

如何学习习语言编程

播报文章

原创 | 浏览：1585 | 更新：2016-10-07 17:47

诞生于上世纪70年代的C语言是一门古老的语言了,但作为一门底层语言,时至今日它仍然非常强大。而习语言是一种新的中文编程语言,基于C语言语法扩充中文支持的编程语言。下面就来看看如何开始用习语言编程吧。

声明本文是由本人将“如何学习C语言编程”C语言翻译成习语言和修正,而且原文的C语言有部分错漏,或也有部分和习语言不一样导致错误

366 如何学习C语言编程

工具/原料

- 习语言
- Microsoft Visual Studio Express

部分 1: 准备工作

- 1
- 学习前你首先要有习语言和一个编译器。

理解基本概念。 C语言是一门古老的语言,但却十分强大。它最初是为Unix操作系统设计的,但后来被移植到了几乎所有的操作系统上,并得到了很多扩展。C语言的现代版本是C++。

C语言本质上是由函数构成的,在函数体中你可以使用变量,条件语句,循环等语句来存储和处理数据。
- 2
- 查看一些基本的代码。 下面是用C语言写的一段非常基本的代码,阅读这些代码,尝试理解这种语言的不同部分是如何工作的,并对程序的运作原理有初步的认识。 "

#包含“习语言系统.接口”

整数类型 主函数()

```
{  
    格式输出("你好,世界!%行");  
    暂停();  
    返回 0;  
}
```

"[1]

这里的#包括“习语言系统.接口”指令在程序开始之前就出现了,它的功能是把包含有你需要的函数的库加载进来。在这个例子中,引入“习语言系统.接口”使得我们能够使用格式输出() 和 暂停() 这两个函数。

这里的主函数() 指令会告诉编译器,程序需要运行一个叫做“主函数”的函数,该函数运行完毕后返回一个整数值。所有的C语言都要运行一个“主函数”函数。

“{}”符号表示括号内的所有内容都是函数的一部分。在本例中,他们标记了所有的内容都是“主函数”函数的一部分。

格式输出() 函数能够将小括号中的内容显示在用户的屏幕上。双引号保证了这个字符完全按照字面的样子输出,%行 组合告诉编译器这串字符输出完之后将光标移动到下一行。

“;” 符号表示一行的结束。绝大部分C代码都以分号结束,习语言也一样。

“暂停()”函数告诉编译器这段程序要等待一个按键的输入才能够继续。考虑到大部分编译器在运行完程序之后会立刻关掉程序窗口,这个功能还是很有用的,因为这样可以让程序保持运行直到有键被按下才会结束。

“返回 0”指令表示程序的结束。请注意“主函数”函数是一个 整数类型 的函数,也就是说当函数结束时需要返回一个整数。如果返回0则表示程序正确的执行了,其他数字表示程序运行时发生了错误。

回季计划

今日支出3757元

写经验 有钱赚 >>

陈捷联

已关注

作者的经验

win10卸载自带预装应用软件

windows10卸载自带预装应用软件

Baidu经验

游戏延迟PING值

By 小怪杂谈



聊天机器人

如要投诉,请到百度经验投诉中心,如要提出意见、建议,请到百度经验管理吧反馈。

3

尝试编译这段程序。把上面的代码输入到你的代码编辑器中,然后保存为"*.习"文件。用你的编译器编译它, 点击"控制台编译连接"按钮即可。

注意编辑器是不带编译器的。我们一般用微软的Microsoft Visual Studio Express放进习语言来编译我们写的程序!

4

要养成写注释的习惯。注释是代码中的一部分,它不会被编译,但是却可以告诉你代码做了些什么。这对于提醒你自己的代码是干什么的以及让其他开发者理解你的代码都很有帮助。

在习语言中添加注释只需要把要注释的部分的前面添加【, 后面添加】。

不要吝啬你的注释,除了特别简单明了的地方都尽量加上注释吧。

注释功能也可以快速的屏蔽一部分代码但不删除它们。只需要给你想要排除的代码用注释标签包起来它们就不会被编译。如果你想要改回来,去掉注释标签即可。

END

部分 2: 变量的使用

1

理解变量的功能。变量是用来存储数据的,不管是计算得出的还是用户输入的数据。变量在使用前要先定义,并且有不同的类型可以选择。

有以下几种常见的变量类型 整数类型, 字节类型, 和 浮点数类型。每种变量类型都代表一种数据存储的格式。

学习声明变量。变量在使用前要先被创建出来,或者叫"声明"。声明一个变量只需要在变量类型的后面写出变量的名字即可。比如,下面就是一些变量声明的例子:"

浮点数类型 x;

