

# 实验三 集成编程环境与编码规范 实验报告

姓名： 姜丽

学号： 08133341

班级： 计算机科学与技术 13-5 班

编写：2015 年 12 月 30 日

# 目录

1 实验目的.....	1
2 实验内容.....	1
3 实验结果.....	1
3.1 Java.....	1
3.1.1 JAVA 语言编程规范.....	1
3.1.2 JAVA jdk 的安装, 环境的配置, 集成开发环境 Eclipse 的安装.....	2
3.1.3 集成开发环境的使用.....	5
3.1.4 集成开发环境的调试技术.....	8
3.2 PHP.....	10
3.2.1 PHP 语言基础.....	11
3.2.2 安装配置 PHP.....	14
3.2.3 PHP 集成开发环境 PhpStorm 的下载安装.....	14
3.2.4 集成开发环境的使用及调试技术.....	18
3.3 C#.....	29
3.3.1 C#语言基础.....	29
3.3.2 集成开发环境 Visual Studio 2008 的安装.....	32
3.3.3 集成开发环境的使用.....	32
3.3.4 集成开发环境的调试技术.....	35
4 实验小结.....	37

---

## 1 实验目的

学习 PHP、.Net、JAVA 集成编程环境，熟悉各语言的编码规范。

## 2 实验内容

- (1) 学习 PHP 语言基础，学习 PHP 集成编程环境 EPP 或 PhpStorm，了解常用插件，掌握 PHP 断点调试功能；
- (2) 学习 C#语言基础，学习.Net 集成编程环境，掌握集成环境中的调试技术；
- (3) 学习 JAVA 集成编程环境 Eclipse，参考《Java 编码规范及实践》，熟悉 JAVA 的编码规范。

## 3 实验结果

### 3.1 Java

#### 3.1.1 JAVA 语言编程规范

##### (1) 命名规范

所有包、类、接口、方法、属性、变量、参数等均使用英文单词进行命名，尽量使用贴近问题域的表意丰富的名称，名字中只能出现字母、下划线、\$、数字，并且只能以字母、下划线开头，在一个程序中的一个标识符只用于一个用途。

其中，包名应该用小写字母，不要出现下划线等符号。

类名应使用名词组合，其中每个单词首字母大写。异常类应使用 Exception 作为后缀。抽象类应使用 Abstract 前缀。查询方法应使用 find 作为前缀。

接口名应使用字母”I”加上名词组合，其中每个单词首字母大写的格式。接口以 able 作为后缀，代表了一种能力。

变量名和参数名使用名词组合的方式，第一个单词小写，其余单词的首字母大写，其余字

---

母小写。变量的名字应该与类型名称一致。作用范围较大的变量应使用长名称，作用范围小的变量应使用简化名称，提高编程效率。

常量名使用大写字母，并使用下划线做间隔。用常量代替实际的数字，以方便以后对数字的修改。

方法名应使用动词开头，由动词+名词组成，第一个单词小写，其余单词的首字母大写，其余字母小写。

缩写字母应让首字母大写。

## (2) 注释规范

代码应和注释保持同步，如果代码和注释不同步，则阅读代码的人会想到到底是代准确还是注释准确，会使人糊涂。

注释尽量简洁，注释决不是可有可无，一个正规的程序文本中注释行的数量占到整个程序的 1/3 到 1/2。

注释分为序言性注释和功能性注释，序言式注释是至于每个程序模块的开头部分，给出程序的整体说明，对于理解程序本身具有引导作用。包括程序标题、主要算法、接口说明、有关数据描述、模块位置、开发简历。功能性注释嵌在源程序中，用于描述其后的语句或程序段是在做什么工作。描述的是思想而不是解释怎么做。

注释描述的是一段程序而不是一句程序。使用缩进和空行，是程序和注释容易区别。

## (3) 视觉组织

代码缩进，应使用 4 个空格作为一个单位进行缩进。空循环体也要使用玩真的 {} 块。条件语句即使是单条语句也要用括号括起来。运算符两边应该各有一个空格，保留字后边应该跟随一个空格，逗号和分号后边应跟随一个空格，冒号两个应各有一个空格。在文件头部注释、package、import 语句之间，class 之间，方法之间，变量声明和具体代码之间，块注释和行注释的前面都要加空行。

### 3.1.2 JAVA jdk 的安装，环境的配置，集成开发环境 Eclipse 的安装

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>，到该网页下载 JAVA jdk 后需要配置环境变量，而后可以运行 java 程序。



图 3-1 配置环境变量

```
命令提示符
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>javac
用法: javac <options> <source files>
其中, 可能的选项包括:
-g 生成所有调试信息
-g:none 不生成任何调试信息
-g:<lines,vars,source> 只生成某些调试信息
-nowarn 不生成任何警告
-verbose 输出有关编译器正在执行的操作的消息
-deprecation 输出使用已过时的 API 的源位置
-classpath <路径> 指定查找用户类文件和注释处理程序的位置
-cp <路径> 指定查找用户类文件和注释处理程序的位置
-sourcepath <路径> 指定查找输入源文件的位置
-bootclasspath <路径> 指定查找输入源文件的位置
-extdirs <目录> 覆盖所安装扩展的位置
-endorseddirs <目录> 覆盖签名的标准路径的位置
-processor:<none,only> 控制是否执行注释处理和/或编译。
-processor <class1>[,<class2>,<class3>]... 要运行的注释处理程序的名称; 绕过默
认的搜索进程
-processorpath <路径> 指定查找注释处理程序的位置
-parameters 生成元数据以用于方法参数的反射
-d <目录> 指定放置生成的类文件的位置
-s <目录> 指定放置生成的源文件的位置
-h <目录> 指定放置生成的本机标头文件的位置
-implicit:<none,class> 指定是否为隐式引用文件生成类文件
-encoding <编码> 指定源文件使用的字符编码
-source <发行版> 提供与指定发行版的源兼容性
-target <发行版> 生成特定 VM 版本的类文件
-profile <配置文件> 请确保使用的 API 在指定的配置文件中可用
-version 版本信息
-help 输出标准选项的提要
-A关键字 [=值] 传递给注释处理程序的选项
-X 输出非标准选项的提要
-J<标记> 直接将 <标记> 传递给运行时系统
-Werror 出现警告时终止编译
@<文件名> 从文件读取选项和文件名
```

```
命令提示符
D:\>javac HelloWorld.java

D:\>java HelloWorld
Hello World!

D:\>_
```

