Day 02

2018年4月28日

8:54

课前回顾:

介绍window基本常用命令

cd

dir dir/p

md

rd

del

copy

上面这些命令属于window系统中的内部命令,内置好的命令

notepad

calc

mspaint

mstsc

上面4个命令不属于window的内置命令

jdk中安装目录中bin目录中的java命令和javac命令javadoc命令

不属于window的内置命令,需要设置环境变量,就是为了能够在

任何目录中都可以敲入java命令,首先在当前目录中寻找,如果寻找不到

去定义的环境变量指定的路径中去寻找%JAVA\_HOME%\bin

或C:\Program Files\Java\jdk1.8.0\_65\bin

配置环境变量:

系统环境变量:

JAVA\_HOME C:\Program Files\Java\jdk1.8.0\_65

path ;%JAVA\_HOME%\bin

在doc.tedu.cn

系统环境变量:

JAVA\_HOME C:\Program Files\Java\jdk1.8.0\_65

path ;%JAVA\_HOME%\bin

classpath .;%JAVA\_HOME%\lib

检验环境变量是否能用:

win+r 启动cmd命令行模式

java -version 查看当前系统下jdk的版本

javac 回车

环境变量配置成功,命令可用

开始编写java代码:

public class Hello{

public static void main(String[] args){

System.out.println("Hello World!!");

}

}

在cmd中,用javac编译用java命令执行

javac Hello.java javac e:\test\Hello.java

java Hello 此时是执行的class字节码文件

计算机生成了可选文字:
E:XtestXHello.• va 一 Note d++ Administrato 
# 旧 考 旧 搜 视 (V) 格 忒 〔 M ） 言 旧 设 (T) 宏 孓 刂 运 行 〔 R ） # 凹 窗 匚 〔 W ， 2 
目 g 
Hell•xja 
Col ： 2 
:0 阳 
Ln ： 7 
可 以 任 意 命 名 
a r S 
1 
2 
3 
4 
5 
6 
Java source file 
工 C C 1 a S S 
He 110 
公 有 类 的 名 字 跟 文 件 名 相 同 
主 函 数 ， 是 程 序 入 所 有 软 件 必 须 至 少 有 一 个 main 入 规 定 
u 
写 n 多 行 代 码 
工 C S a 工 C VOI ma1n 
r 工 n 
ystem.out.println （ "Hello Wo 1 d ！ ！ " 
说 明 ： 
java 中 的 关 键 字 
public 公 有 
class 代 表 类 定 义 
static 静 ； 
void 无 返 回 类 型 
java 中 系 统 类 
String 代 表 字 符 串 
String0 代 表 字 符 串 数 组 
System 系 统 类 
println 是 方 法 ， 打 印 输 出 内 容 
DosXWind 
继 续 添 加 方 法 （ 函 数 ） ， 属 性 
可 以 继 续 添 加 类 的 定 义 ， 就 是 不 能 为 public 
一 个 java 文 件 中 可 以 放 置 多 个 类 定 义 ， 但 公 有 只 能 
有 一 个 
Class 
class bb( 
length 
： 11 
lines 
： 7 

计算机生成了可选文字:
通 过 某 种 编 辑 " 工 具 软 件 ： 
记 事 本 notepad++,editplus,ultraedit,eventeidt 等 
eclipse,myeclipse,netbeans,idea 等 
用 工 具 软 件 编 写 / 查 看 代 码 ， 编 写 完 java 代 码 ． java 文 件 
jdk 
编 译 
通 过 jdk 把 ． java 的 源 代 码 文 件 编 译 成 ． class （ 字 节 码 文 件 
加 载 到 jvm 
jvm,java 虚 拟 机 ， 把 .class 字 节 码 文 件 放 在 jvm* 执 行 
通 过 os 操 作 系 统 用 的 还 是 本 地 计 算 机 的 硬 件 
操 作 系 统 ， 负 责 管 理 计 算 机 的 硬 件 
硬 件 

Java中定义包(package)

java中叫做包package,包最终对应的是文件夹

package cn.tedu;//package是关键字,打出一个cn.tedu包

public class Hello{

public static void main(String[] args){

System.out.println("Hello World!!");

}

}

带包编译和带包执行

javac -d 编译后的存放的目录 java文件名

计算机生成了可选文字:
javac -d E:XXex Demo.java--- 要 繼译 的 是 Demo-java ， 编 译 之 后 
放 到 E ： \ \ ex 目 录 下 

java -cp 编译后的存放的目录 包名.Class文件名

计算机生成了可选文字:
java -cp C:xexxtest cn 、 tedu.Demo - class 文 件 是 Demo.class 
在 cn.tedu 包 下 ， cn 包 再 C:xexxtest 目 录 下 

总结:

0.前提必须有jdk+环境变量

1.编写java代码

2.用javac编译java代码为class字节码文件

3.用java命令执行字节码文件

4.带包 javac -d

java -cp

java中标识符:就是给某个东西起个名

java中的起名规则:

