Java基础-Java语言基础（二）最终版

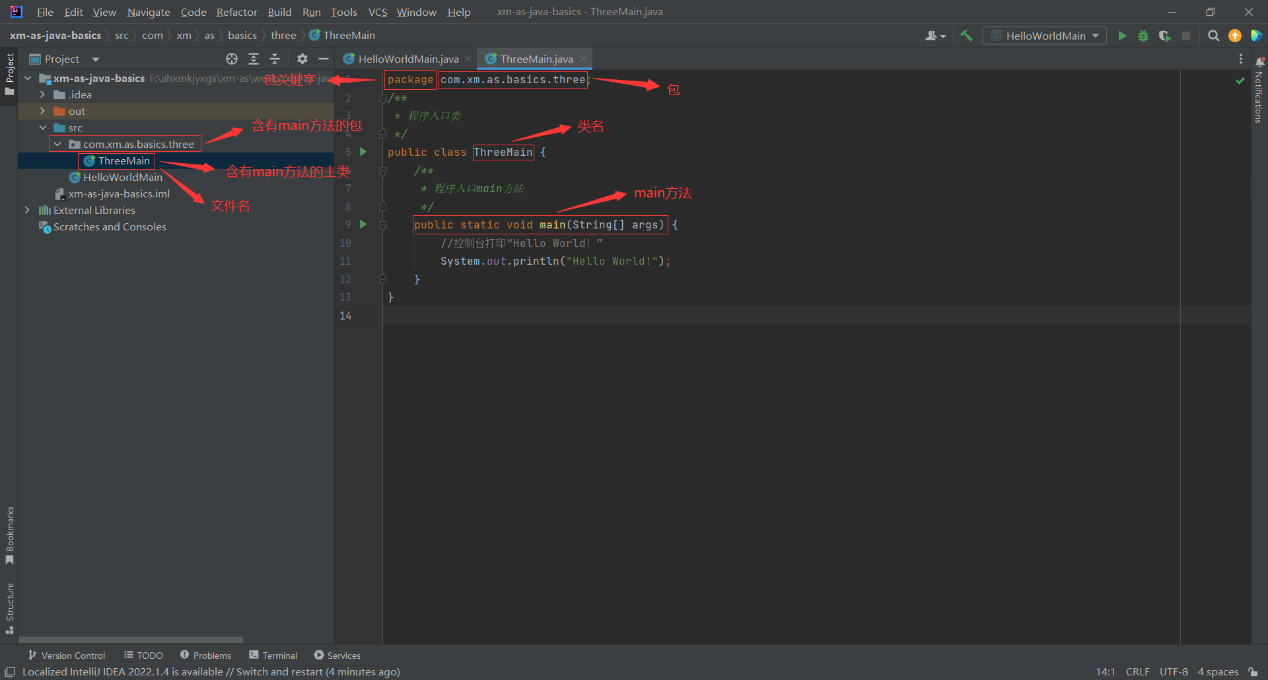
1. Java包、类、main方法

1、Java包分为缺省包与自定义包

2、包的关键字是：package

3、Java包会根据业务模块进行命名和拆分，每个包里都包含若干类

4、包含main方法的Java包，如下图所示：



main方法的代码分析：

①public为访问修饰符

②class为类关键字

③ThreeMain为类名

④static为关键字

⑤void为返回类型

⑥main为程序入口方法名

⑦String为String类

⑧args为字符串数组

⑨System.out.println()为在控制台输出打印语句

⑩/\*\*中间内容省略\*/或//为程序注释，编译运行时省略

5、包含main方法的类被称为Java主类

6、Java带有main方法的主类是Java程序的基本组成部分

7、Java除了主类之外，还有很多类，如：实体类、业务类、工具类、算法类等等，会根据具体的需求描述、写法描述等而叫法不同

8、每一个Java程序都必须包含一个main方法

9、main方法从“{”开始，到“}”结束

10、main方法在主类中也被称为主方法

11、main方法是Java程序开始执行的入口

12、注1：Java代码中所有的标点符号都是英文字符

13、注2：Java代码中类名和文件名必须相同

14、注3：Java代码是区分大小写的，类名首字母按规范需要大写，满足驼峰命名规则

1. Java关键字

1、Java关键字是Java程序语言中被赋予特定含义的单词，不可以把这些单词作为其它代码的命名

2、Java关键字，如下表所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| package | public | protected | default | private |
| class | static | void | byte | short |
| int | long | float | double | char |
| boolean | final | if | else | switch |
| case | while | do | for | break |
| continue | return | extends | abstract | interface |
| implements | import | new | this | super |
| throw | throws | try | catch | finally |
| synchronized | instanceof | goto | volatile | strictfp |
| transient | dafault |  |  |  |