图 3-2 运行

下载安装 Eclipse 集成开发软件：

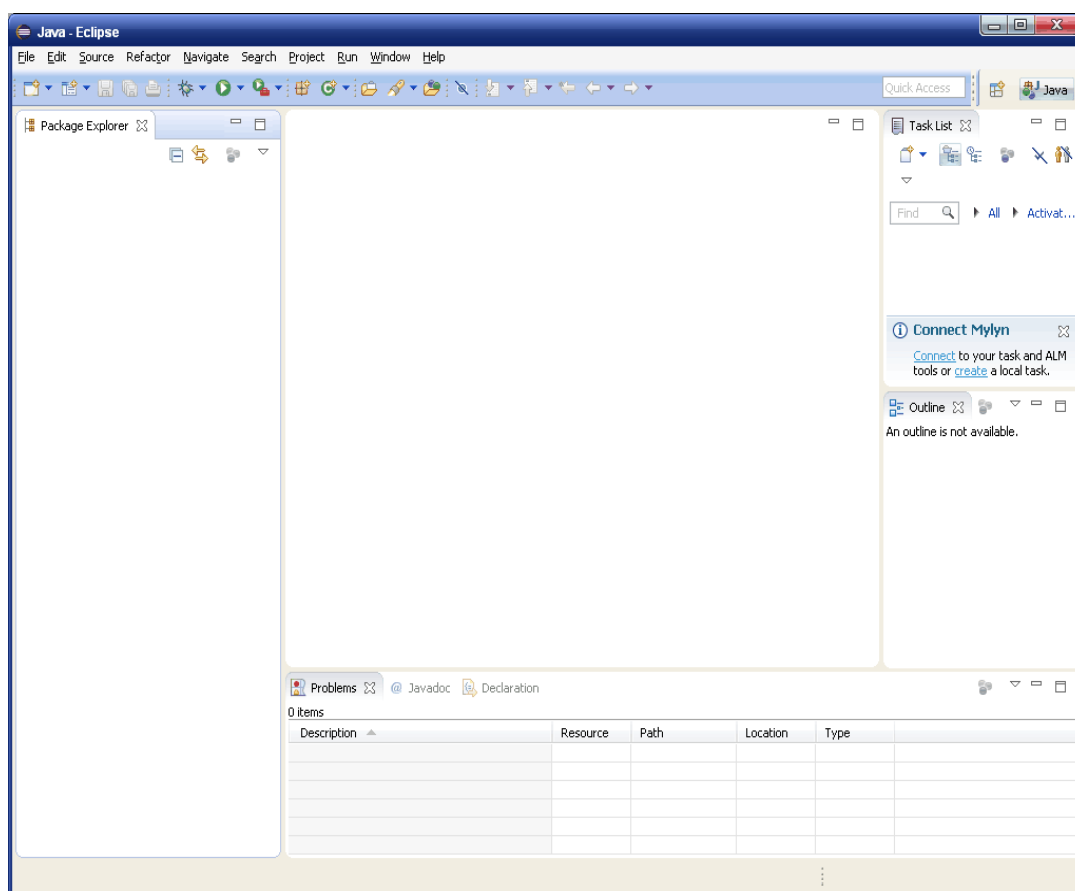


图 3-3 安装 Eclipse 后的界面

### 3.1.3 集成开发环境的使用

点击右键选择 New，选择 Java Project，新建一个 Java 工程：

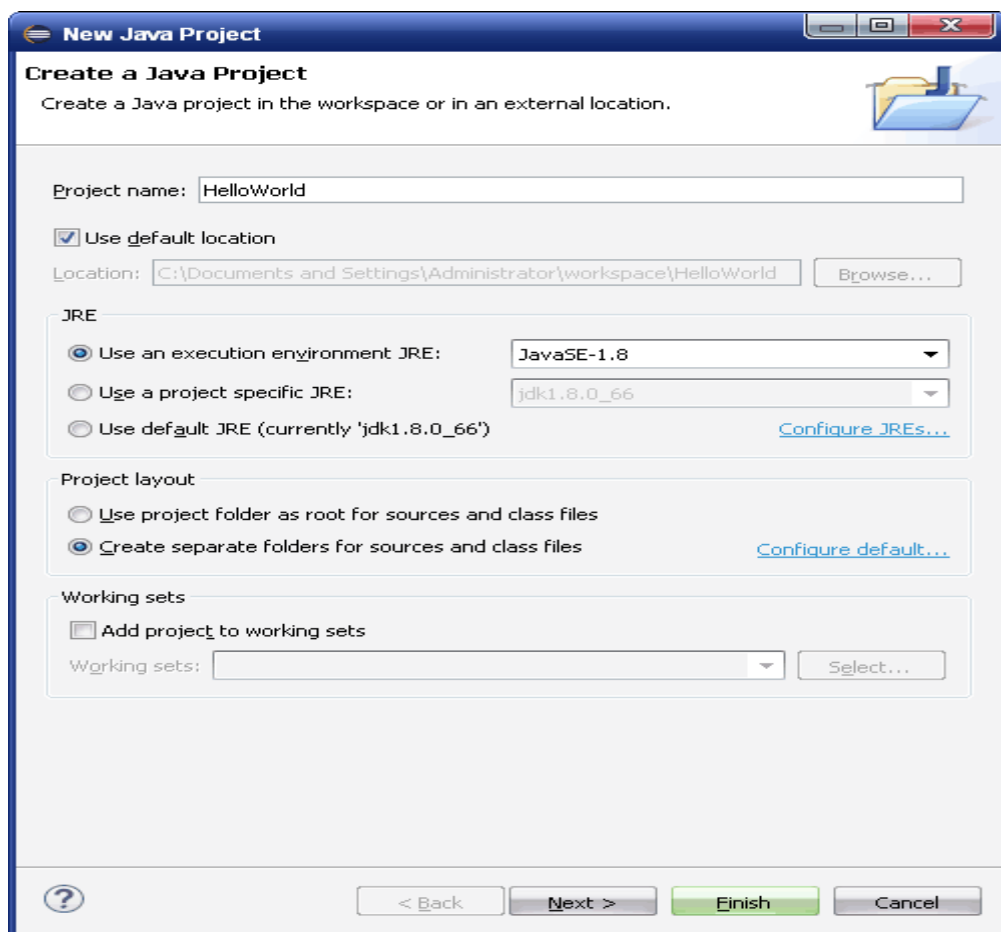


图 3-4 新建 Java 工程

点击右键选择 New，选择 Class，新建一个类：



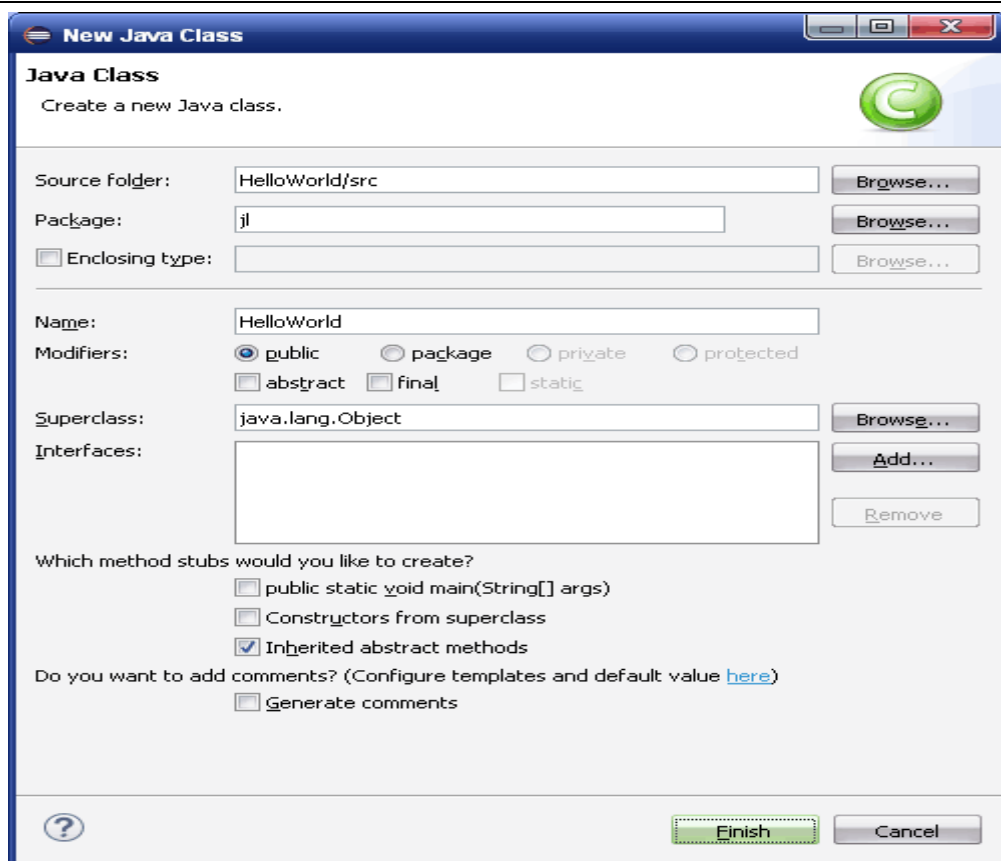


图 3-5 新建类

编写一个 HelloWorld 程序

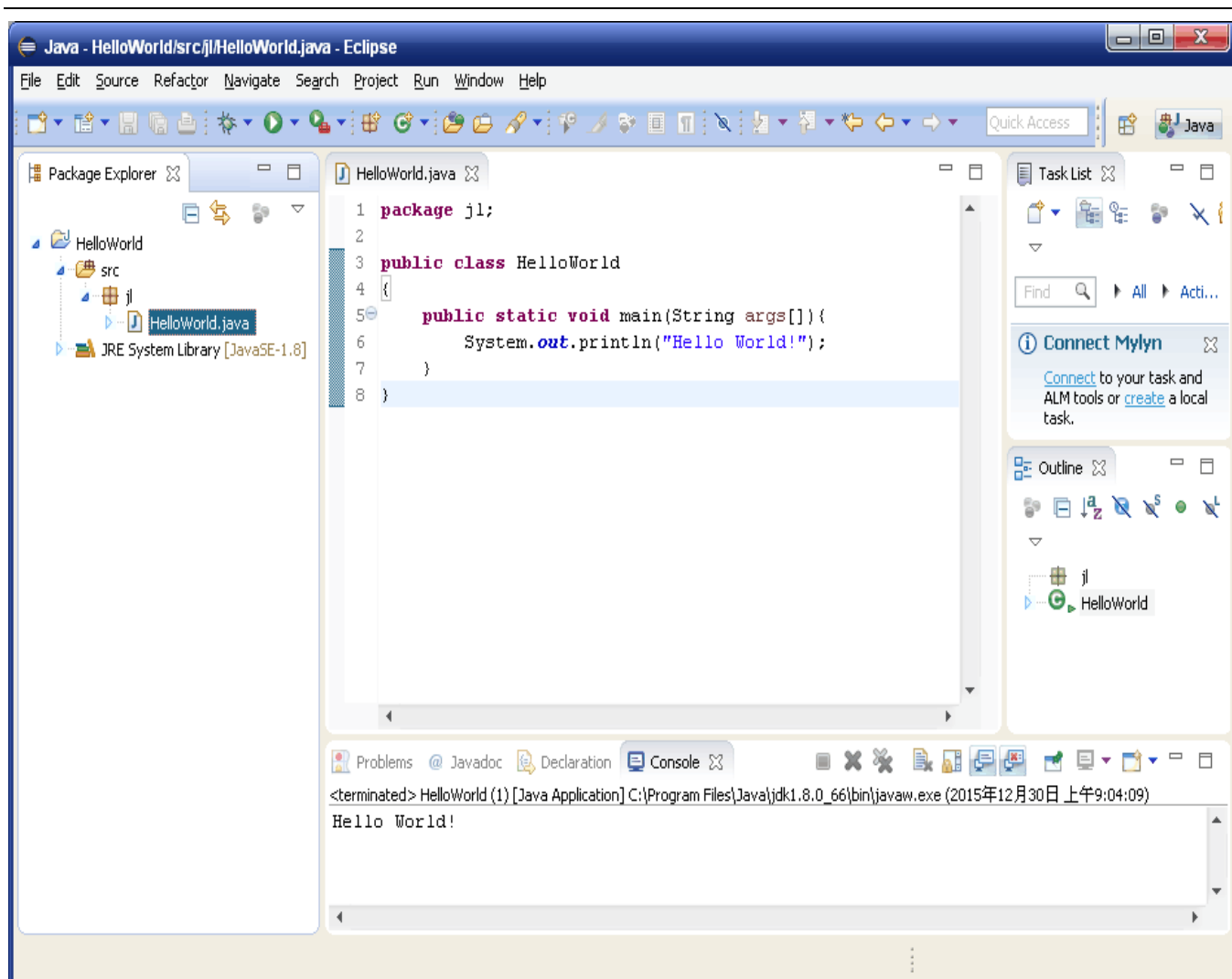


图 3-6 编写并运行程序

### 3.1.4 集成开发环境的调试技术

断点就是多行的程序中，调试时，当程序运行到设置的断点处，就会停止，然后显示出相关的运行参数，信息，以供检查程序的错误所在。

设置断点的方法有三个：

- (1) 点击菜单栏“Run”，在弹出的下拉框内点击“Toggle Breakpoint”，设置断点。
- (2) 把鼠标移动到想要设置断点的行，在行号前面空白的地方双击，就会出现断点。
- (3) 把鼠标移动到要设断点的地方，直接用快捷键“Ctrl+Shift+b”，设置断点。

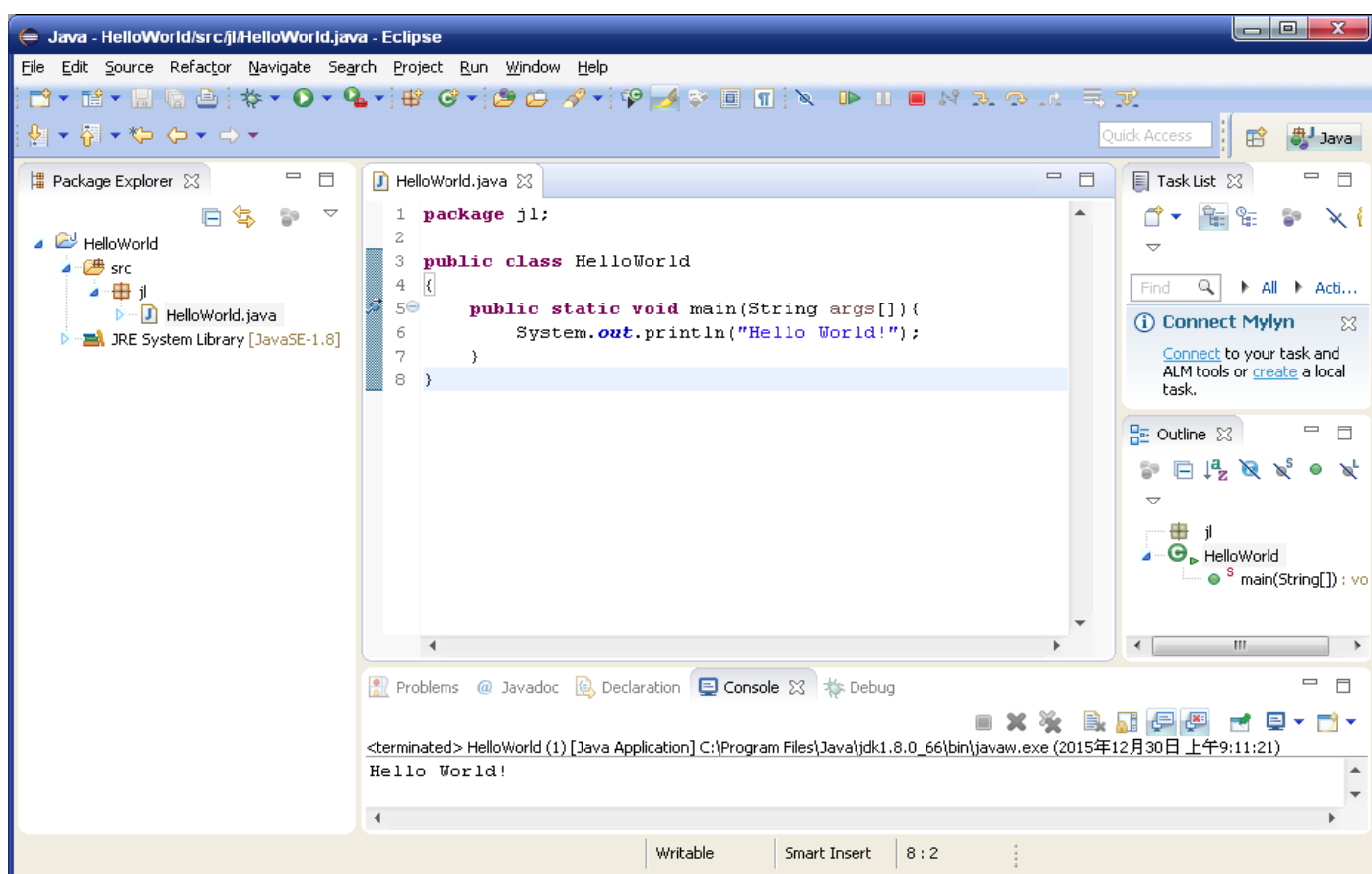


图 3-7 设置断点

设置完断点点击菜单栏中的 Run 下的 Debug As，选择 Java Application，进入调试模式。

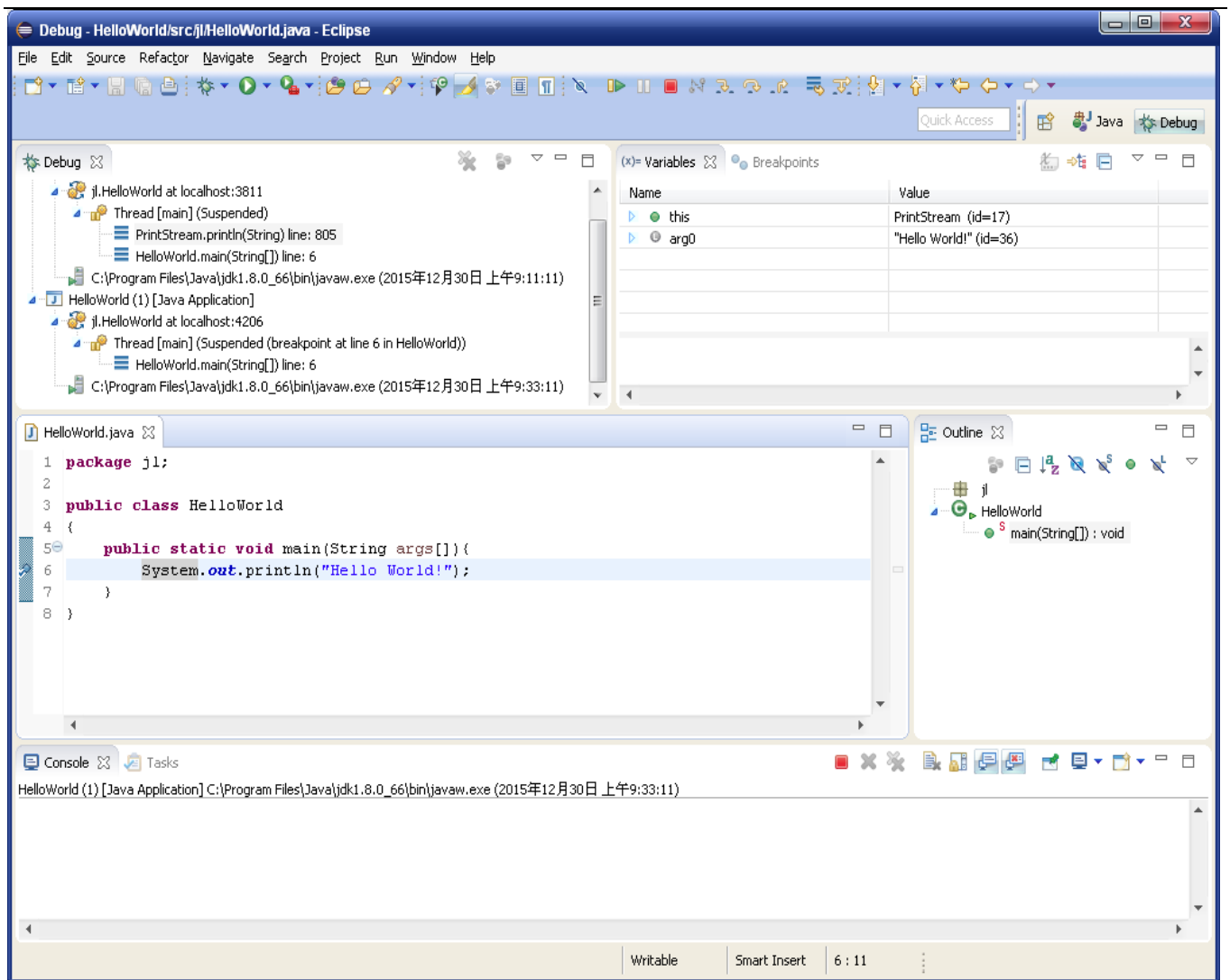


图 3-8 调试模式

右上方就是所有的断点列表,可以查看断点信息,可以通过点击菜单栏中的 Run,选择 step int (步入), step over (步过), step return (返回) 让程序一步一步慢点运行。

## 3.2 PHP

PhpStorm 是一个轻量级的和智能的 PHP IDE, 包括 HTML / CSS 编辑器, JavaScript 编辑器, 并全面支持 PHP。提供智能代码完成功能, 快速导航, 错误检查, 运行单元测试, 或提供可视化的调试。

PHP 是英文超文本预处理语言 Hypertext Preprocessor 的缩写。PHP 是一种 HTML 内嵌式的语言, 是一种在服务器端执行的嵌入 HTML 文档的脚本语言, 语言的风格有类似于 C 语言, 被广泛地运用。

---

### 3.2.1 PHP 语言基础

#### (1) 命名约定

##### a 常量名:

全局常量, 用'\_'分隔每个单词, 使用 TPLIB\_前缀。如: `define("TPLIB_TEST_HELLO", "Hello world!");`

类常量, 用'\_'分隔每个单词, 如: `class TPLIB_Package { const EXAMPLE_CONST; }`, 尽量使用类常量代替全局常量。

##### b 变量名:

局部变量的字母都使用小写, 使用“\_”作为每个词的分界。如: `$php_var;`

全局变量应该带前缀 ‘g’。这样可以知道一个变量的作用域。例如: `global $gLog; global &$grLog;`

静态变量应该带前缀 ‘s’。

例如: `function test() {static $msStatus = 0;}`

函数、方法的参数, 首字母为小写。如: `function test_func($paramName) { }`

##### c 类命名:

使用大写字母作为词的分隔, 其他的字母均使用小写, 名字的首字母使用大写, 不要使用下划线(' \_')

例如: `class NameOneTwo`      `class Name`

##### d 类属性:

使用骆驼命名法命名, 与函数、方法的参数命名方法相同;

##### e 类的方法:

采用与类命名一致的规则, 例如 `getCache()`、`echoName()`。名字采用“动词+宾语”的形式。例如:

`class NameOneTwo`

---

```
{  
  
function DoIt() {};  
  
function HandleError() {};  
  
}
```

#### f 函数命名:

所有函数名必须全部使用英文小写字母，函数名中的单词之间使用下划线间隔。某些情况下，函数名中更适合使用动词。例如：print\_login\_status(), get\_user\_data(), 等等。

#### g 引用变量和函数返回引用:

引用必须带 ‘ r’ 前缀，这样使得类型不同的变量容易辨认，它可以确定哪个方法返回可更改对象，哪个方法返回不可更改对象。

例如

```
class Test  
  
{  
  
var mrStatus;  
  
function DoSomething(&$rStatus) {};  
  
function &rStatus() {};  
  
}
```

#### h 名称空间与类命名:

使用帕斯卡命名方法，如 HelloWorld。

#### i 文件命名:

包含文件应该以.inc.php 方式命名，例如 config.inc.php。单独类的文件使用 Classname.php 方式命名，包中的类使用 PackageName/ClassName.php 命名。

### (2) 书写规则:

#### a 使用大括号

---

在语言结构 (if, else, while, switch, for, foreach) 中请在陈述与执行的代码分行, 执行的代码放到大括号中, 大括号不可省略。

#### b 大括号的位置

在语言结构 (if, else, while, switch, for, foreach) 中和类 (class)、函数 (function)、方法 (method) 中, 左、右大括号必须单独占一行, 与其声明处在相同的缩进级别。

#### c 数组格式

对于数组的定义, 可以使用分行表述每个 "key => value," , 每行开头使用一个 Tab 进行缩进。右括号和该 array (的起始行保持对齐。

#### d 在运算符之间使用空格

在比较运算符 (>、<、>=、<=、==、===、!=、<>、!==)、赋值运算符 (=)、数学运算符 (+、-、\*、/、%)、位运算符 (&、|、^、~、>>、<<)、逻辑运算符 (!、&&、||)、冒号(:)、问号(?)、字符串连接运算符 (.)、字符串连接赋值运算符 (.=) 前后, 以及左括号 ( ( ) 前 (函数调用例外)、逗号 (,) 后使用空格进行间隔。

