在.NET 核心组件的支持下运行。



图 1.1 .NET 体系结构

1. .NET 框架的两个核心组件

.NET 框架包括两个核心组件,即公共语言运行环境(CLR)和框架类库。

(1) 公共语言运行环境(CLR)

公共语言运行环境又称公共语言运行时(Common Language Runtime, CLR)或公共 语言运行库,在.NET 框架的底层。其基本功能是管理用.NET 框架类库开发的应用程序 的运行并且提供各种服务。

.NET 将开发语言与运行环境分开,一些基于.NET 平台的所有语言的共同特性(如数据类型、异常处理等)都是在公共语言运行环境层面实现的,在.NET 上集成的所有编程语言编写的应用程序均需通过公共语言运行环境才能运行。使用公共语言运行环境的一大好处是支持跨语言编程,凡是符合公共语言规范(Common Language Specification, CLS)的语言所编写的对象都可以在公共语言运行环境上相互通信、相互调用。

(2) 框架类库

框架类库是一个面向对象的可重用类型集合,该类型集合可以理解成预先编写好的 程序代码库,这些代码包括一组丰富的类与接口,程序员可以用这些现成的类和接口来生成.NET应用程序、控件和组件。例如,运用 Windows 窗体类可以轻松地创建窗体、菜

4

单、工具栏、按钮和其他界面元素,从而大大简化 Windows 应用程序的开发难度。程序员可以直接使用类库中的具体类,或者从这些类中派生出自己的类。.NET 框架类库是程序员必须掌握的工具。

2. Microsoft 中间语言和即时编译器

.NET 框架上可以集成许多编程语言,这些编程 语言共享.NET 框架的庞大资源,还可以创建由不同 语言混合编写的应用程序,因此可以说.NET 是跨语 言的集成开发平台。

如图 1.2 所示,.NET 框架上的各种语言分别有 各自不同的编译器,编译器向 CLR 提供原始信息,各 种编程语言的编译器负责完成编译工作的第一步,即 把源代码转换为用 Microsoft 中间语言(Microsoft Intermediate Language, MSIL)表示的中间代码。



MSIL 是一种非常接近机器语言的语言,但还不能直接在计算机上运行。第二步编译工作就是将中间代码转换为可执行的本地机器指令(本地代码),在 CLR 中执行,这个工作由 CLR 中包含的即时编译器(Just In Time, JIT)完成。

### 知识点 2 C # 概述

C#语言是微软公司专门为.NET 平台量身打造的程序设计语言,是一种强大的、面向对象的程序设计语言,它是为生成运行在.NET 框架上的企业级应用程序而设计的。

C#看起来与 Java 有着惊人的相似,它包括诸如单一继承、接口等特征以及与 Java 几乎同样的语法和编译成中间代码再运行的过程。但是 C#与 Java 仍然有着明显的不 同,它借鉴了 Delphi的一个特点,与 COM(组件对象模型)是直接集成的,而且它是微软 公司.NET Windows 网络框架的主角。

微软公司对 C # 的定义为: C # 是一种安全、现代、简单的由 C 和 C ++ 衍生而来的面向对象编程语言。它根植于 C 和 C ++ 语言之上,并可以立即被 C 和 C ++ 的使用者所熟悉。设计 C # 的目的就是综合 Visual Basic 的高生产率和 C ++ 的行动力,目前,C # 已经成为 Windows 平台上软件开发的绝对主流语言。

1. C # 的 由 来

微软公司从 1998 年 12 月开始了 COOL 项目,直到 1999 年 7 月 COOL 被正式更名 为 C # (读作 C Sharp)。2000 年 6 月,微软公司在奥兰多举行的"职业开发人员技术大 会"上正式发布了新的语言 C #,它是一种面向对象的、运行于.NET Framework 之上的 高级程序设计语言。安德斯·海尔斯伯格(Anders Hejlsberg,1960—),丹麦人,Turbo Pascal 编译器的主要作者,Delphi 和 C # 之父,同时也是.NET 创立者。



2. C # 的 特 点

C#是一种安全、稳定、简单、优雅且由C和C++衍生出来的面向对象的编程语言,它继承了C和C++强大功能的同时,去掉了一些它们的复杂特性。其具体特点如下。

- (1) 简洁的语法。
- (2) 与 Web 的紧密结合。
- (3) 精心的面向对象设计。
- (4) 完整的安全性与错误处理。
- (5) 版本处理技术。
- (6) 灵活性和兼容性。

#### 3. C # 、NET 与 Visual Studio 的关系

.NET 框架是微软公司推出的一个全新的开发平台; Visual Studio 则是微软公司为 了配合.NET 战略推出的集成开发环境,同时它也是目前开发 C # 应用程序最好的工具; C # 只是基于.NET 框架的程序开发语言的一种,它并不是.NET 的一部分。