1. 标识符

1、标识符用来标识类名、变量名、方法名、文件名等等

2、Java语言规定，标识符由任意顺序的字母、下划线“\_”、美元符号“$”、数字组成，第一个字符不能是数字，标识符不能是Java关键字

3、合法的标识符如：code、code\_s、code$、code1、code\_$1

1. 基本数据类型

1、整数类型

①整数类型用来存储整数数值，可以是正数或负数，不可以有小数部分

②整数类型包括：byte、short、int、long

③整数类型的表示形式包含：十进制、八进制、十六进制

④十进制如：0、100、-100

⑤八进制必须以0开头

⑥十六进制必须以0X或0x开头

⑦byte的内存空间为8位，short的内存空间为16位，int的内存空间为32位，long的内存空间为64位，8位等于一个字节

⑧示例代码如下：

|  |
| --- |
| public static void main(String[] args) {  byte byte1 = 100; //声明byte类型变量byte1并赋值  short short1 = 1000; //声明short类型变量short1并赋值  int int1 = 2000; //声明int类型变量int1并赋值  long long1 = 3000; //声明long类型变量long1并赋值  long result = byte1 + short1 + int1 + long1; //求上述整数类型变量值之和  System.out.println("上述整数类型变量值之和：" + result);  } |

2、浮点类型

①浮点类型用来存储含小数数值

②浮点类型包括：单精度浮点类型“float”、双精度浮点类型“double”

③浮点类型默认为double类型，如果使用float类型，需要在小数后面添加“F”或“f”，不可以省略，double类型也可以在后面添加“D”或“d”，省略也可以

④float的内存空间为32位，double的内存空间为64位，8位等于一个字节

⑤示例代码如下：

|  |
| --- |
| public static void main(String[] args) {  float f1 = 100.01f; //声明float类型变量f1并赋值  double d1 = 200.02; //声明double类型变量d1并赋值  double d2 = 300.03d; //声明double类型变量d2并赋值  double rfd = f1 + d1 + d2; //求上述浮点类型变量值之和  System.out.println("上述浮点类型变量之和：" + rfd);  } |

3、字符类型

①字符类型用于存储单个字符

②字符类型包含：char

③字符类型变量要以单引号括起来，如：’a’

④char的内存空间16位，8位等于一个字节

⑤示例代码如下：

|  |
| --- |
| public static void main(String[] args) {  char c = 'a'; //声明char类型变量c并赋值  char cs = 97; //声明char类型变量cs并赋值，因为字符'a'在unicode表中的排序位置为97，所以'a'可以换成97  System.out.println("char类型c：" + c);  System.out.println("char类型cs：" + cs);  } |

⑥转义字符是一种特殊的字符变量，以“\”开头，后面加上一个或多个字符

⑦Java转义字符，如下表所示：

|  |  |
| --- | --- |
| 转义字符 | 所表示含义 |
| \’ | 单引号字符 |
| \” | 双引号字符 |
| \\ | 反斜杠字符 |
| \t | 制表符 |
| \r | 回车 |
| \n | 换行 |
| \b | 退格 |
| \s | 空格 |
| \0 | 空字符 |
| \f | 换页 |
| \ddd | 八进制字符 |
| \uxxxx | 十六进制字符 |

⑧示例代码如下：

|  |
| --- |
| public static void main(String[] args) {  char c1 = '\''; //单引号字符  char c2 = '\"'; //双引号字符  System.out.println("单引号字符：" + c1);  System.out.println("双引号字符：" + c2);  } |

4、布尔类型

①布尔类型用于逻辑判断，基本体现在流程控制中

②布尔类型包括：true、false

③true代表“真”，false代表“假”

④布尔类型值不能与整数类型值进行转化

⑤示例代码如下：

|  |
| --- |
| public static void main(String[] args) {  boolean b1 = true; //声明boolean类型变量b1并赋值  boolean b2 = false; //声明boolean类型变量b2并赋值  System.out.println("布尔类型b1：" + b1);  System.out.println("布尔类型b2：" + b2);  } |

1. 运算符

1、赋值运算符

①赋值运算符用“=”表示，是一个二元运算符

②是将右边的值赋给左边的类型变量

③赋值运算符先得到右边的结果，再赋值给左边的类型变量

④示例代码如下：

|  |
| --- |
| //赋值运算符  int i1 = 100; //声明int类型变量i1并赋值  int i2 = 200; //声明int类型变量i2并赋值  int i = i1 + i2; //变量i1和变量i2相加之后的结果赋值给变量i  System.out.println("int类型变量i：" + i);  int i3,i4,i5; //声明int类型变量i3、i4、i5  i3 = i4 = i5 = 500; //i3、i4、i5同时赋值  System.out.println("int类型变量i3：" + i3);  System.out.println("int类型变量i3：" + i4);  System.out.println("int类型变量i3：" + i5); |

2、算术运算符

①算术运算符包含：加“+”、减“-”、乘“\*”、除“/”、取余“%”