#### e 运算符优先级

对于容易引起迷惑的表达式中不同运算符的优先级, 请使用括号来区分优先级。

#### f 语法结构

在 PHP 中 echo、exit(die)、return、continue、break、include、include\_once、require、require\_once 等都属于语法结构, 大部分语法结构都有两种形式:

```
echo 'This is a string';
```

```
echo('This is a string');
```

在 PHP 规定的允许使用的格式下, 尽可能使用前一种语法结构的格式, 而不要使用函数参数/表达式的形式, 仅在参数包含表达式时才需要用括号将其括起来。由于 echo() 是语法结构、没有返回值, 所以速度比 print() 快。在输出多个字符串时, 使用 echo 的多参数方式 (逗号, 间隔), 会比字符串连接方式 (点. 间隔) 有更好的性能。

#### g 缩进/制表符/空格规则

---

使用制表符缩进或使用三到四个空格为每层次缩进。

h 每行一个语句

除非这些语句有很密切的联系，否则每行只写一个语句。

i 短方法

方法代码要限制在一页内。

j PHP 代码标记

统一使用<?php ?>

k 注释

// 这是单行注释

# 这也是单行注释

/\*

这是多行注释块

它横跨了

多行

\*/

### 3.2.2 安装配置 PHP

登 陆 php 网 站 <http://windows.php.net/download#php-7.0> 下 载  
php-5.4.45-Win32-VC9-x86.zip，解压缩后即可。

### 3.2.3 PHP 集成开发环境 PhpStorm 的下载安装

首先登陆 PhpStorm 官网 <http://www.jetbrains.com/phpstorm/>。根据系统环境 (windows/mac)，下载对应的版本的 30 天使用版。