名称中(标识符中),可以有,字母(大小写),数字,\_,$

就是以数字开头,不能是关键字,区分大小写

错的:

1ab ,public,a b,a%b

对的:

张三(不建议), Ab ,ab

命名方法:

驼峰命名法(骆驼命名法) 定义方法名,属性名

从第二单词开始所有的单词第一个字母大写,其余的小写

userName ,userPassWord

帕斯卡命名法: 定义类名字

所有单词的第一个字母都要大写

StudentClass , TestClass

用上面的命名规则

可以定义类名 帕斯卡

可以定义方法名 驼峰

可以定义变量名 int age; 驼峰

可以定义常量名 final double MATH\_PI=3.14; 都要大写单词间用\_间隔

可以定义报名 都是小写中间用.间隔 cn.tedu

注释:注释的东西,不是有效的java代码

//单行注释

/\*

多行注释

\*/

/\*\*

文档注释

\*/

数据在计算机(java)中的表示:

计算机/java最终需要底层硬件,

用二进制只有两个数 0 1 逢二进一

用八进制 有8个数 0-7 逢八进一

用十进制 有10个数 0-9 逢十进一

用十六进制 有16个数 0-9 a-f 逢16进一

做进制的目的,就是为了能快速,简短,准确的表达数据(二进制数据)

二进制 八进制 十六进制 十进制

0000 0 0 0

0001 1 1 1

0010 2 2 2

0011 3 3 3

0100 4 4 4

0101 5 5 5

0110 6 6 6

0111 7 7 7

1000 10 8 8

1001 11 9 9

1010 12 a 10

1011 13 b 11

1100 14 c 12

1101 15 d 13

1110 16 e 14

1111 17 f 15

10000 20 10 16

Java程序中表达二进制数据 数据的前面加上前缀0b/0B(jdk1.7及以上)

比如:

0b1010 0B1010

java程序中表达八进制数据 数据的前面加上前缀0

比如:

017+01=20

java程序中表达十六进制数据 数据的前面加上前缀0x/0X

比如:

0x1ab

进制转换:

十进制跟非十进制转换

除权(二,八,十六)取余数,倒着取

十 --> 二

十 --> 八

十 --> 十六

比如: 13十进制转二进制

计算机生成了可选文字:
工 3 一 工 工 O 工 

计算机生成了可选文字:
26 

本位乘以权位(二,八,十六)的次方序号

二-->十

八-->十

十六-->十

比如:

10011010110

计算机生成了可选文字:
1 * 2 10 次 方 + 1 * 2 7 次 方 + 1 * 2 6 次 方 + 1 * 2 的 4 次 方 + 1 * 2 的 2 次 方 + 1 * 2 的 1 次 方 

非十进制(二,八,十六)之间的转换

二 <-----> 八 三个二进制位对应一个八进制位

比如:

100 101 101 111 二进制

4 5 5 7 八进制

二 <---> 十六

1001 0110 1111 二进制

9 6 f 十六进制

八 <---> 十六 借助二进制中转

计算机中的数据码制表示 原码 反码 补码

对于正数，原反补三码一致。

int i = 5;

原码：00000000 00000000 00000000 00000101

反码：00000000 00000000 00000000 00000101

补码：00000000 00000000 00000000 00000101

对于负数，反码是在原码的基础上最高位不变，其余位1变0，0变1;

补码在反码的基础上+1

注意：最高位是符号位，0表示正数，1表示负数

int i = -3;

原码：10000000 00000000 00000000 00000011

反码：11111111 11111111 11111111 11111100

补码：11111111 11111111 11111111 11111101

在java编程中如何表达数据:

在所有的程序中就做了三件事:

1.获取要运算的数据

2.运算获取的数据

3.把运算的结果显示或存储(持久化硬盘或数据库)

常量:量值是固定的

整型常量: 123 567

小数常量:1.23 56.7

字符常量:用单引号括起来的单个字符 'a' '1'

字符串常量:用双引号括起来的0个或多个字符

""

"abc\_123"

"123"

布尔:只有两个值true 和false

true

false

空常量: null,只要看到是null,就是一个空值/空引用

带有final的量也是常量

final double MATH\_PI=3.14D

变量:量值是变化的,就是存储数据的空间的内容是可以根据需求变化的

变量的定义

语法:声明一个变量

类型 变量名;

在java中一定要先定义变量,然后才能给变量赋值

比如:

先定义后赋值

int age; //声明一个变量

age=20;//给变量赋值

age=21;

在定义的同时就赋值

int weight=200;

分析变量的语法定义

类型 变量名;//为什么要这样定义一个变量

就是以变量名为理由,跟jvm管理的内存申请类型这么大空间

int age=20;

int类型在内存中占用4个字节

以age为理由跟内存申请4个字节空间,空间里存储的值是20这个常量值

以后可以通过变量名获取变量中的值,也可以把其他常量值赋值给

这个变量名指定的空间,把原来的值覆盖掉

结论:

用变量和常量存储数据,可以跟根据这些数据,结合业务要求,做运算

内存中如何表达常量和变量

是通过变量名和常量名来表达的,

通过变量名和常量名在内存中寻址(内存地址,二进制)

