**软件工程中级实训——Ant学习笔记**

1. **Ant的作用：**

Ant用来构件项目，只需要执行简单的执行命令就可以。Ant是比较好的构件工具，但脚本目录要自己写，命令中存在着依赖关系，编译，打包等，而且编译目录要自己指定。

1. **Ant的特点：**

（1）Ant不像Maven一样，没有正式的约定，项目的目录结构可随意定义。但是你却要明确的告诉Ant，源码在哪里，编译后的文件放在何处。

（2）Ant是一个程序化的工具，你需要明确告诉ant去做什么，什么时候做。如：什么时候去编译，什么时候去复制，压缩，打包。

（3）Ant没有生命周期，所以你需要自己去定义目标之间的关系和依赖，必须手动去为每个项目附上一个任务序列。

1. **Ant的XML元素**

**1.<project>元素**

是Ant生成文件的根元素；一般形式如：

<project default=" "[ basedir="." ] [name="projectname"]>

（1）default的值是默认执行的target名；

（2）basedir是指定基准目录，一般都是basedir="."

（3）name是指工程名字；

**2.<target>元素**

是project元素的子元素，在project元素中能够有多个target，一个target标签可以依赖其他的target；一般形式如下：

<target name=" " [ depends=" A "] [ if =" prop1 "] [ unless=" prop2"]>

</target>

（1）name表示target的名称；

（2）depends中填写其他target名称(可以有多个名称，用逗号分割)，若依赖A，如果执行此target，则会先执行A；

（3）表示仅当属性设置时才执行此target；

（4）unless表示只有没有设置该属性名，才能执行此target；

**3.<property>元素**

是project的子元素；property用于定义属性，一般形如：

(1)<property name=" pname " value=" pvalue "/>

如果要使用此property，则需要${pname}；类似于表达式语言；

(2)<property file="a.properties"/> 通过此属性文件导入属性；

如果单纯想使用$，则通过$$表示；

1. **Task介绍**

在Ant中Task是target的子元素，即一个target中可以有多个task；而Task分为：

**(1)核心Task；**

**(2)可选Task；**

**(3)自定义Task；**

下面介绍一些常用的核心Task；

**1.<echo>**

用于单纯输出，一般形如：

<echo> </echo>

举例：<echo>hello</echo>

**2.<javac>**

用于编译java文件，一般形式如下：

<javac srcdir=" src " destdir=" class " [classpath=""]/>；

(1).srcdir表示源程序的目录。

(2).destdir表示class文件的输出目录。

(3).include表示被编译的文件的模式。

(4).excludes表示被排除的文件的模式。

(5).classpath表示所使用的类路径。