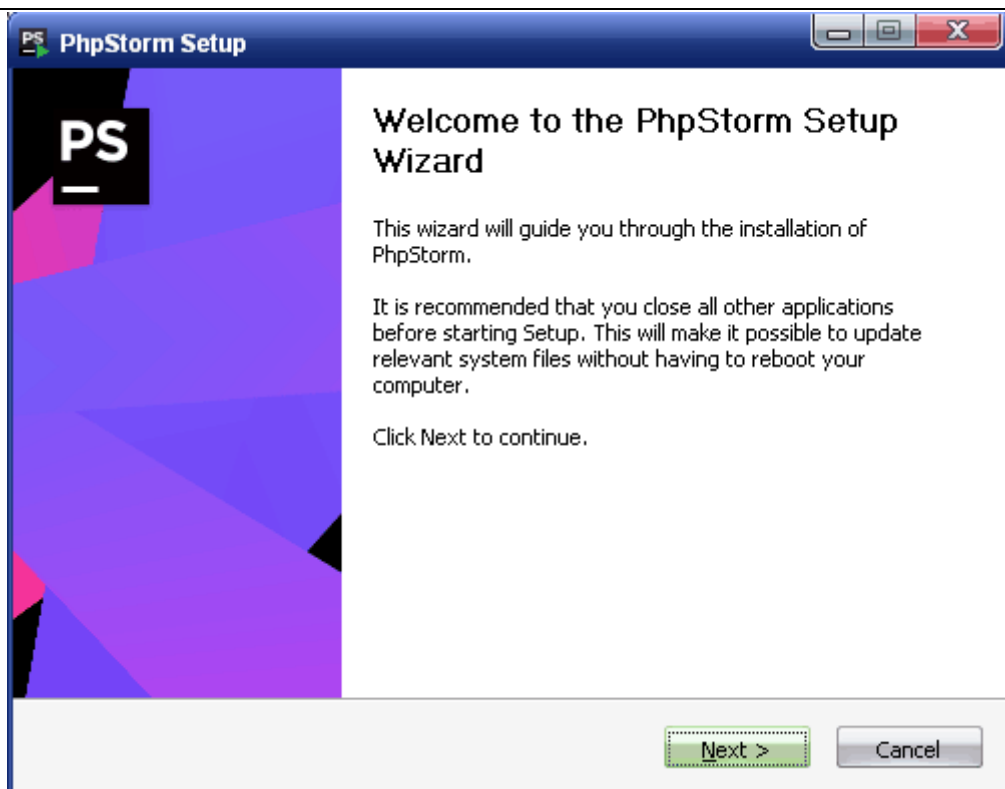


图 3-9 下载界面 1

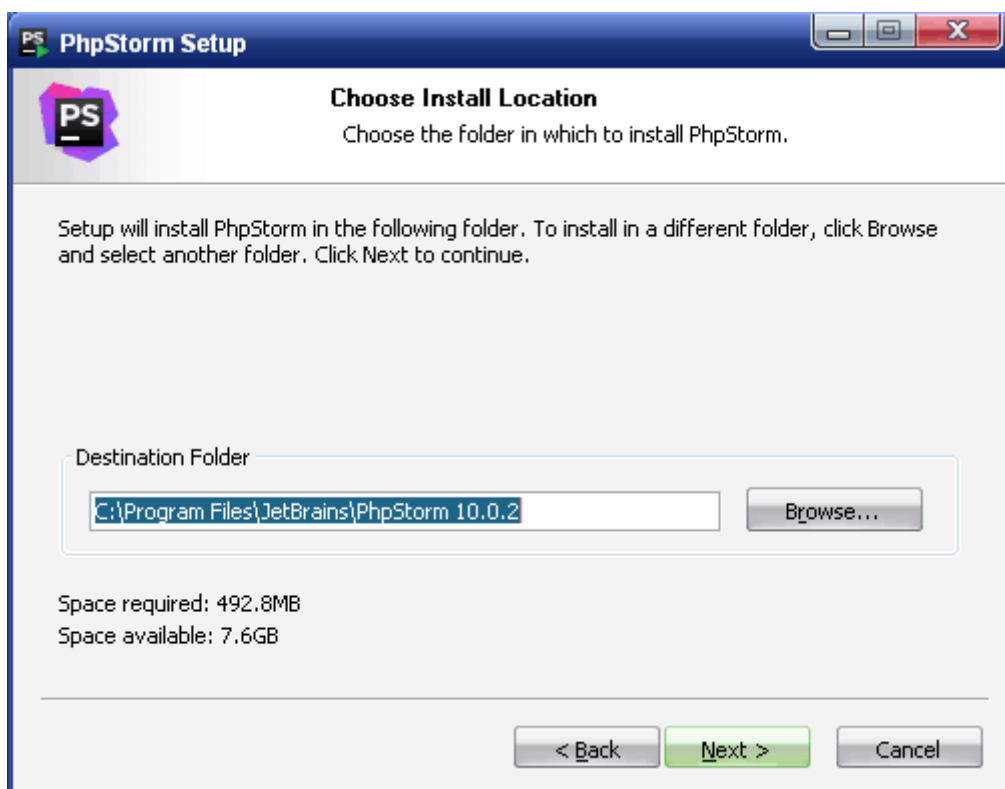


图 3-10 下载界面 2

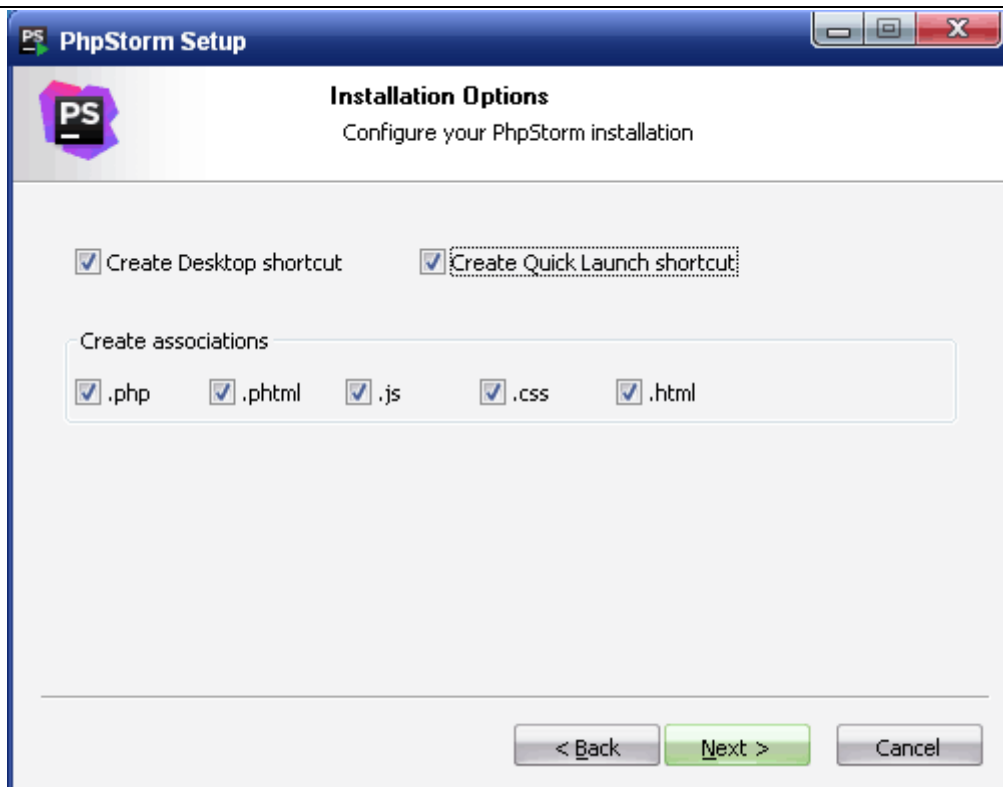


图 3-11 下载界面 3

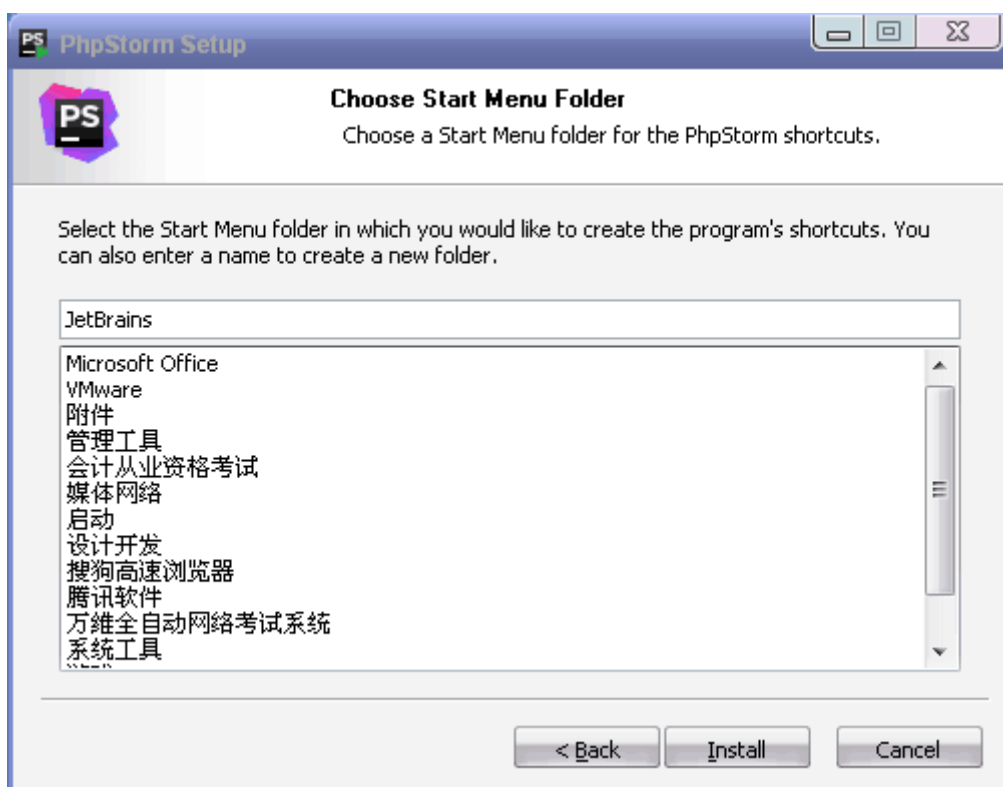


图 3-12 下载界面 4

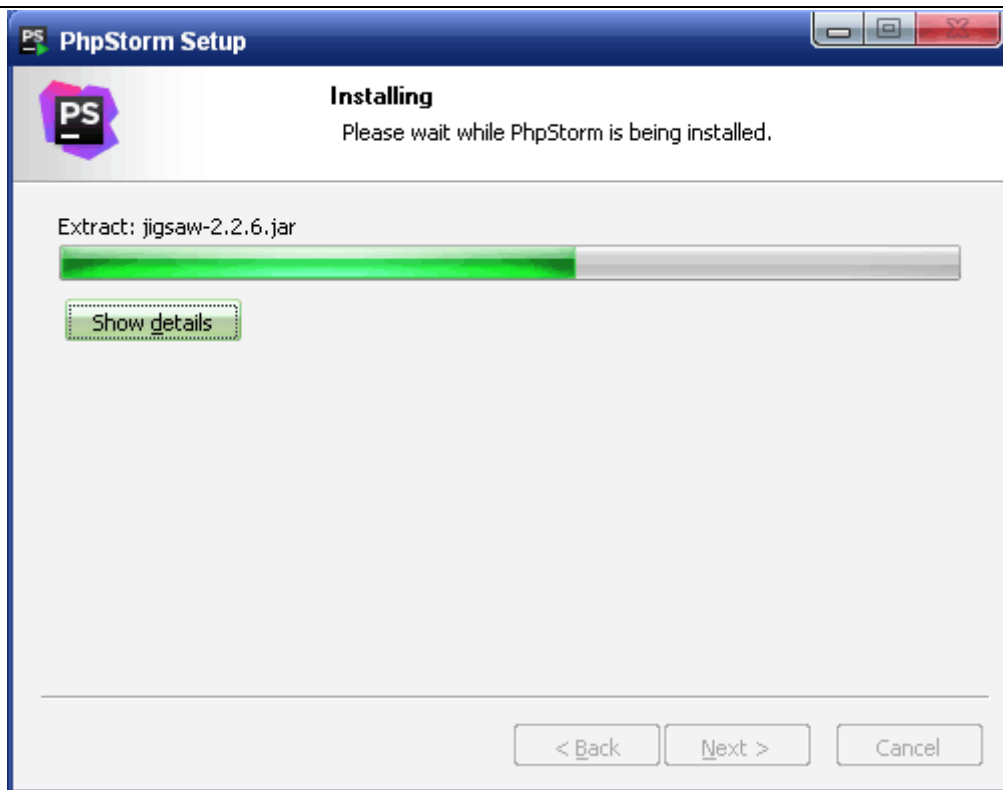


图 3-13 下载界面 5

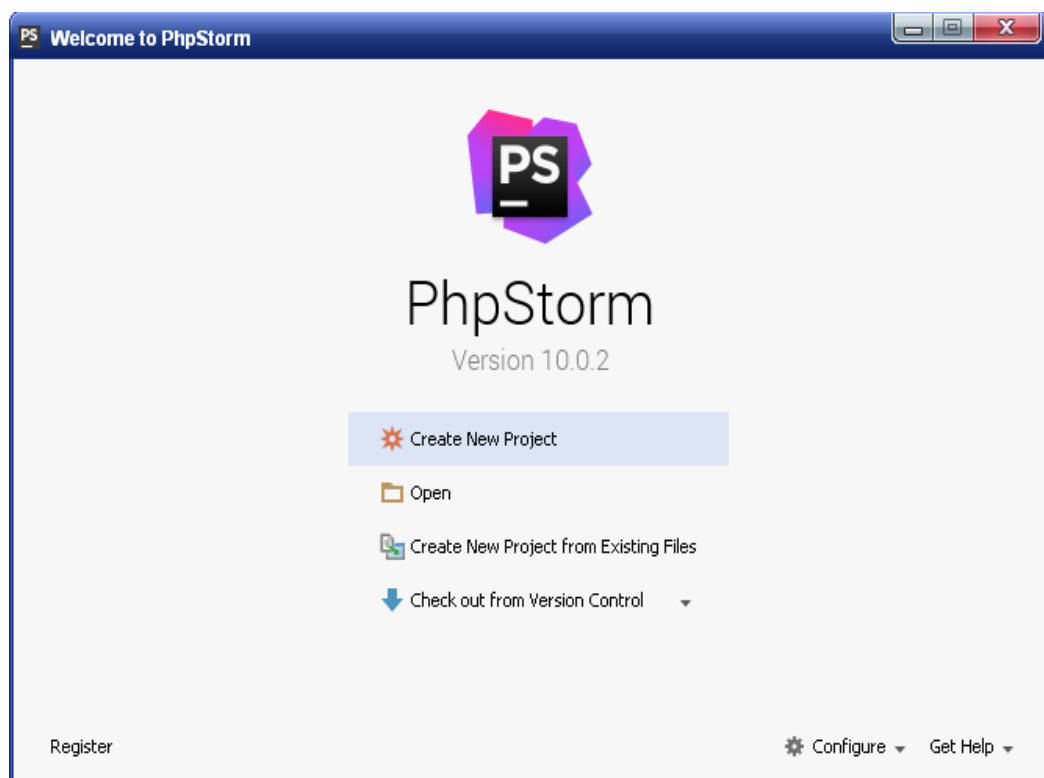


图 3-14 安装完成后的界面

### 3.2.4 集成开发环境的使用及调试技术

打开软件，点击 File 下的 New Project, 新建工程 hello

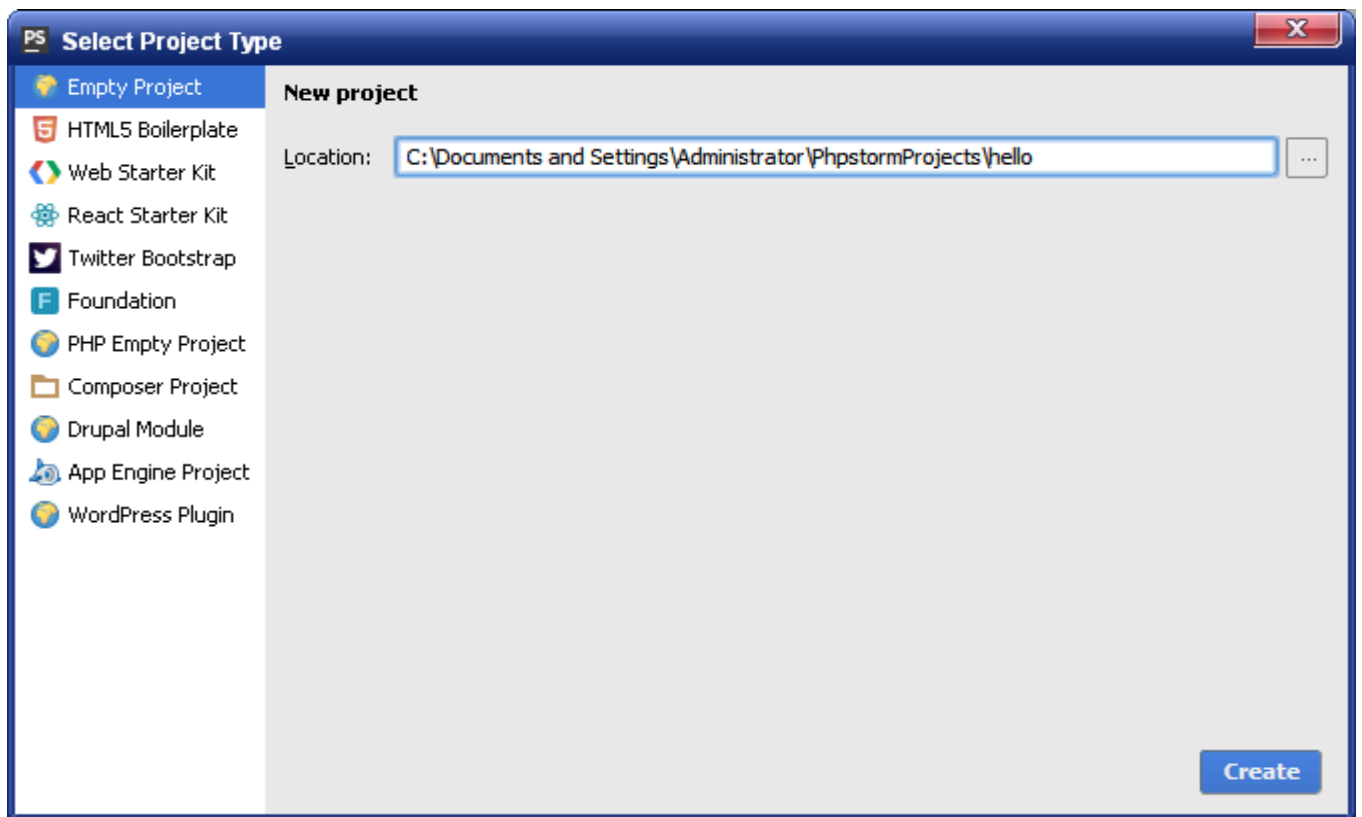


图 3-15 新建工程

(1) 新建、编写、调试 html 文件：

在项目中创建 html 文件：

右键点击 hello 项目，选择 New，单击 HTML File, 指定 html 文件名称

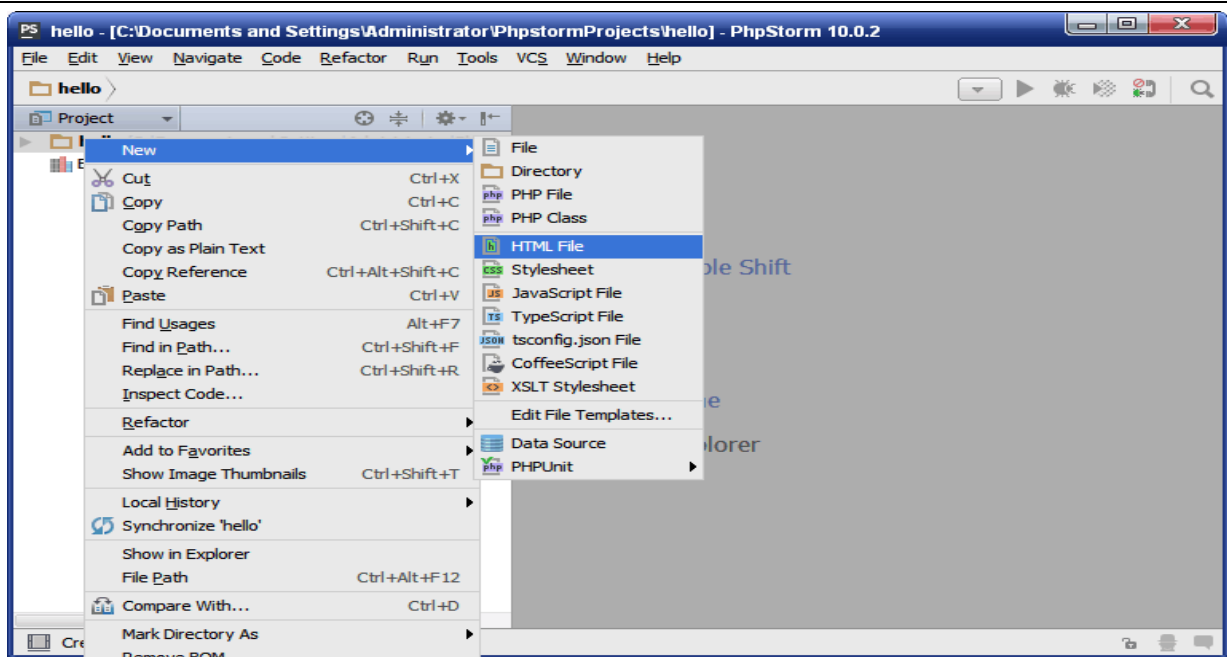


图 3-16 新建 HTML 文件

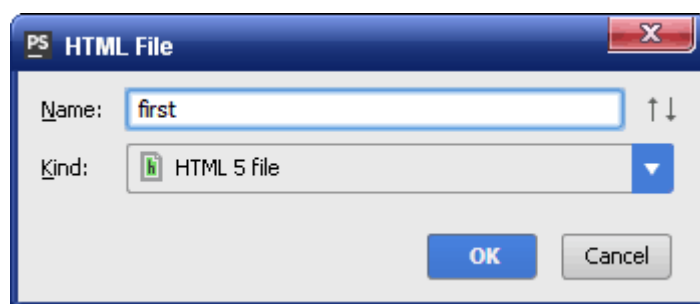


图 3-17 新建 first.html 文件

在 first.html 文件中编写 Hello World 程序：