计算机生成了可选文字:
变 量 名 
数 据 类 型 变 量 名 = 數 据 ； 
intj: 7 ； 
i = 6; 
数 据 类 型 变 量 名 ； 
变 量 名 = 數 据 ． 

java中的八个基本数据类型:

byte:

short:

int

long

float

double

char

boolean

所有语言都有基本数据类型,java也有八中基本数据类型

java规定了跟内存申请空间多大,空间中的数据就会有范围了

byte 1 个字节 范围 -27 ----- 27-1

byte b=129;

Type mismatch: cannot convert from int to byte

类型 不匹配 不能 转换 从 int 到 byte

short 短整形 2个字节 表示范围是-215~215-1 --- -32768~32767

int 整型 4个字节 表示范围是-231~231-1 --- -2.1\*1010~2.1\*1010

long 长整型 8个字节 表示范围 -263~263-1 --- -9\*1018~9\*1018

后边添加L/l标志是一个长整型：long l = 31864853188L;

float 单精度浮点 占4个字节 表示范围 -1038~1038

要求必须以F/f标志float类型的数字，

float f = 3.23f;

double 双精度浮点 占8个字节 表示范围-10308~10308

小数默认为double

System.out.println(3.4);

double d = 4.2;

double d2 = 5.6D;

double d3 = 6.01d;

char---字符型---占2个字节--- 0~216\_1 --- 0~65535

char ch='a'; char ch='中';

转义字符：‘\t’ ---table 制表符 ‘\r’---return 回车 ‘\n’---next/newLine换行

‘\\’ ---\ ‘\’’ --- 单引号 ‘\”’ --- 双引号

boolean 布尔型 只能表示true/false

内存大小根据jdk版本和操作系统来确定

boolean b = true;

总结:用八种数据类型可以存储指定类型的数据

补充内容:

用eclipse做开发工具开发java代码:

首先去www.eclipse.org下载eclipse的的安装包(选java EE版本)

下载后解压eclipse的压缩包,路径中不要带中文和空格

解压后的目录中有eclipse.exe的启动文件,启动eclipse

启动eclipse的时候,必须先安装好jdk,在启动eclipse的时候,

自动寻找本机的jdk,如果有引用,然后正常启动eclipse,

如果没有寻找到,会eclipse不能启动

如果正常启动eclipse,会提用户指定工作空间(workspace),这个workspace

存储的当前eclipse中所有的项目源代码

计算机生成了可选文字:
： ， Workspace Launcher 
a workspace 
Eclipse stores you r projects in a folder called a workspace 
C 卜 00 ' e a workspace folder to use for this session. 
、 Work ' p30 e:XworkspaceXmars2 
工 作 空 间 的 目 录 么 有 
自 动 创 建 
Browse 
不 打 对 勾 ， 每 次 启 动 都 提 示 
， 囗 use this a ' the defaultand do notask again 
这 个 窗 口 不 建 议 打 对 勾 

点击"ok" 正常启动eclipse

在Project explorer 中右键单击空白处---new--project(工程或项目)

计算机生成了可选文字:
： ， ： New Project 
a wizard 
Wizards: 
L:' 」 3v3 ， ro 」 ect 
isting Ant Buildfile 
Plug-in Project 
General 
Eclipse Modeling Framework 
Finish 
Cancel 
Java 
Java E E 
JavaScript 
JAXB 
0 
， Back 

计算机生成了可选文字:
New Java ProJe 
提 示 创 建 一 个 java project 
Create a Java Project 
Create a Java project in the workspace or I n a n external | 003ti0m 
创 建 一 个 java 项 目 在 工 作 空 间 或 在 一 个 额 外 位 置 
Project name• 」 ava ， rp 」 e 司 
工 程 项 目 的 名 称 
囝 use default location 
Location: E:XworkspaceXmars2XJavaPrqect 
JRE 
Use a n executlOn JRE: 
0 
use a project ' pec 罱 JRE: 
se default JRE (currently 3re1.8.0 651 〕 
存 储 位 置 
JavaSE-I.8 
Jre1.8.0 65 
Configure 」 RE “ 
使 用 默 认 jre 版 本 ， 根 据 工 作 空 间 当 
Project layout 
前 jre 版 本 匹 配 
0 
use proj 
回 Create separate folders for sources and class files 
添 加 src 的 文 件 夹 
Worki 
囗 
Add project to working sets 
Configure default.. 
Select. 
Working sets: 

计算机生成了可选文字:
： ， ： Open Associated Perspective? 
打 开 指 定 视 图 
This kind Of project associa 
This perspective 症 designed to support Java development. It 0 “ a 
Package Explorer, a Type Hierarchy, and Java-specific navigation actions. 
DO you wa nt to open this perspective now 
囗 
Remember my decision 

项目创建完后

在src创建包package,

在包中可以创建若干java类

给eclipse设置全部项目为utf-8

window --- preferences(首先项)---general---

workspace---test file encoding--- utf-8