②“+”可以作为数值的正号，如：+100

③“-”可以作为数值的负号，如：-100

④示例代码如下：

|  |
| --- |
| //算术运算符  double db1 = 200.20; //声明double类型变量db1并赋值  double db2 = 100.10; //声明double类型变量db2并赋值  System.out.println("相加和：" + (db1 + db2));  System.out.println("相加差：" + (db1 - db2));  System.out.println("相乘积：" + (db1 \* db2));  System.out.println("相除商：" + (db1 / db2));  System.out.println("取余：" + (db1 % 3)); |

3、自增、自减

①自增自减运算符是单目运算符，可在变量之前，也可在变量之后

②自增自减运算符针对的变量必须是整型或浮点型

③自增，如：i++（变量i之后，i的值加“1”），++i（变量i之前，i的值加“1”）

④自减，如：i--（变量i之后，i的值减“1”），--i（变量i之前，i的值减“1”）

⑤实例代码如下：

|  |
| --- |
| //自增、自减  int i10 = 10; //声明int类型变量i10并赋值  int i20 = 20; //声明int类型变量i20并赋值  int i30 = 30; //声明int类型变量i30并赋值  int i40 = 40; //声明int类型变量i40并赋值  System.out.println("自增i10++：" + (i10++));  System.out.println("自增++i20：" + (++i20));  System.out.println("自减i30--：" + (i30--));  System.out.println("自减--i40：" + (--i40)); |

4、比较运算符

①比较运算符用于变量之间、其它类型信息之间的比较

②比较运算符的运算结果为布尔类型

③比较运算符包括：“>”、“<”、“==”、“>=”、“<=”、“!=”

④实例代码如下：

|  |
| --- |
| //比较运算符  int i51 = 51; //声明int类型变量i51并赋值  int i52 = 52; //声明int类型变量i52并赋值  System.out.println("i52 > i51的值：" + (i52>i51));  System.out.println("i52 < i51的值：" + (i52<i51));  System.out.println("i52 == i51的值：" + (i52==i51));  System.out.println("i52 >= i51的值：" + (i52>=i51));  System.out.println("i52 <= i51的值：" + (i52<=i51));  System.out.println("i52 != i51的值：" + (i52!=i51)); |

5、逻辑运算符

①逻辑运算符可以组合其它运算符共同组成一个新的表达式，返回值为布尔类型

②逻辑运算符包括：逻辑与“&”或“&&”、逻辑或“||”、逻辑非“!”

③示例代码如下：

|  |
| --- |
| //逻辑运算符  int i61 = 61; //声明int类型变量i61并赋值  int i62 = 62; //声明int类型变量i62并赋值  System.out.println("逻辑值1：" + ((i62>i61)&&(i62>=i61)));  System.out.println("逻辑值2：" + ((i62<i61)||(i62<=i61)));  System.out.println("逻辑值3：" + (!(i62<i61))); |

6、三目运算符

①三目运算的格式：条件表达式 ? 值x : 值y

②若条件表达式的值为true，整个表达式值为x，否则整个表达式值为y

③示例代码如下：

|  |
| --- |
| //三目运算符  int i71 = 71; //声明int类型变量i71并赋值  int i72 = 72; //声明int类型变量i72并赋值  System.out.println("三目运算值为：" + (i72>i71?i72:i71)); |

7、位运算符

①按位与“&”，运算法则是：如果两个整型数据对应位都是1，结果位为1，否则为0。如果两个数据精度不同，结果的精度与精度高的数据相同。

②按位或“|”，运算法则是：如果两个数据对应为都是0，结果位为0，否则为1。如果两个数据的精度不同，结果的精度与精度高的数据相同。

③按位取反“~”，运算法则是：数据的二进制中的1改为0，0改为1。

④按位异或“^”，运算法则是：如果两个数据的二进制同时为0或同时为1时，结果为0，否则为1。如果两个数据的精度不同，结果的精度与精度高的数据相同。

⑤移位操作，对数据按二进制位进行移位操作，有三种移位，分别为：左移“<<”、右移“>>”、无符号右移“>>>”。（注：左移就是将运算符左边的数据的二进制数据，按照运算符右边数据指定的位数向左移动，右边移空的部分补0。右移，当使用“>>”符号时，如果最高位是0，右移空的位就填入0；如果最高位是1，右移空的位就填入1。无符号“>>>”右移，无论最高位是0还是1，左侧被移空的高位都填入0。移位运算符适用的数据类型有byte、short、char、int、long。）

8、运算符优先级

①运算符的优先级由高到低的顺序依次是：增量和减量运算、算术运算、比较运算、逻辑运算、赋值运算

②两个运算符相同的优先级，左边的表达式比右边的表达式先运算

1. 本小节结束语

希望初学者认真操作练习，以达到熟练掌握。