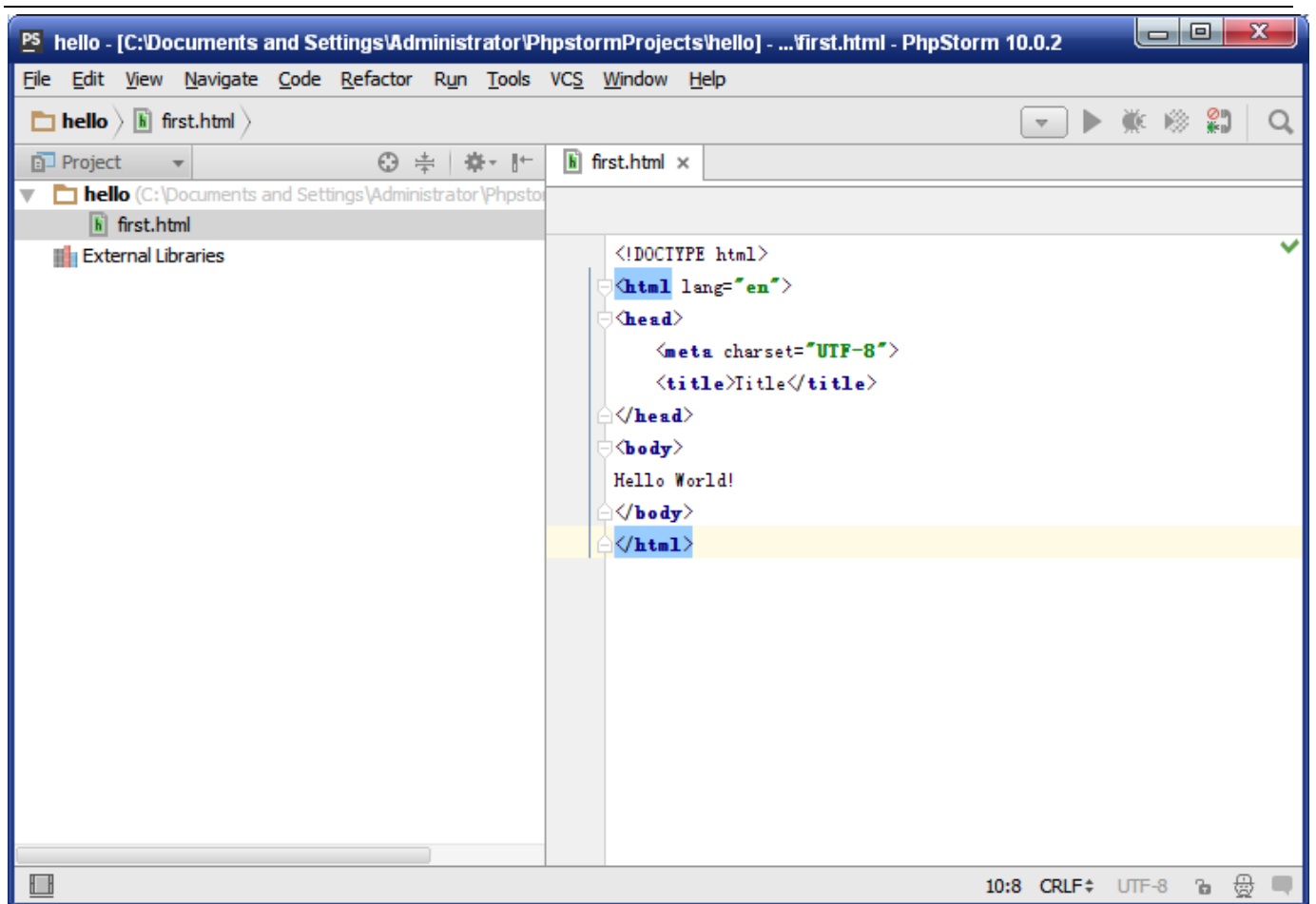


图 3-18 修改 first.html 文件

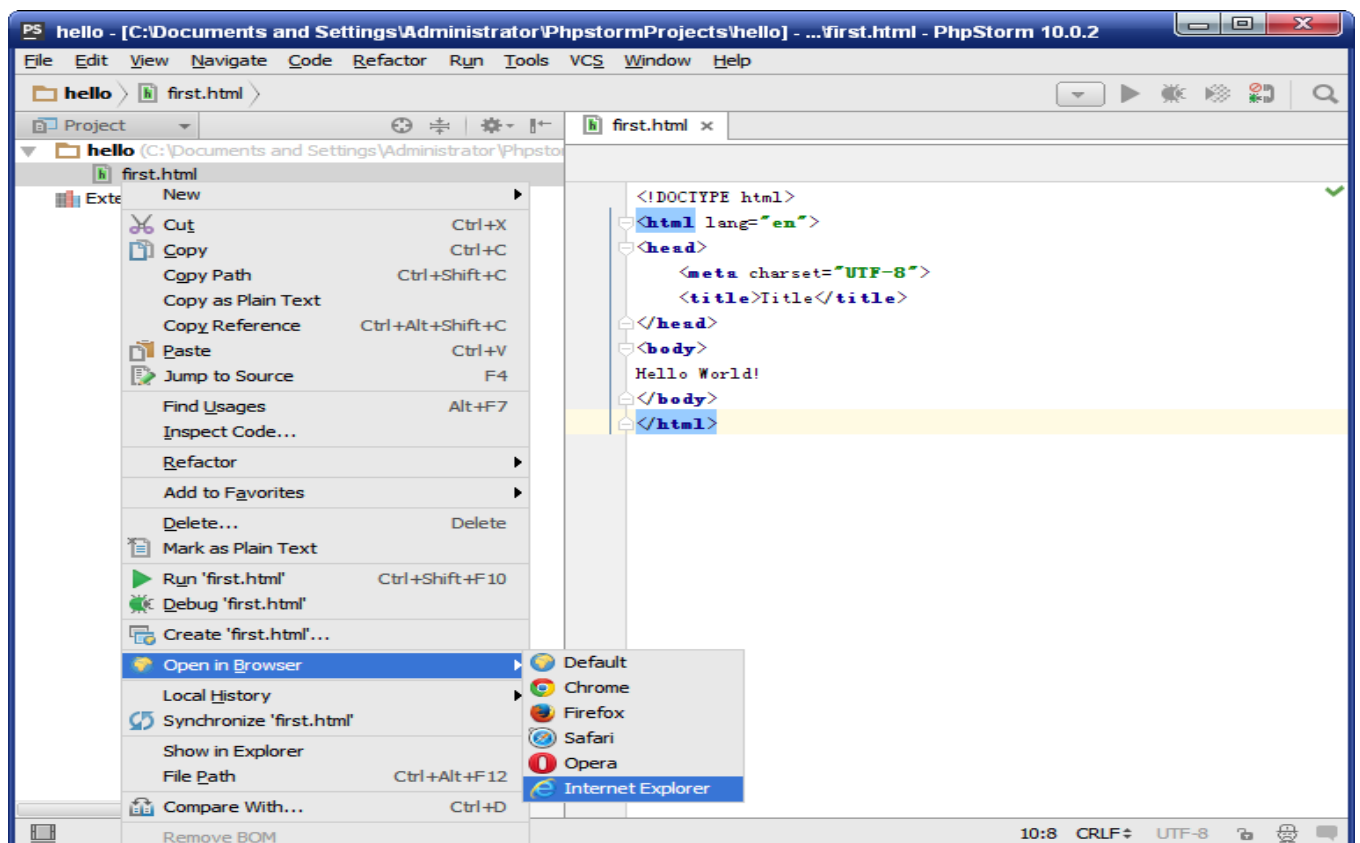


图 3-19 查看程序运行结果

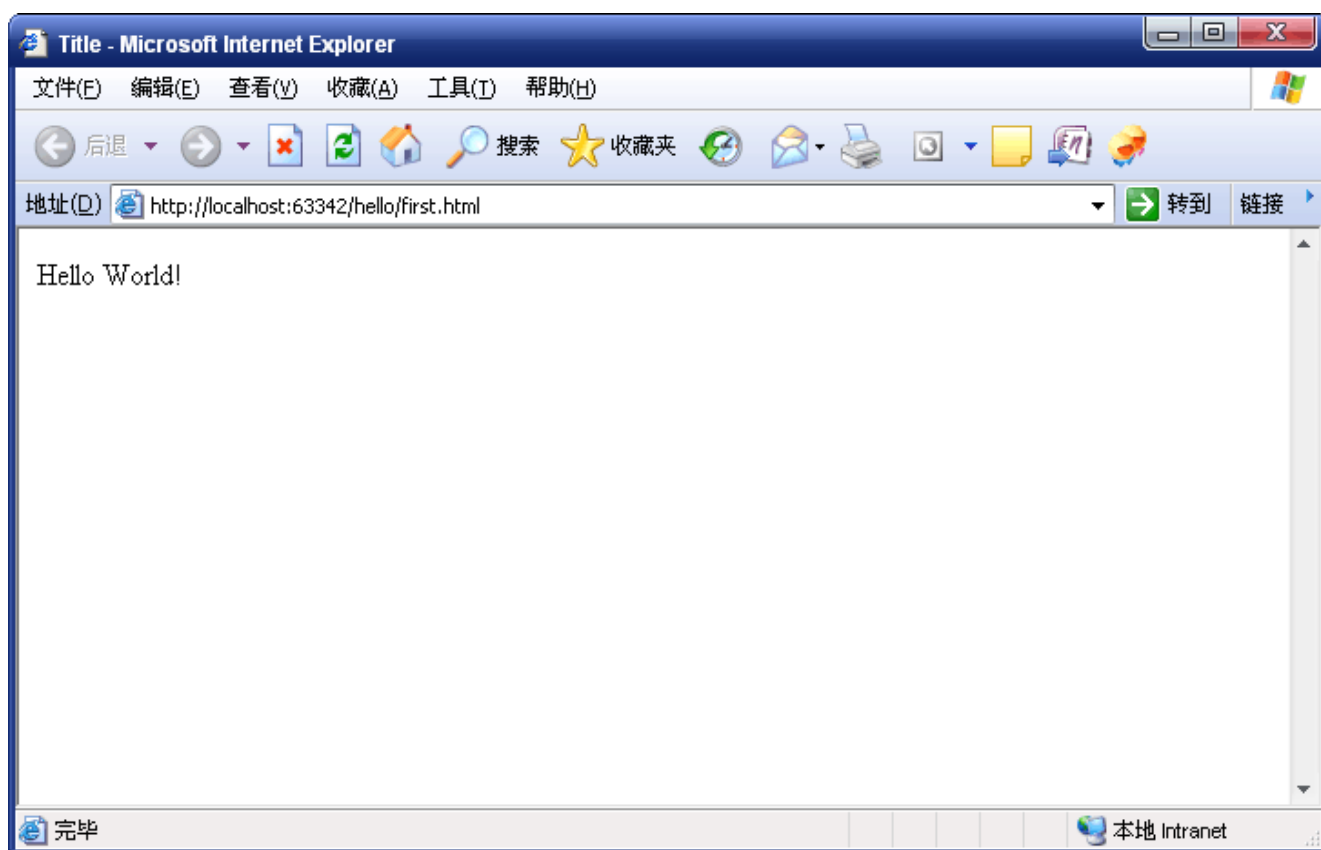


图 3-20 运行结果

(2) 新建、编写、调试 Php 文件：

在项目中创建 php 文件：右键点击 HelloWorld 项目，选择 New，单击 PHP File，指定 php 文件名。

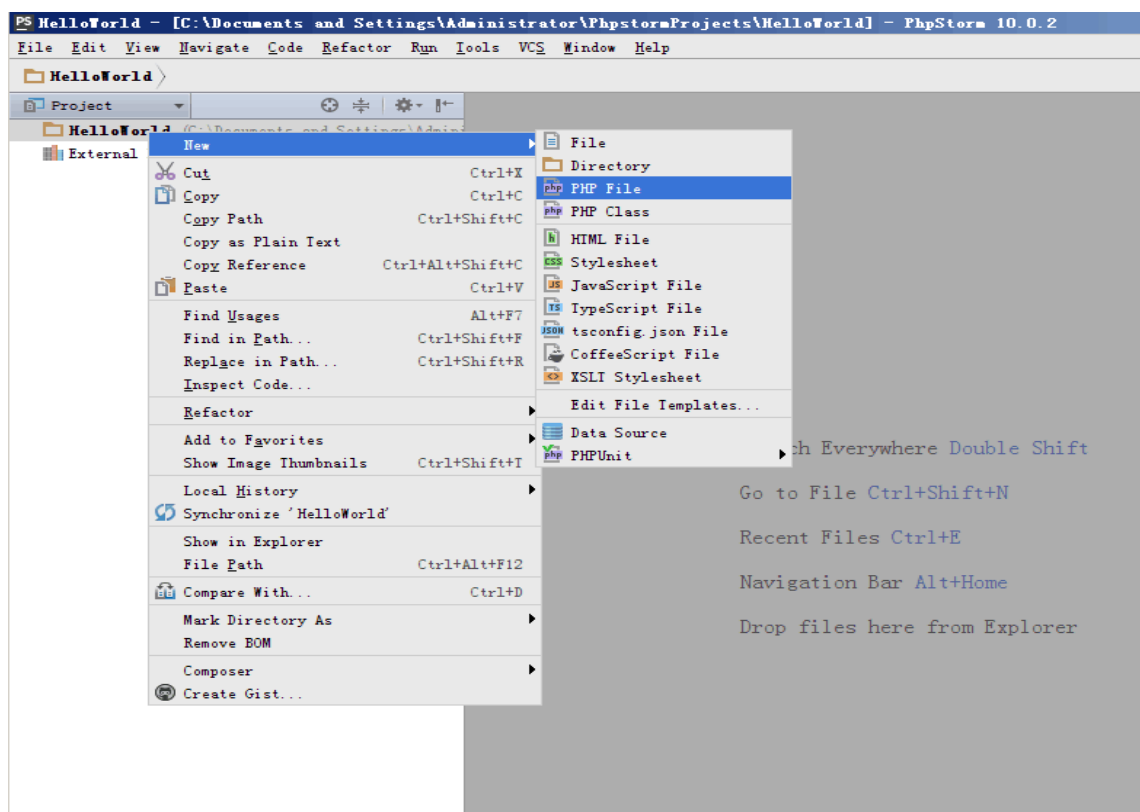


图 3-21 新建 PHP File

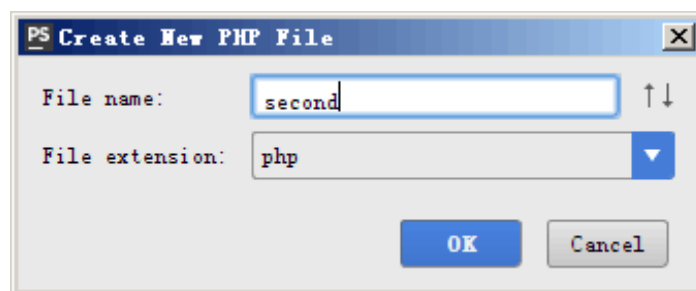


图 3-22 新建 second.php 文件



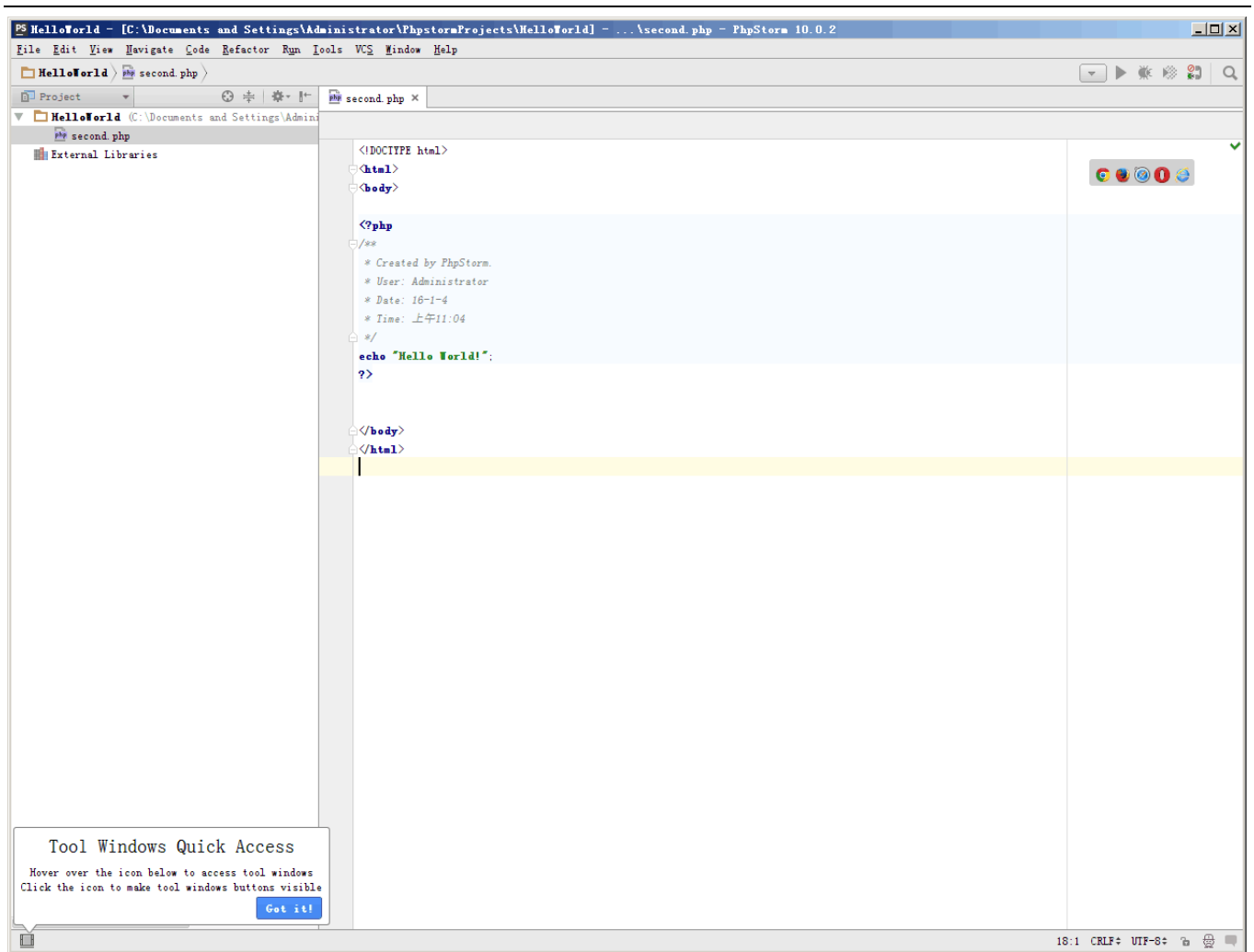


图 3-23 编写 php 文件

编写完成后，直接点击右上角网页符号，在网页上运行 php 程序，会出现如图 22 的情况：

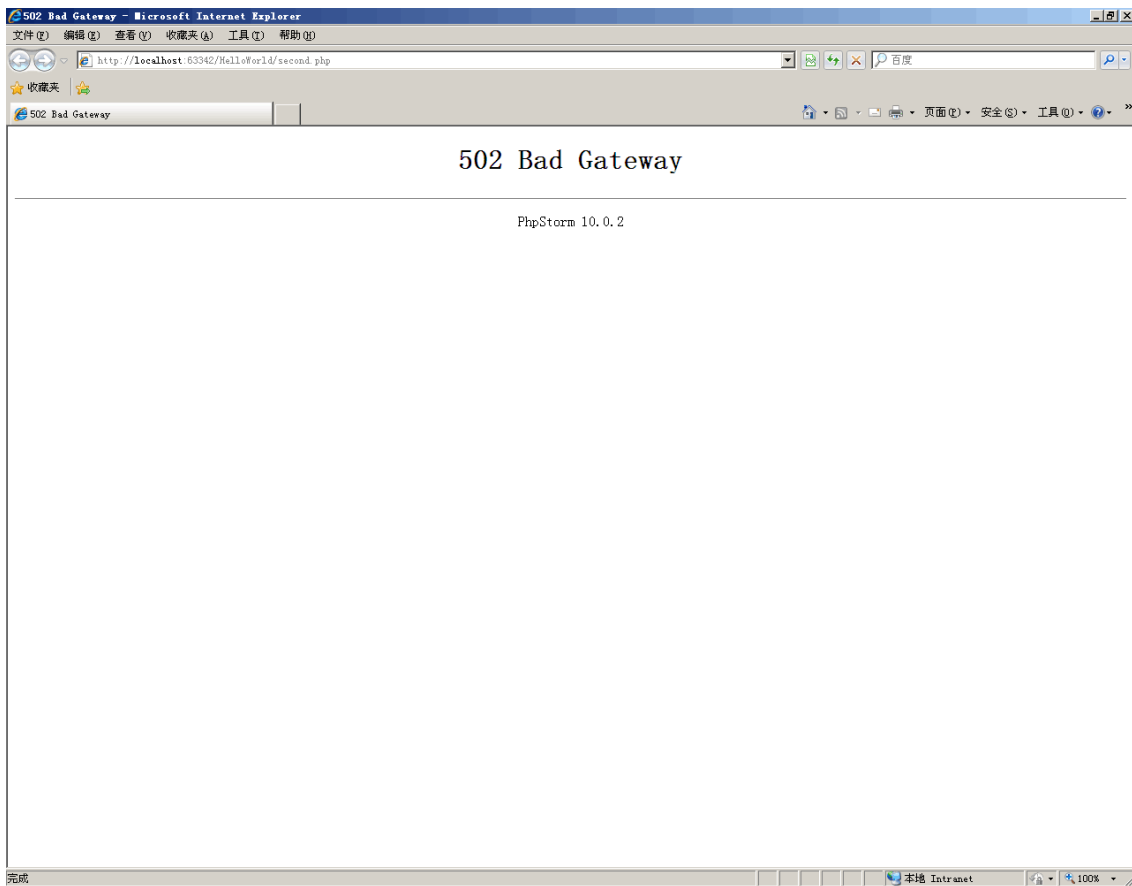


图 3-24 运行结果图



图 3-25 运行失败原因

要使 php 程序正常运行，还要设置一下 File 菜单下的 settings 选项下的 php Interpreter，选择 Interpreter 后的目录按钮，添加下载的 php 解压缩文件夹下的 php.exe 文件即可正常运行。

