1. 基本类型包装类
2. 定义

将基本的数据类型封装成一个类，可以在该类中定义更多的功能方法来操作该数据类型。

1. 常用操作

基本数据类型与字符串之间的转换

（三）包装类

基本数据类型 包装类

byte Byte

short Short

int Integer

long Long

float Float

double Double

char Character

boolean Boolean

1. Integer类
2. 定义

将原始类型int包装成类

1. 构造方法
2. public static Integer valueOf(int i)

返回表示指定的 int 值的 Integer 实例

1. public static Integer valueOf(String s)

返回一个保存指定值的 Integer 对象 String

1. 使用示例

public class Integer类 {  
 public static void main(String[] args) {  
 Integer i1 = Integer.*valueOf*(100);  
 System.*out*.println(i1);  
  
 Integer i2 = Integer.*valueOf*("100");  
 System.*out*.println(i2);  
 }  
}

1. int和String类型的相互转换
2. int转为String
3. int后直接加空字符串
4. String类的静态方法valueOf()

示例：

public class int转String {  
 public static void main(String[] args) {  
 int num = 200;  
 String s1 = num + "";  
 System.*out*.println(s1);  
  
 String s2 = String.*valueOf*(num);  
 System.*out*.println(s2);  
 }  
}

1. String转int类型
2. 字符串转成Integer，在调用valueOf方法
3. Integer静态方法parseInt()

示例：

public class String转int {  
 public static void main(String[] args) {  
 String s = "100";  
 Integer i = Integer.*valueOf*(s);  
 int x = i.intValue();  
 System.*out*.println(x);  
  
 int y = Integer.*parseInt*(s);  
 System.*out*.println(y);  
 }  
}

1. 装箱和拆箱

装箱：将基本数据类型转换为对应的包装类类型

拆箱：将包装类类型转换为对应的基本数据类型

自动装箱与拆箱，不需要调用方法，直接赋值操作即可。

如：

手动装箱：

Integer i1 = Integer.value(100);

自动装箱：

Integer i2 = 100;

手动拆箱：

Int i3 = i1.intValue() + 100;

自动拆箱：

Int i4 = i1 + 100;

注意：在使用包装类时，需要进行判断null的操作。