字节类型 name;

整数类型 a, b, 甲, 乙;

"

注意,你可以在一行中声明多个变量,只要它们的类型是一样的就行,你只需用逗号把变量名隔开即可。

和大多数的C代码一样,变量的声明也要以分号结尾。

理解在何处声明变量。变量的声明必须要放在每个代码块之前(代码块是指用大括号{}包起来的一段代码)。如果你在代码块后边声明变量,程序就不能正确执行了。

用变量来存储用户的输入。现在你了解到了变量的一些基本原理, 你可以写一段简单的程序来存储用户的输入。这次你需要用到另外一个叫scanf的函数, 它的功能是把用户的输入赋值给指定的变量。"

#包含 "习语言系统.接口"

整数类型 主函数 ()

{

整数类型 甲;

格式输出("请输入一个数:");

格式输入("%整数类型", &甲);

格式输出("你输入了: %整数类型", 甲);

暂停();

返回 0;

}

"

这里的""%整数类型"符号告诉 格式输入 函数在用户的输入中找出整数。

"甲"前面的 & 符号告诉 格式输入 在哪里找到要修改的变量, 并把输入的整数值存进去。

最后的 格式输出 命令读出输入的整数并返回给用户。

处理变量。你可以用数学表达式来处理之前存储的变量。需要注意一个重要的差别: 在数学表达式中单个=是赋值号,作用是把等号右边的值赋给等号左边的变量, 而==则是比较两个变量是否相等。"

甲 = 3 * 4; 【把 甲 设为3*4,也就是12】

甲 = 甲 + 3; 【把 甲 的值增加3,然后把新的值赋值给 甲】

甲 赋值 甲 + 3; 【把 甲 的值增加3,然后把新的值赋值给 甲】

甲 == 15; 【检查x是否等于15】

甲 等于 15; 【检查x是否等于15】

甲 < 10; 【检查x是否小于10】

"

END

部分 3: 使用条件语句

1

理解条件语句的基本概念。大多数程序都是由条件语句驱动的, 这样的语句可以判断一个条件是 真 还是 假, 然后据此执行不同的动作。最基本的条件语句是 如果 语句。

习语言中的 真、假 和你平常理解的有点不太一样。真和任何非0的数总是相等的。当你执行一个比较时,如果结果是 真,会返回一个"1"。如果结果是 假,会返回0。弄清楚这一点能帮助你更好的理解 如果 语句的执行过程。

2

学习基本的比较符号。条件语句是以比较大小的数学表达式为核心的。下面列出了最常用的一些比较符号:"

```
> /* 大于 */
< /* 小于 */
>= /* 大于或等于 */
<= /* 小于或等于 */
== /* 等于 */
!= /* 不等于 */
```

"

"

```
10 > 5 真
6 < 15 真
8 >= 8 真
4 <= 8 真
3 == 3 真
4 != 5 真
```

"

习语言可以使用中文: 大于、小于、大于等于、小于等于、等于、不等于。

注意 等于是==, 作为比较用, 不能作为=用, 赋值 才是=,

其实很多编程的很初学者都会搞混, 比较是否相等概念, 和赋值的概念!

我开始也, 其实怪小数学, 没有讲清楚比较和赋值概念, 然后就是初中的代数, 也没有这解释这两种=的不同。

3

写一个如果语句。利用IF语句,你可以根据一个表达式计算的结果决定之后的程序如何运行。之后学习了其他条件语句后你可以把它们组合起来实现更强大的功能,不过现在写一段简单的代码熟悉一下就行了。"

#包含 "习语言系统.接口"

整数类型 主函数 ()

```
{
    如果 (3 小于 5)
        格式输出("3比5小");
    暂停();
    返回 0
}
```

"

4

使用 否则 / 或者 语句来扩展你的条件判断。在IF语句中你可以添加 否则 和 或者 语句来处理更多不同的结果。否则后面的语句在 如果 中的判断结果为 假 时执行。或者 则可以让你在一个代码块中使用多个 如果 语句来处理更多的情况。阅读下面的代码看一下他们是怎么工作的。"

#包含 "习语言系统.接口"

整数类型 主函数 ()

```
{
    整数类型 岁数;
    格式输出("请输入您的年龄: ");
```

```

    格式输入( "%d", &岁数 );
    如果 ( 岁数 <= 12 ) {
        格式输出( "你是个孩子!\n" );
    }
    或者 ( 岁数 < 20 ) {
        格式输出( "年轻的感觉真好!\n" );
    }
    或者 ( 岁数 < 40 ) {
        格式输出( "你充满了青春的活力!\n" );
    }
    否则{
        格式输出( "充满智慧的年纪! \n" );
    }
    暂停();
    返回 0;
}