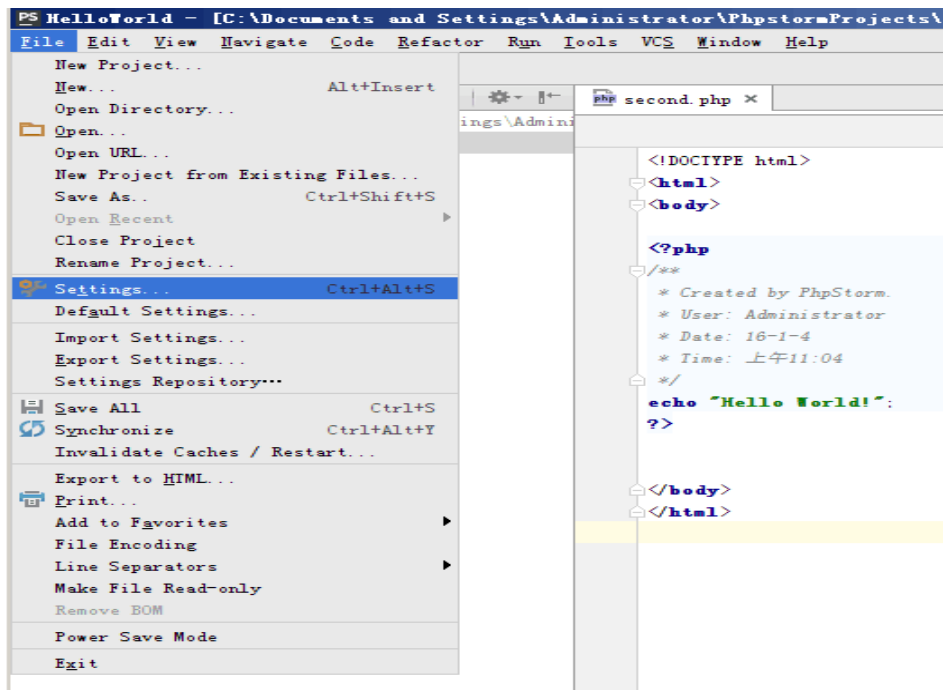


图 3-26 修改 setting 1

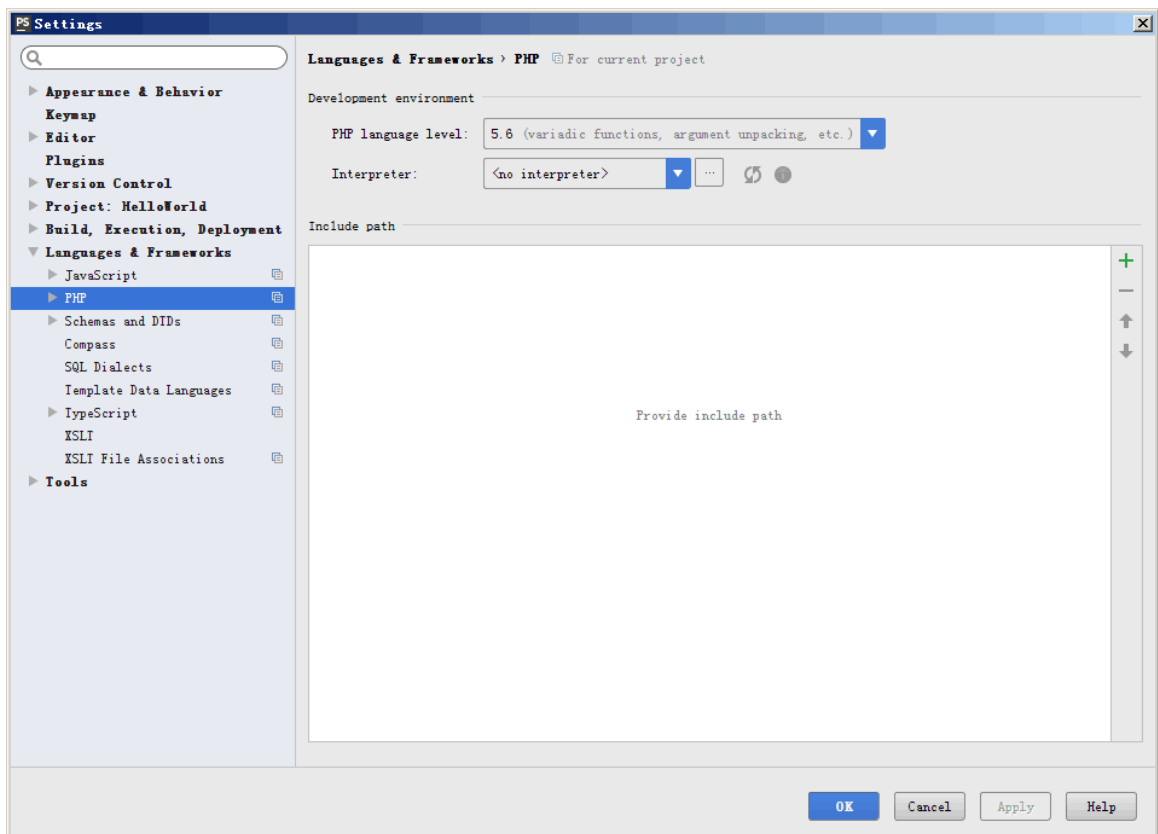


图 3-27 修改 setting 2

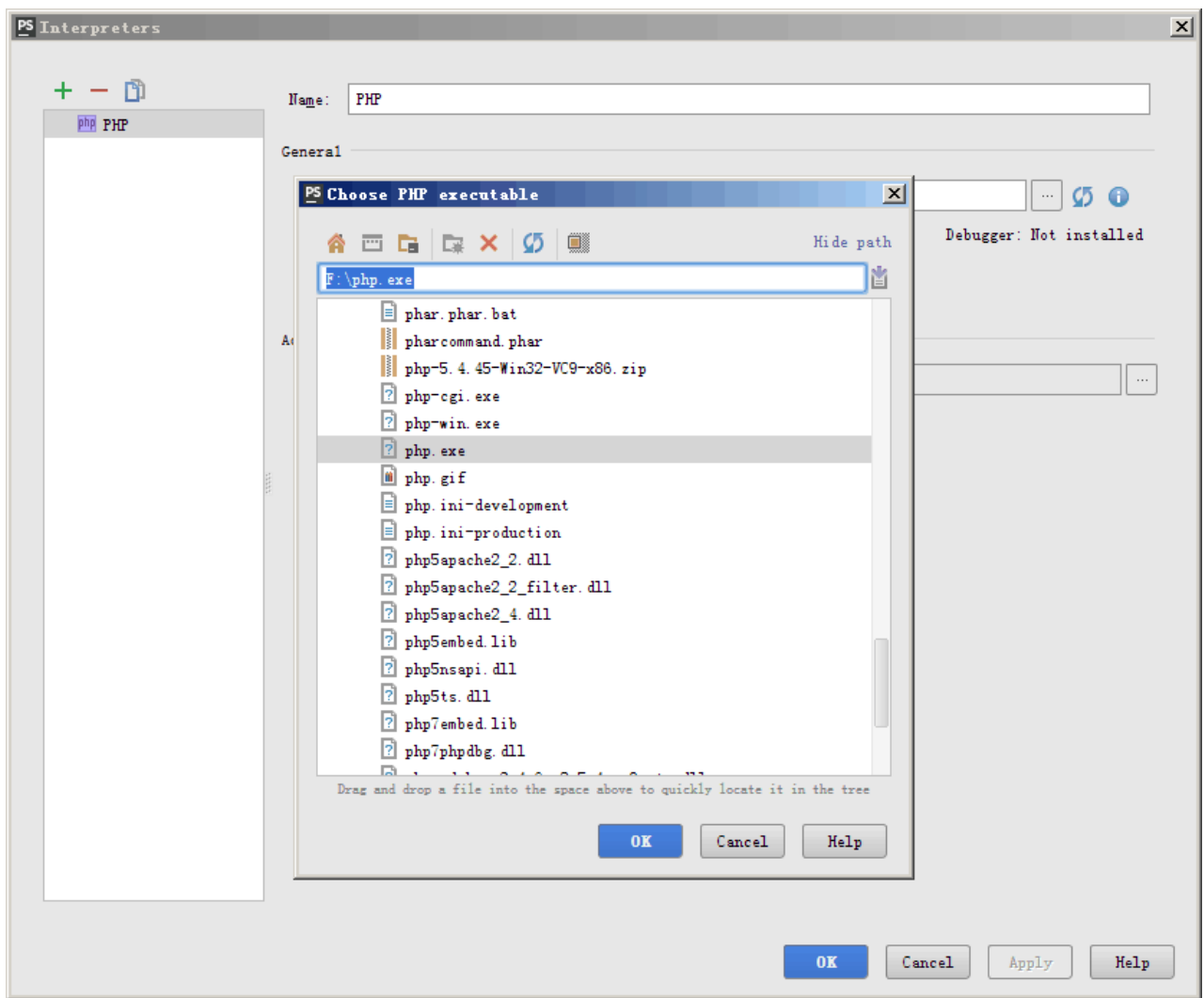


图 3-28 修改 setting 3

点击右上角网页符号

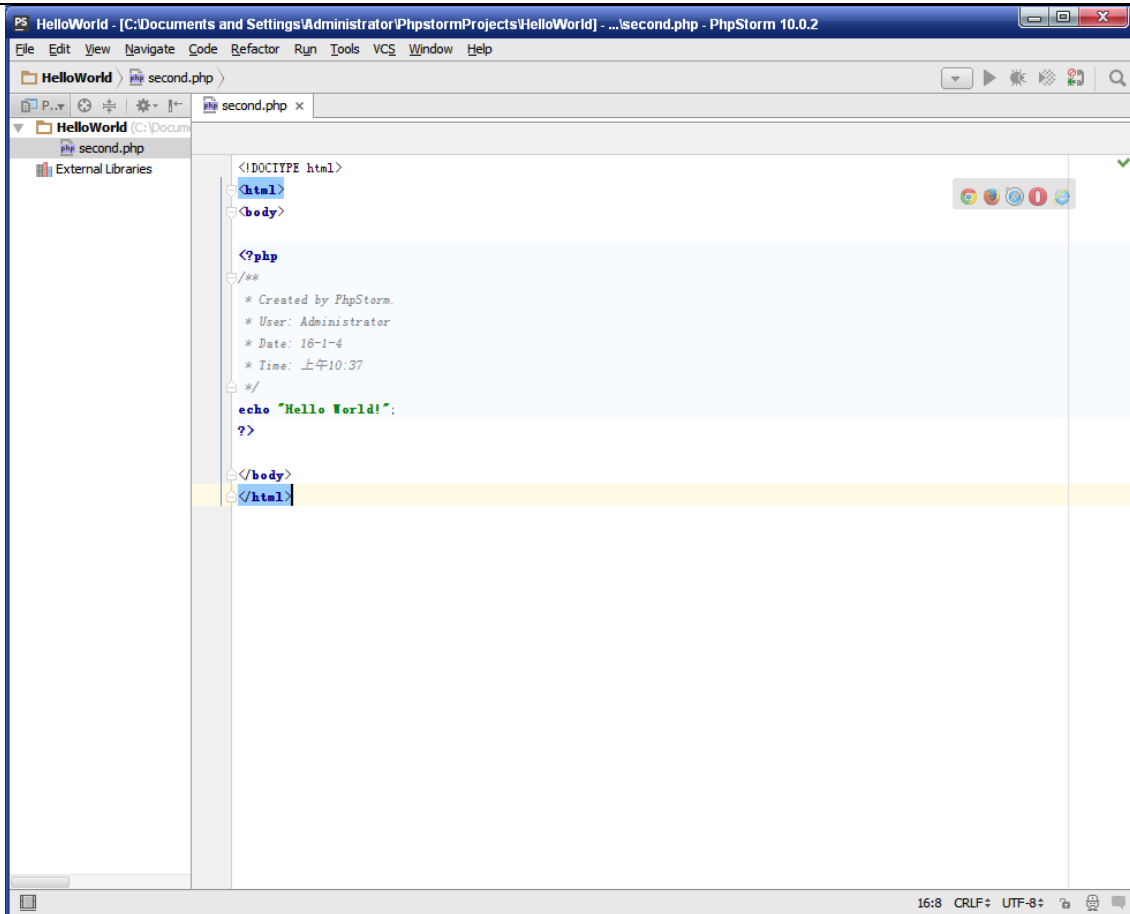


图 3-29 运行前

在网页上运行 php 程序如下：

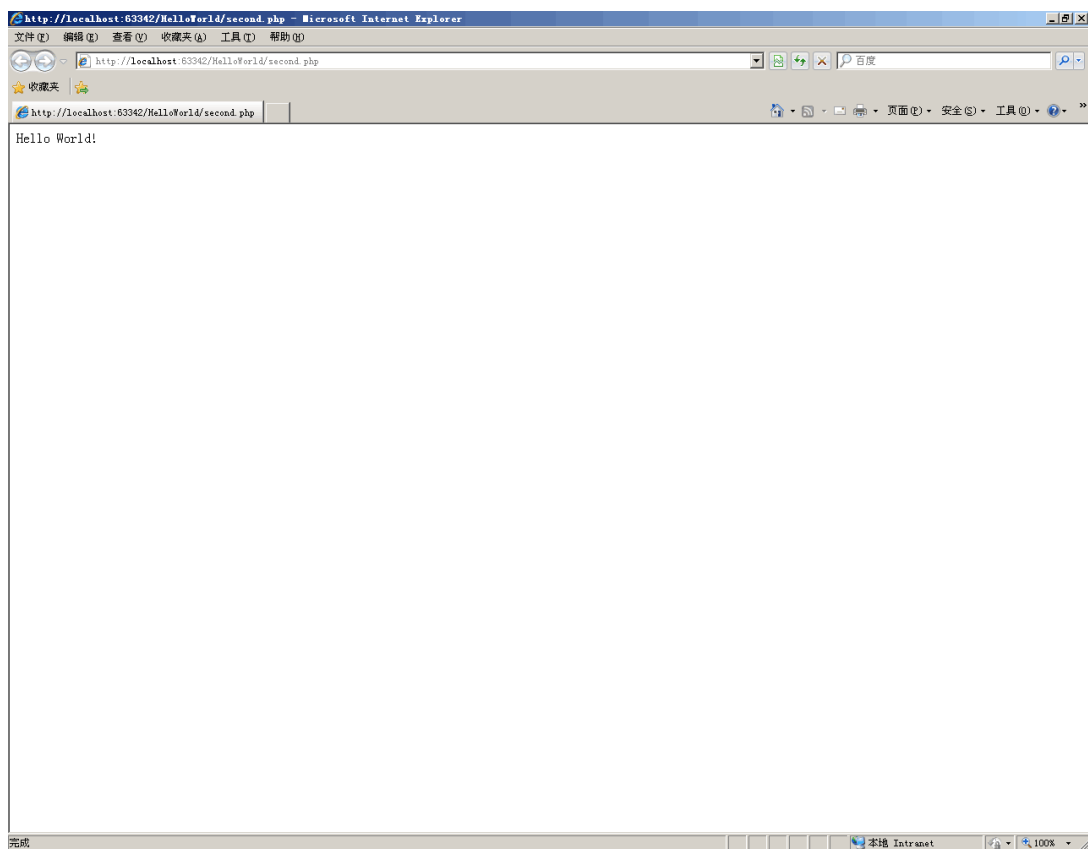


图 3-30 正常运行界面

在要开始调试的那一行的最开头点击一下，会出现一个红点，表示从这里开始调试。点击上面工具栏中的绿色虫子，开始调试。开始调试后程序会执行到你要调试的这一行，便停止，要继续向下调试，需要点击最上面的逐过程，就会向下一行执行。调试过程中可以这样查看各个变量的值。将鼠标放在你想要知道其值的变量上，程序就会自动提示此时值为多少。这样便于了解调试过程中每时每刻变量的值是多少， 执行到程序的最后一行，程序自动终止。

