第 3 章 第一个仓颉程序


6min

3.1 运行 Hello World 程序

做好仓颉的开发准备后，就可以编写第一个仓颉程序了，这里以最简单的 Hello World 程序为例，演示编写、编译及运行的整个过程。

步骤 1：创建 hello_world.cj 的源程序文件，然后使用合适的编辑器进行编辑，代码如下：

```
//Chapter3/hello_world.cj

/*
程序主入口
*/
main() {
    //打印 hello world
    print("Hello, World!\n")
}
```

步骤 2：编译 hello_world.cj 源程序。在 Linux 平台上的编译命令如下：

```
cjc -o hello_world hello_world.cj
```

编译成功后，将会生成 hello_world 可执行文件。

步骤 3：执行 hello_world。命令及回显如下：

```
./hello_world
Hello, World!
```

这样就完成了第 1 个仓颉程序的运行。

如果是 Windows 平台，编译、运行命令及回显如下：

```
cjc -o hello_world.exe hello_world.cj
.\hello_world.exe
Hello, World!
```

3.2　仓颉程序基本规则

根据 3.1 节的 Hello World 程序，总结仓颉程序编写的基本规则如下。

1. 仓颉源程序文件的扩展名是.cj

这是仓颉源程序文件的专用扩展名，就像 C 语言的.c 扩展名，Java 语言的.java 扩展名一样，是仓颉源程序文件的特有标志，如果把.cj 改成别的扩展名，则编译时会报错。

2. 仓颉程序需要有一个程序执行的入口函数 main

这一点和大部分编程语言类似，程序执行时首先找到 main 函数，从 main 函数开始执行，一个可执行的仓颉程序需要有且只能有一个 main 函数。main 函数的返回值可以是整数类型或者 Unit 类型，main 函数可以没有参数，如果有参数，参数必须是 Array < String > 类型。如果仓颉程序不是可执行的，则可以没有 main 函数。

3. 仓颉程序对大小写敏感

不管是仓颉本身的关键字，还是定义的函数、变量名称都对大小写敏感，不能混用。

4. 仓颉程序源码可以添加注释，注释有规定的语法

单行注释使用//，在//后面并在换行以前的内容会被编译器忽略；多行注释使用/ * 开始 * /结束，这两者中间的一行或者多行内容都会被编译器忽略。良好的注释可以增加源程序的可读性，让开发者可以更好地理解程序的逻辑。

5. 仓颉程序拥有丰富的内置函数库，可以直接使用

print 函数是仓颉 core 库中的全局函数，除此之外，还有 io 库、net 库、time 库、os 库等几十个内置库，这些库里包含的函数可以在导入所在的包后直接调用，详细的包管理方法会在后续章节介绍。

3.3　仓颉程序的编译

仓颉源文件可以使用 cjc 编译命令进行编译，cjc 编译命令功能强大，参数众多，在本节只介绍最简单的几种，更多的参数将在后续章节介绍。

编译命令的格式如下：

```
cjc [options] file(s)
```

其中 options 为可选的参数，file 为一个或者多个待编译的文件，以 3.1 节 hello_world.cj 为例，最简单的编译方式就是直接编译，输入的命令如下：

```
cjc hello_world.cj
```

这样，在 Linux 环境下默认生成的输出文件的名称是 main，在 Windows 环境下是 main.exe。

典型的可选参数如下所示。

1. --output <value>,-o <arg>

设置输出文件的路径,其中-o 为简写形式,示例如下:

```
cjc hello_world.cj -- output /data/hello
```

该示例将编译成 hello 文件并输出到/data 目录下。

2. --version,-v

打印编译器版本信息,其中-v 为简写形式。

3. --help,-h

显示帮助信息,其中-h 为简写形式。

如果一个仓颉程序包含多个源程序文件,例如 a.cj、b.cj、c.cj,则可以在编译时列出这些文件,示例如下:

```
cjc a.cj b.cj c.cj -o d
```

该命令将编译这些源程序文件,并生成 d 文件。