在安装 Visual Studio 的同时,.NET 框架也会自动安装上。安装过程中可以选择安装 C # 语言、VB 语言或者 C ++ 语言等,也可以选择都安装。C #、.NET 与 Visual Studio 各个版本之间的对应关系如表 1.1 所示。

集成开发环境版本	开发平台版本	C#语言版本
Visual Studio 2002	.NET Framework 1.0	C # 1.0
Visual Studio 2003	.NET Framework 1.1	C# 1.1
Visual Studio 2005	.NET Framework 2.0	C # 2.0
Visual Studio 2008	.NET Framework 3.5	C# 3.5
Visual Studio 2010	.NET Framework 4.0	C # 4.0
Visual Studio 2012	.NET Framework 4.5	C# 5.0
Visual Studio 2015	.NET Framework 4.6	C # 6.0
Visual Studio 2017	.NET Framework 4.7	C# 7.0

表 1.1 C # 、.NET 与 Visual Studio 各个版本之间的对应关系

### 知识点 3 Visual Studio 2012 集成开发环境

每一个正式版本的.NET Framework 都会有一个与之对应的集成开发环境,微软公司称之为 Visual Studio,也就是可视化工作室。其中,Visual Studio 2012 版本是一个功能强大的集成开发环境,在该开发环境中可以创建控制台程序、Windows 应用程序、ASP. NET 应用程序和 ASP.NET 服务等。

Visual Studio 2012 集成开发环境如图 1.3 所示,主要包括菜单栏、工具栏、窗体设计

器、工具箱、属性窗口、解决方案资源管理器和代码编辑器等。

M N	/indowsFormsApplic	ation1 - M	icrosoft Vis	ual Studio(	(管理员)		快速启动	) (Ctrl+	Q)	م	_ 0	x
文件(E)	编辑(E) 视图(V)	项目(P)	生成(B)	调试(D)	团队(M)	SQL(Q)	工具	DI	U试(S)	体系结构((	) 分析(N	)
窗口(W	) 帮助( <u>H</u> )											
G -	0 🕆 🖀 🗎 🛛	12-	€ - ►	启动 ▼ De	ebug +	<b>#</b>	-   = =	= =	11	<u>  </u>   *  <u> </u>		. <del>  </del>
工具箱		¶× Fo	rm1.cs [设计	-] +> ×				*	解决方	5案资源管理	₩ 00000 -	Ψ×
搜索工具	見箱	<u>م</u>						50	G	oinc	- 0 🗇 🕯	a
▷ 所有	Windows 窗体		🖳 Form1				23		搜索的	山方安资源	= 宇理器(Ctrl+	Q -
▲ 公共	控件								132.304			
h.	指针									啡伏刀条 ₩ ℡Window:	FormsApp	licatic
ab	Button								Þ	Prope	rties	
$\checkmark$	CheckBox								Þ	■引用		
	CheckedListBox									App.c	onfig	
Ē	ComboBox							þ	Þ	Form1	cs	
Α	Label								P	C# Progr	am.cs	
A	LinkLabel										Ver VEE Ante	+
	ListBox								解決力	「蒸箕…」 図1)	(贫源官 3	制造
	DateTimePicker								属性		•••••••••••	Ψ×
	ListView								Form	1 System.W	/indows.For	ms
o.	MaskedTextBox									↓ √⊐ <b>f</b>	P	
	MonthCalendar	i.			0			-6	⊞ Siz	e	300, 300	
Les 1	NotifyIcon								Siz	eGripStyle	Auto	
D\$	NumericUpDown								Sta	rtPosition	WindowsDe	fai
~	PictureBox	_							Tag	9		- 11
服务器资	资源管理器 工具箱	÷							Tex	kt .	Form1	-
就绪												.#

图 1.3 集成开发环境

1. 菜单栏

菜单栏主要包括"文件""编辑""视图""项目""生成""调试"等菜单项,不同的 Visual Studio 版本和不同的运行环境,菜单项略有不同。这些菜单项提供了程序设计过程中的所需功能。

2. 工具栏

工具栏以图标形式提供了常用命令的快速访问按钮,单击某个按钮可以执行相应的操 作。Visual Studio 2012 将常用命令按功能的不同进行了不同分类。可以通过"视图"→"工 具栏"命令来打开不同的工具栏。

3. 解决方案资源管理器

使用 Visual Studio 2012 开发的每一个应用程序都叫作一个解决方案,每一个解决方案可以包含一个或多个项目。一个项目通常是一个完整的程序模块,并且可以有多个 文件。