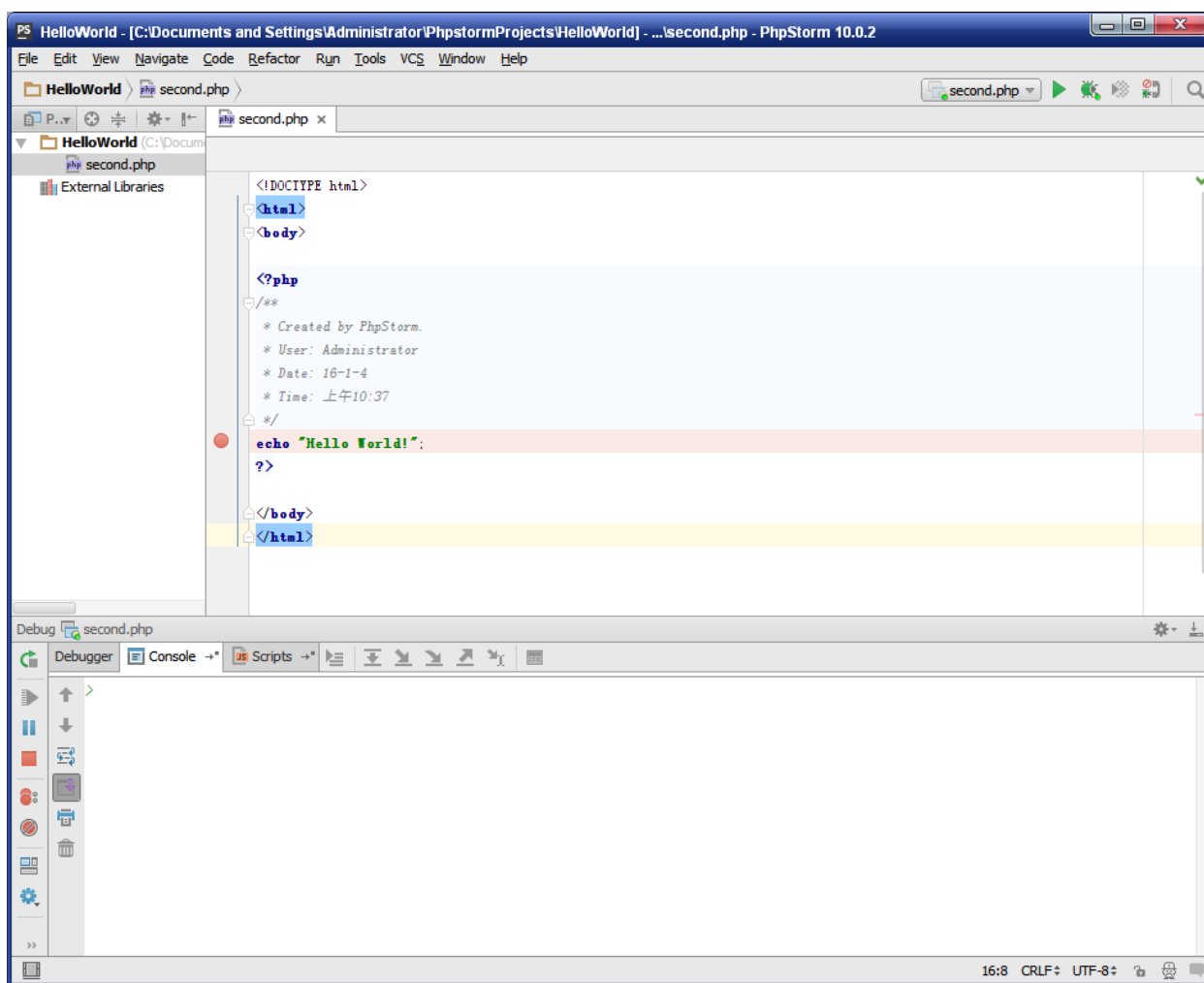


图 3-31 断点调试

### 3.3 C#

C#（读做 C sharp）是一种安全的、稳定的、简单的、优雅的，由 C 和 C++ 衍生出来的面向对象的编程语言。是微软公司发布的一种面向对象的于 .NET Framework 之上的高级程序设计语言。

在当前的软件开发行业中，C# 已经成为决定的主流语言，可以和 Java 平分天下。C# 作为一个全新的编程语言，可以实现大多数编程者需要的功能。包括 Windows 桌面应用程序、Windows 服务程序、Web 应用程序、Web 服务程序等。支持它的集成开发环境包括 Visual Studio .NET (VS)，我选的集成开发环境是 Visual Studio 2008，它支持 .NET 框架 3.5。

#### 3.3.1 C# 语言基础

(1) 类：

---

一个基本的 C# 类中包含数据成员、属性、构造器和方法。属性可以是静态或实例成员。

在 C# 中类的声明与 C++ 和 Java 很相似。但是，不像 C++，C# 结构体与类不支持继承。但是，与 Java 相同的是，一个结构体可以实现接口 (interface)。Java 的关键字 `import` 已经被替换成 `using`，它起到了同样的作用。

类可以是抽象的和不可继承的：一个被声明成 `abstract` 的类不能被实例化，它只能被用做一个基类，C# 关键字 `lock` 就像 Java 关键字 `final`，它声明一个类不是抽象的，但是它也不能被用做另一个类的基类界面：就象在 Java 中一样，一个界面是一组方法集合的抽象定义。当一个类或结构体实现一个界面的时候，它必须实现这个界面中定义的所有方法。一个单一的类可以实现几个界面，也许以后会出现一些微妙的差别，但是这个特点看起来与 Java 相比没有变化。

在 Java 中，一个接口是一组方法集合的抽象定义。当一个类或结构体实现一个接口的时候，它必须实现这个接口中定义的所有方法。一个单一的类可以实现几个接口，也许以后会出现一些微妙的差别，但是这个特点看起来与 Java 相比没有变化。

两个基本类：一个名叫 `object` 的类是所有其他类的基类。而一个名叫 `string` 的类也象 `object` 一样是这个语言的一部分。作为语言的一部分存在意味着编译器有可能使用它，无论何时在程序中写入一句带引号的字符串，编译器会创建一个 `string` 对象来保存它。

## (2) 布尔运算：

条件表达式的结果是布尔数据类型，布尔数据类型是这种语言中独立的一种数据类型。从布尔类型到其他类型没有直接的转换过程。布尔常量 `true` 和 `false` 是 C# 中的关键字。

## (3) 错误处理：

像 Java 中那样，通过抛出和捕捉异常对象来管理错误处理过程。

## (4) 内存管理：

由底层 .NET 框架进行自动内存垃圾回收。

## (5) 接口：

接口是其他类型为确保它们支持某些操作而实现的引用类型。接口从不直接创建而且没有实际的表示形式，其他类型必须转换为接口类型。

一个接口定义一个协定。实现接口的类或结构必须遵守其协定。接口可以包含方法、属性、索



---

引器和事件作为成员。

#### (6) 强类型:

C#是一个强类型的语言, 它的数值类型有一些可以进行隐式转换, 其他的必须显式转换, 隐式转换的类型只能是长度短的类型转换成长类型, int 可以转换成 long、float、double、decimal, 反之必须显式的转换。

#### (7) 编译:

程序直接编译成标准的二进制可执行形式. 但 C#的源程序并不是被编译成二进制可执行形式, 而是一种中间语言(IL), 类似于 JAVA 字节码。

#### (8) 预编译:

C# 中存在预编译指令支持条件编译, 警告, 错误报告和编译行控制. 可用的预编译指令有: #define, #undef, #if, #elif, #else, #endif, #warning, #error, #line。

没有了#include 伪指令, 无法再用#define 语句对符号赋值, 所以就不存在源代码替换的概念——这些符号只能用在#if 和#elif 伪指令里. 在#line 伪指令里的数字 (和可选的名字) 能够修改行号还有#warning 和#error 输出结果的文件名。

#### (9) 操作符重载:

一些操作符能够被重载, 而另一些则不能. 特别的是, 没有一个赋值运算符能够被重载. 能够被重载的单目操作符是: + - ! ~ ++ -- true false 能够被重载的二元运算符是: +、-、\*、/、%、&、|、^、<<、>>、==、!=、>、<、>=、<=。

#### (10) 类型:

C# 中的类型一共分为两类, 一类是值类型, 一类是引用类型。值类型和引用类型是以它们在计算机内存中是如何被分配的来划分的。值类型包括结构和枚举, 引用类型包括类、接口、委托等。还有一种特殊的值类型, 称为简单类型, 比如 byte, int 等。

基本数据类型: C#拥有比 C/C++或者 Java 更广泛的数据类型. 这些类型是 bool、byte、ubyte、short、ushort、int、uint、long、ulong、float、double 和 decimal, 像 Java 一样, 所有这些类型都有一个固定的大小。又像 C 和 C++一样, 每个数据类型都有有符号和无符号两种类型。与 Java 相同的是, 一个字符变量包含的是一个 16 位的 Unicode 字符, C#新的数据类型是 decimal