```

"[2]"

这段代码接收用户输入的一个数据然后传递给 如果 语句。如果这个数据满足第一个条件,则第一个 格式输出 被执行。如果没有满足第一个条件,则后面的各个 或者 会逐个进行判断直到有一个满足条件的分支为止。如果没有任何分支满足条件,则 否则 语句被执行。

END

部分 4: 学习循环语句

1

理解循环的原理。循环是编程中很重要的一部分, 它们让你可以重复执行一段代码直到满足特定条件为止。这个机制使你很容易的实现重复的动作,同时省去了每次做条件判断的麻烦。

有3种类型的循环:

步进循环(循环变量赋初值; 循环条件; 循环变量增量){语句}、

当(条件表达式) 重复{语句}

重复 {语句}当(条件表达式);和重复 {语句}直达(条件表达式);

2

使用 步进循环 。这是最常见和好用的循环类型。它会不断的运行循环内的函数直到循环条件不再成立。步进循环 需要包含3条语句:初始化变量,循环条件,和变量更新的方式。如果你不需要其中的某个语句,把该处空着打一个分号即可,否则的话循环会无限运行。[3]"

#包含 "习语言系统.接口"

整数类型 主函数 ()

```

{
    整数类型 甲;
    步进循环(甲 赋值 0 ; 甲 小于 15 ; 甲 ++){
        格式输出("%整数类型%行", 甲);
    }
    暂停();
    返回 0 ;
}

```

在上面的程序中,甲 被设为0,循环继续运行的条件是 甲 小于15。每次循环中 甲的值被打印出来, 并且被增加1 (打印出来的0到14) 。一旦y=15,循环就结束了。

3

使用 当-重复 循环。当-重复 循环比 步进循环 要简单的多。它们只有一个语句,只要该语句为 真 循环就不断执行。你不需要更新变量,不过你可以在循环体中做这些事。 "

#包含 "习语言系统.接口"

整数类型 主函数()

```

{
    整数类型 甲;

```

```

甲 赋值 0;
当(甲 小于等于 15 )重复{
    格式输出("%整数类型%行", 甲);
    甲++;
}
暂停();
返回 0 ;
}
"

```

这个循环每执行一次 甲++ 命令就把 甲 的值增加1。一旦 甲 达到16,循环就结束了。(记住只有在 甲 小于等于15的条件下循环才会执行。)

4

使用 重复-当 循环。这种循环在你想要确保一个循环至少要被执行一次时非常管用。在 步进循环 和 当-重复 循环中,循环条件的检测是在循环开始之前进行的,这也就意味着有可能第一次检测就无法通过,那样的话循环体一次都不会被执行。然而“重复-当”循环会先执行一次循环体然后再做检测,这就保证了循环体至少会被执行一次。 ”

#包含 “习语言系统.接口”

```

整数类型 主函数()
{
    整数类型 甲;
    甲 赋值 5;
    重复{
        格式输出("%整数类型%行", 甲);
    }当(甲 不等于 5);
    暂停();
    返回 0 ;
}
"

```

在上面的循环中,即使循环条件检测的结果为 假 还是会展示一条信息。变量 甲 的值被设为5而当-重复 循环 被设置为只有当 甲 不等于5时才运行,所以循环执行到条件检测时就会终止。但信息还是被展示出来了,因为条件检测是在输出信息之后的。

“重复-当”循环中的 当 语句必须以;结尾。这是唯一一种循环体以分号结尾的情形。

END

部分 5: 使用函数

1

理解函数的基本原理。 函数是可以被程序的其他部分调用的自成一体的代码块。使用函数可使你更容易重复一段代码,同时也让程序变得简单易读、便于修改。函数中可以包含前面提到的所有技术,甚至可以包含其他函数。

前面的例子中的 主函数() 就是一个函数, 暂停()同样也是。

要想写出高效且易读的代码,函数是至关重要的。用好函数可以使你的程序条理更清晰。

2

从函数原型开始。在真正开始编写一个函数之前,你最好先搞清楚你要完成什么功能,并从函数原型开始编写。函数的基本语法格式为: “返回值类型 函数名 (参数1, 参数2, ...);”。比如下面是一个把两个数相加的函数: ”

```

整数类型 甲加乙 ( 整数类型 甲, 整数类型 乙 )
"

```

上面的代码创建了一个把输入的 甲 、 乙 相加然后返回他们的和的函数。

3

把函数添加到程序中。你可以用上面的函数原型实现一个把用户输入的两个数相加的函数。下面的程序展示了“add”函数是如何处理输入的数字的。 ”