解决方案资源管理器位于集成开发环境右上方。如果在集成开发环境中已经创建了 方案或项目,则项目中所有文件以分层树的形式显示,如图 1.4 所示。



工具箱在默认情况下位于集成开发环境的左侧,其中包括了 Visual Studio 2012 的重要工具。它含有许多可视化的控件,用户可以从中选择相应的控件并将它们添加到窗体上,进行可视化界面的设计。如果没有显示,可以通过"视图"→"工具箱"命令将其打开。 工具箱中的控件和各种组件按照功能进行了分组,如图 1.5 所示。



图 1.4 解决方案资源管理器



图 1.5 工具箱的分组

5. "窗体设计器/代码编辑器"窗口

"窗体设计器/代码编辑器"窗口是 Visual Studio 2012 集成开发环境的主窗口。窗体 设计器用于进行可视化的设计,用户可以将各种控件放在上面,完成用户界面的设计;代 码编辑器用来进行代码的设计。如果当前项目是 Windows 窗体应用程序,可以使用以下 方法实现两窗体之间的切换。

(1) 按 F7 键显示代码编辑器窗口,按 Shift+F7 组合键显示窗体设计器窗口。

(2) 选择"视图"→"代码"或"视图"→"设计器"命令。

(3)当代码编辑器窗口和窗体设计器窗口被打开后,在主窗口上方就会出现选项卡, 可以通过单击选项卡标签来切换。窗体设计器窗口如图 1.6 所示,在该窗口中可以为 Windows 界面添加并设置控件。代码编辑器窗口如图 1.7 所示,这是一个纯文本编辑器, 在其中可以进行常见的文本编辑操作,如定位、选定、复制、剪切、粘贴、移动、撤销、恢复等 操作。代码编辑器窗口以不同的颜色显示代码中不同含义的内容,如以蓝色显示关键字, 以绿色显示注释,以蓝绿色显示类名。控制台应用程序只有代码编辑器窗口。

6. "属性"面板

"属性"面板在默认情况下位于集成开发环境的右下方,如图 1.8 所示。主要用来设



图 1.6 窗体设计器窗口

置控件的属性和事件。在 Windows 窗体的设计视图下,在"属性"面板中可以设置控件的 属性或者链接用户界面控件的事件。"属性"面板同时采用了两种方式来管理属性和方 法,即按"分类顺序"和"字母顺序",用户可以根据自己的习惯采取不同的方式。

Form1.cs* 中 × Form1.cs [设计]*	-		
🔩 WindowsFormsApplication1.Form1 🔹 🗣 Form1_Load(object sender, EventArgs 🤄	•		
⊟namespace WindowsFormsApplication1		属性	
{		Form1 System.Wi	ndows.Forms.Form
😑 public partial class Form1 : Form		🏥 💱 🖓 🗲 🏓	
{ □ public Form1() {		<ul> <li>Padding</li> <li>RightToLeft</li> <li>RightToL eftLavo</li> </ul>	0, 0, 0, 0 No False
InitializeComponent();		Showlcon	True
}		ShowInTaskbar	True
			496, 401
private void Formi_Load(object sender, EventArgs e)		SizeGripStyle	Auto
- · ·		StartPosition	WindowsDefaultL
}		Tag	
		Text	Form1
}		TopMost	False
		TransparencyKey	/
		UseWaitCursor	False
		WindowState	Normal
		Text 与控件关联的文本。	

图 1.7 代码编辑器窗口

图 1.8 "属性"面板

• 🗆 ×

窗体和控件都有自己的属性,用户可以通过"属性"面板对控件的属性值进行修改。 在集成开发环境下,"属性"面板被关闭后,可以通过"视图"菜单打开或通过快捷键来将其 打开,即通过"视图"→"属性窗口"命令或按 Alt+Enter 组合键。

## 知识点 4 编程初体验

下面分别通过两个实例介绍控制台应用程序和窗体应用程序的编写和运行方法。

【案例 1.1】 创建一个控制台应用程序,输出"Hello, world!",显示效果如图 1.9 所示。

案例设计步骤如下。

(1) 新建项目

选择"开始"→Microsoft Visual Studio 2012→Visual Studio 2012 命令,如图 1.10 所示。



选择"文件"→"新建"→"项目"命令,打开"新建项目"窗口,在左侧选择 Visual C♯, 在右侧选择"控制台应用程序",并在下方的名称中输入 helloworld,位置中选择"E:\

mybook",单击"确定"按钮,如图 1.11 所示。

(2) 编写代码

在代码窗口的 Main 方法中添加两行代码,加入代码后的 Main 方法如下:

```
static void Main(string[] args)
{
    Console.WriteLine("Hello,world!"); //输出一行信息
```