数据类型，对于货币数据，它能存放 28 位 10 进制数字。

(11) 注释：

行注释使用 “//” 表示，块注释使用 “/\*……\*/” 表示，文档注释使用 “///” 表示。

### 3.3.2 集成开发环境 Visual Studio 2008 的安装

到官网下载 Visual Studio 2008

### 3.3.3 集成开发环境的使用

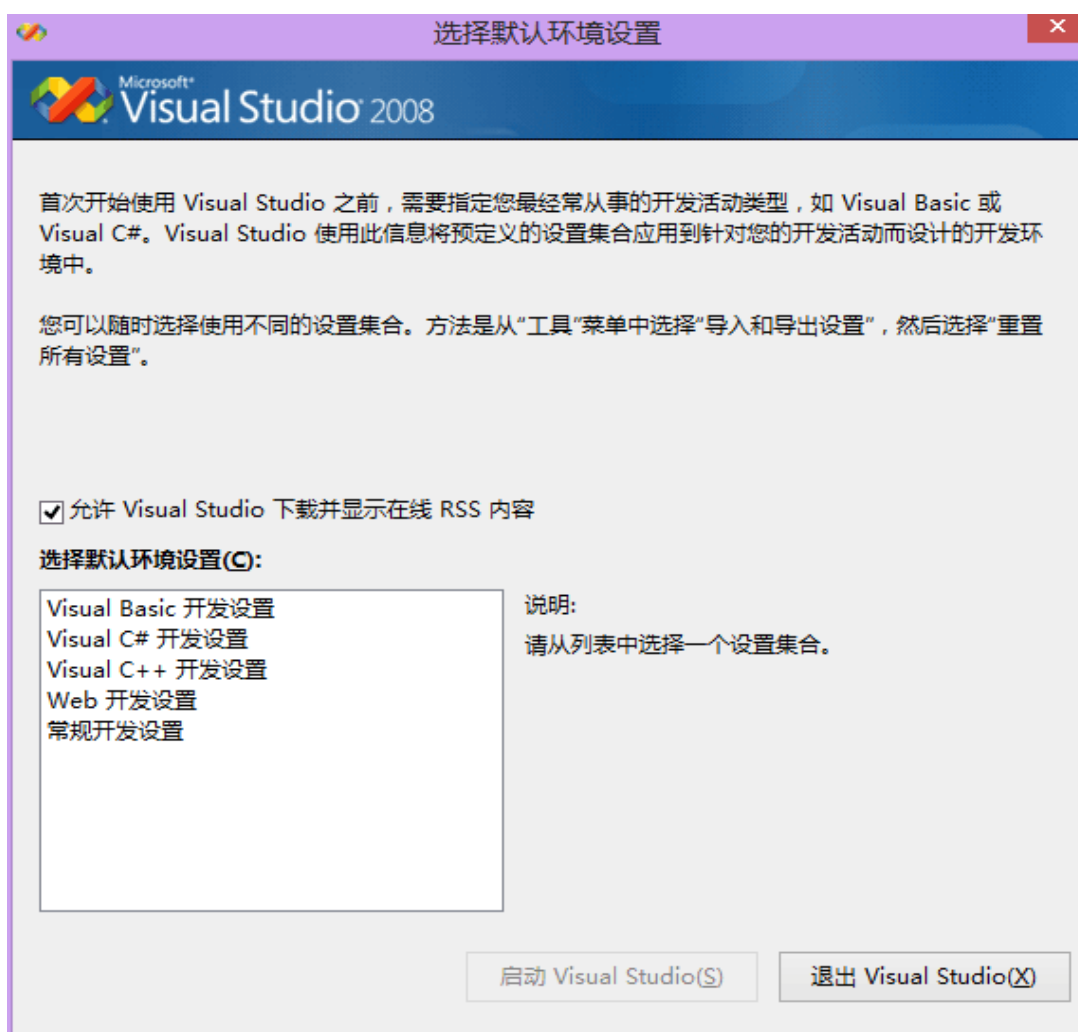


图 3-32 选择 C#



图 3-33 主界面

选择文件菜单下的新建，点击项目，选择 Windows 窗体应用程序，输入项目名称，选择项目地址，新建成功。

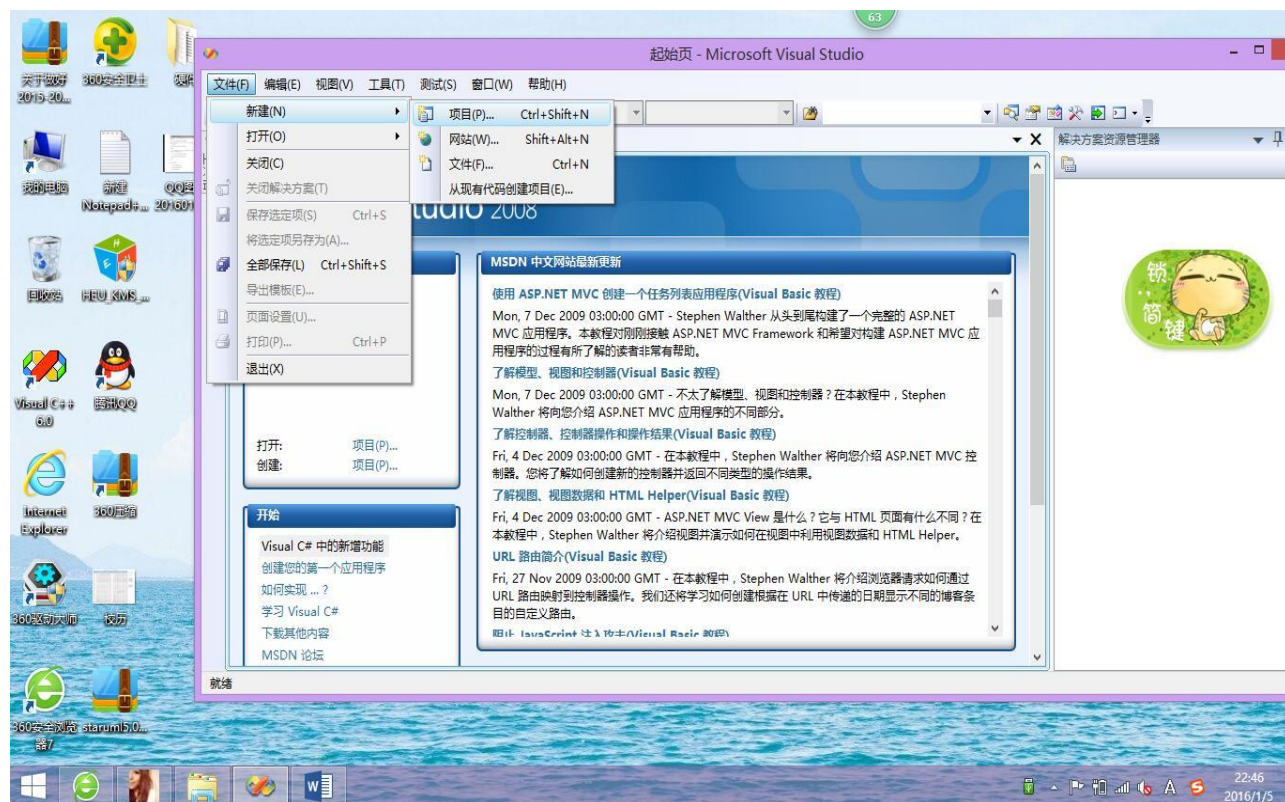


图 3-34 新建项目 1

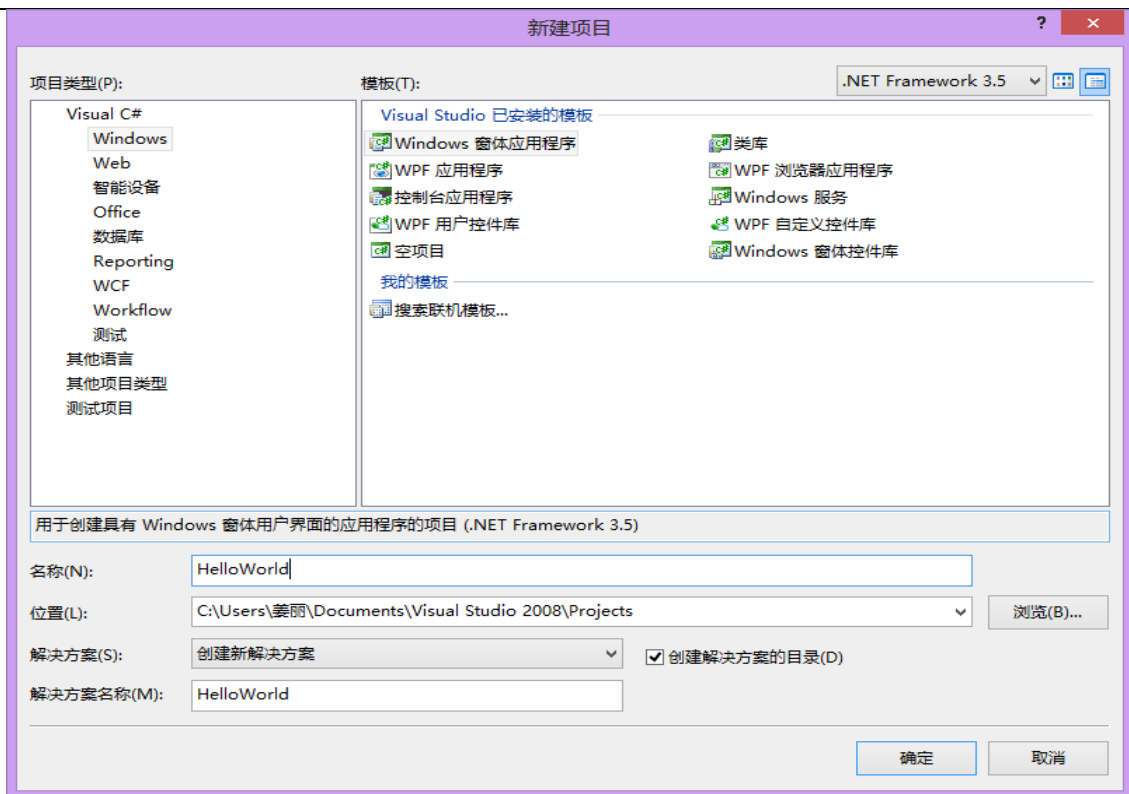


图 3-35 新建项目 2

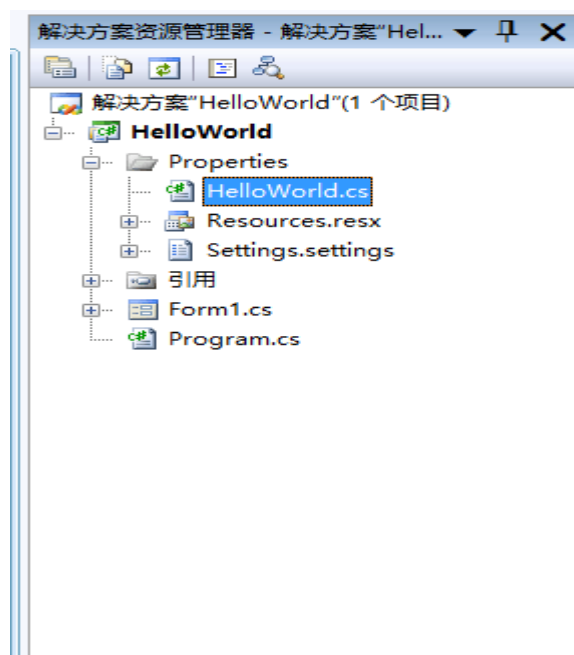


图 3-36 新建项目 3

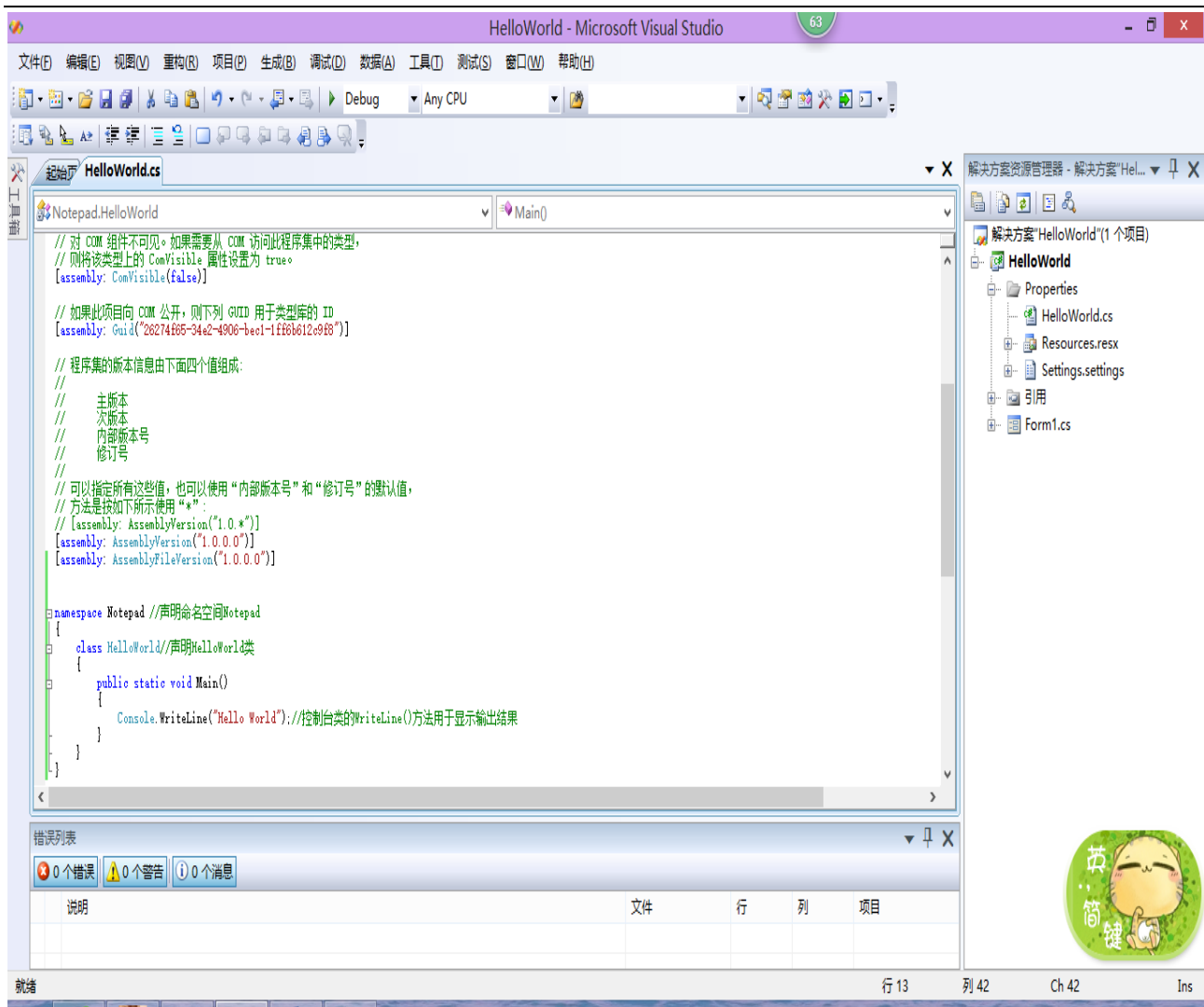


图 3-37 编写程序

### 3.3.4 集成开发环境的调试技术

在菜单栏中选择生成->生成解决方案命令，如果程序没有错误，那么在窗口下方就没有显示错误与警告，状态栏中会显示生成成功，说明程序编译成功，可以运行了。

编译成功的程序，可以直接选择菜单栏中的调试->启动调试命令，或者单击工具栏中的调试按钮，就可以启动调试功能。

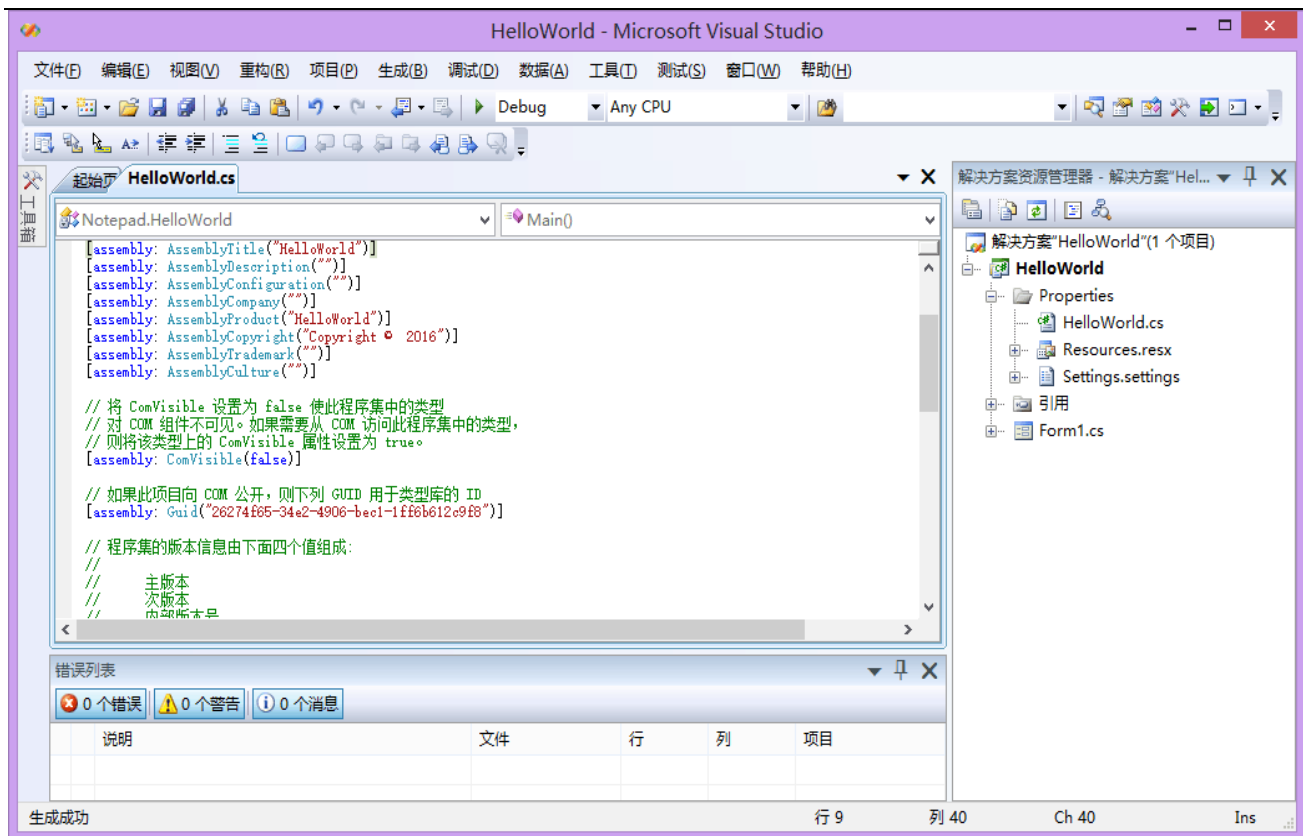


图 3-38 编译程序

单击想设断点的位置就可以设置断点，设置断点后调试即可查看变量的名称和值，排查错误。

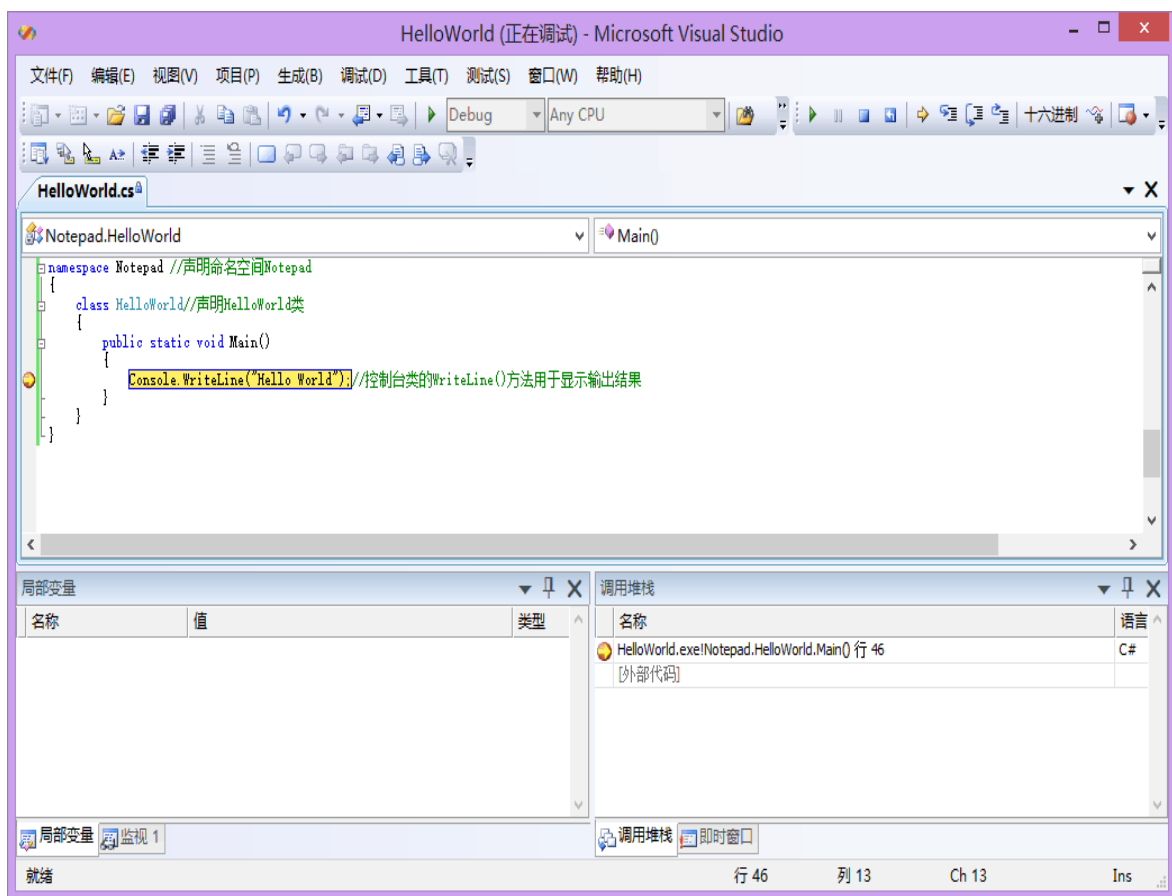


图 3-39 设置断点

---

## 4 实验小结

通过本次试验，我熟悉了 PHP、.Net、JAVA 语言的编码规范，学习了 PHP 集成编程环境 PhpStorm、.Net 集成编程环境 Visual Studio 2008、JAVA 集成编程环境 Eclipse 的基本编程方法和调试方法，受益匪浅。