#包含 “习语言系统.接口”

```

整数类型 甲加乙 ( 整数类型 甲, 整数类型 乙 );
整数类型 主函数()
{
    整数类型 甲;

```

```
整数类型 乙;
格式输出("请输入要求和的两个数:");
格式输入("%整数类型", &甲);
格式输入("%整数类型", &乙);
格式输出("您输入的数字之和为 %整数类型%行", 甲加乙(甲, 乙));
暂停();
返回 0;
}
整数类型 甲加乙(整数类型 甲, 整数类型 乙)
{
    返回 甲 + 乙;
}
"
```

请注意,函数的原型也需要放在程序的顶部,这样能保证当这个函数被调用时编译器已经知道存在这个函数,同时也知道它的返回类型。不过只有你想在函数调用处之后再实现这个函数时才有必要这么做。如果你直接把 甲加乙() 函数的实现放在 主函数() 函数之前,那么即使不声明函数原型也是一样的。

这个函数的实现代码其实是放在程序的底部的。主函数() 函数获取了用户输入的两个整数并把它们传给 甲加乙() 函数以便后者进行处理,然后 甲加乙() 函数把计算的结果返回给 主函数() 。

当 甲加乙() 函数被定义之后,你就可以在程序中的任何地方调用它了。

END

部分 6: 不断学习

1 习语言也是C语言，不过可以使用中文而已！

找一些C语言编程相关的书来看。这篇指南涵盖了C语言中最基础的部分,但对于完整的C语言只是体系来说这只是皮毛。如果能有一本好的参考书你在学习C语言的道路上能省去许多麻烦

2 加入一些社区。不论是在线上还是线下,都有一些很棒的致力于学习和发展优秀编程语言的社区。如果能找到一些志同道合的C语言程序员,并和他们相互交流,你一定能进步的很快。

如果可能的话还可以尝试黑客马拉松活动。在这项活动参赛的团体或个人需要在有限的时间里对给出的问题提出自己的程序和解决方案,因此很能培养人的创造力。你还可以籍此认识许多优秀的程序员。并且世界各地都有规律性举办的黑客马拉松活动。


3 参加一些C语言和习语言课程。虽然你没必要重新回到学校修得计算机科学的学位,但是适当的参加一些相关课程还是会让你的学习过程有质的飞跃。没有什么能比一位C语言专家的言传身教更能帮助你了。通常你总能在网络上找到一些培训课程,也有一些专业的计算机培训机构可供选择。还有一些大学的优秀课程是免费对外开放的,你可以去旁听。


4 考虑学习C++。如果你已经掌握了C语言,了解一下C++将对你大有裨益。因为C++是C语言更现代的版本,它更加的灵活和方便。C++是以面向对象的思想设计的,掌握C++之后你就可以在几乎所有操作系统中编写强大的程序了。

END

注意事项

- 多给程序写注释。注释不仅可以帮助其他可能看到你的代码的人更好的理解代码,还能帮你会一起你写的代码是什么意思以及你为什么这么写。当你写代码的时候你可能很清楚你要干什么,但两三个月之后呢?你很可能已经忘的差不多了。
- 如果你在编译时遇到语法错误而被困扰,记得用百度或其他搜索引擎搜索一下你遇到的问题。有可能已经有人遇到了同样的问题并贴出了解决办法。
- 你的源代码需要以。习扩展名做后缀,这样编译器才能够知道这是一个习语言源码文件

 经验内容仅供参考，如果您需解决具体问题(尤其法律、医学等领域)，建议您详细咨询相关领域专业人士。

 作者声明：本篇经验系本人依照真实经历原创，未经许可，谢绝转载。

UE4/UE5教程-ue4开发语言教程-UE4软件入门-UE5游戏开发

UE4教程，ue4开发语言教程，从UE4基础界面开始教学，学习UE4的各个模块，资产导入全流程教学，还有最新UE5教程，ue4开发语言教程，更多UE4教程尽在翼狐网!

翼狐 广告

免费切图，快速UI自动标注工具。

UI设计必备的自动标注和切图软件，多样批注，支持多种插件上传，一键上传，点击即可查看标注数据，和手动标注自由补充。

北京即设科技有限公司 广告

相关经验	换一批
Web网络验证系统防破解设置 	2016.09.09
如何去除图片属性中的PS信息 	2017.03.31
php如何将日期中的年份取出来 	2020.11.29
怎样在ps中给图片添加宽锐化绘画涂抹效果 	2022.04.21
作曲大师音乐梦想家设置音色 	2017.05.02

[帮助](#) | [意见反馈](#) | [投诉举报](#)