(6).debug表示包含的调试信息。

(7).optimize表示是否使用优化。

(8).verbose 表示提供详细的输出信息。

(9).fileonerror表示当碰到错误就自动停止。

**3.<java>**

运行java类，一般形式如下：

<java classname=" " fork="yes">

【<arg line="param1 param2 param3"/>】

</java>

(1).classname 表示将执行的类名。

(2).jar表示包含该类的JAR文件名。

(3).classpath所表示用到的类路径。

(4).fork表示在一个新的虚拟机中运行该类。

(5).failonerror表示当出现错误时自动停止。

(6).output 表示输出文件。

(7).append表示追加或者覆盖默认文件。

**4.<jar>**

打包成jar包；一般形式如下：

<jar destfile="main.jar" basedir=" ">

<manifest>

<attribute name="Main-Class" value="classname"/>

<!--指定主类-->

</manifest>

</jar>

(1) destfile表示JAR文件名。

(2) basedir表示被归档的文件名。

(3) includes表示别归档的文件模式。

(4) exchudes表示被排除的文件模式。

**5.<mkdir>**

创建目录，可以多层创建，比如a\b\c，则可以连续创建，一般形式如下：

<mkdir dir="a\b"/>

**6.<delete>**

删除目录，一般形式如下：

<delete dir="a\b"/> 可以删除a目录下的b目录；

<delete file="1.txt"/>可以删除文件；

**7.<tstamp>**

时间戳，一般形式如下：

<tstamp />

接下来可以使用${DSTAMP}进行调用当前时间；

**8.<copy>**

复制文件，一般形式如下：

<copy file="file1" tofile="file2"/>

file是源文件；

tofile是目标文件；

**9.<move>**

移动文件，一般形式如下：

<move file="file1" tofile="file2"/>

file是源文件；

tofile是目标文件；

**10.<replace>**

用于替换字符串，类似于String的replace操作，一般形式如下：

<replace file="filename" token="old" value="new"/>

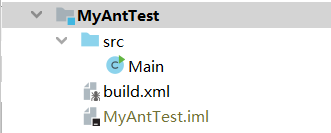
file表示要执行替换的文件；

token表示被替换的字符串；

value表示替换的字符串；

1. **Ant的使用Case**

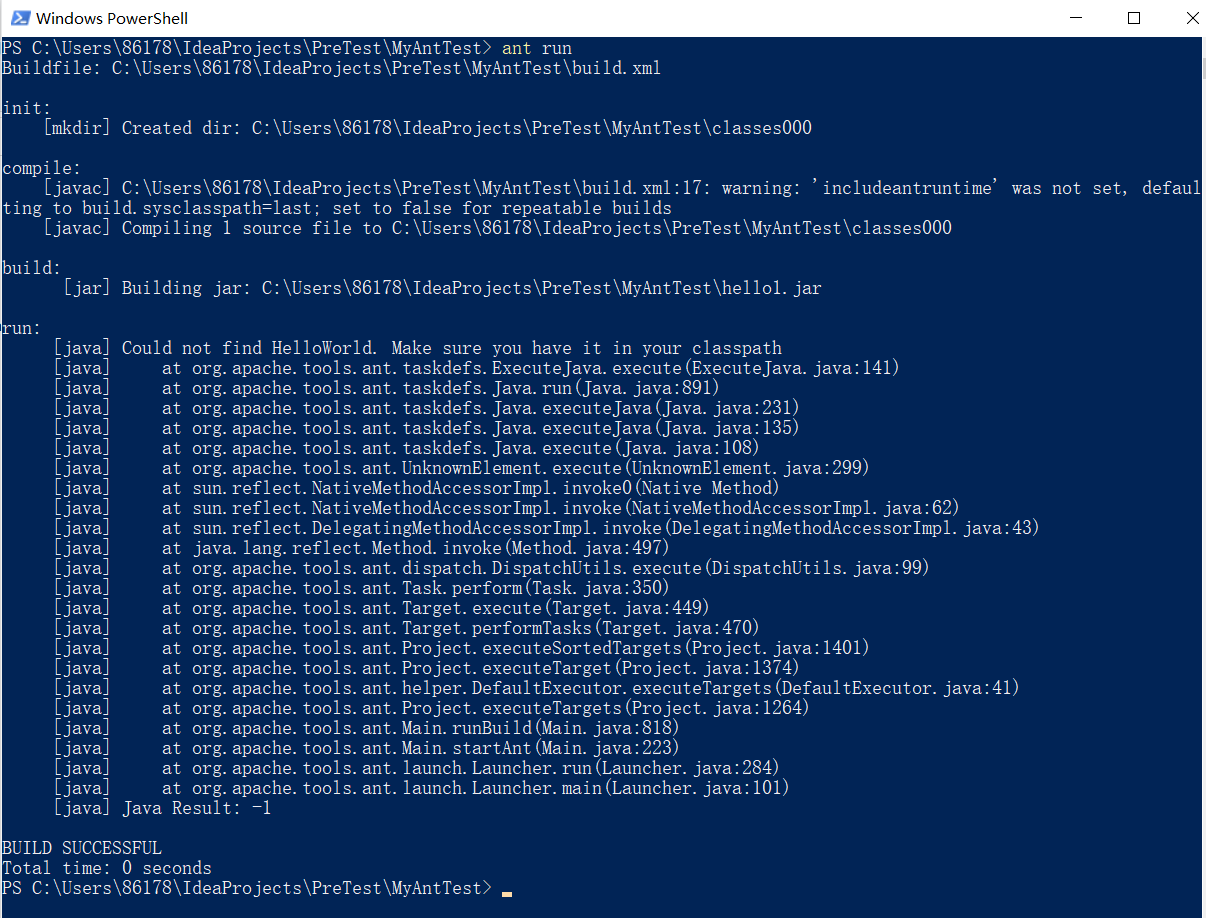
**原来的文件结构：**

****

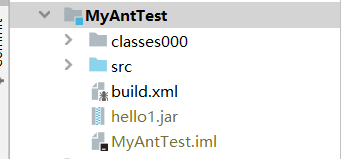
**Build.xml的部分配置：**

****

**命令行运行：**

****

**后来的文件结构：**

****

**Ant运行成功，完成对部分的Ant功能的测试。**