Java语言支持的变量类型有：

1. 类变量(静态变量)：独立于方法之外的变量，用 static 修饰。
2. 实例变量：独立于方法之外的变量，不过没有 static 修饰。
3. 局部变量：类的方法中的变量。

实例变量与静态变量的区别：

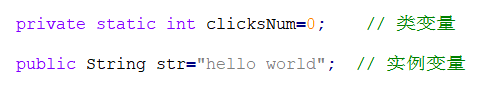
1. 静态变量属于类，该类不生产对象，通过类名就可以调用静态变量。
2. 实例变量属于该类的对象，必须产生该类对象，才能调用实例变量。

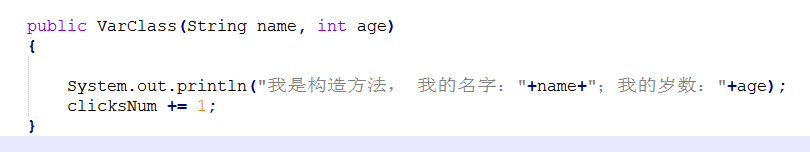
在程序运行时的区别：

1. 实例变量属于某个对象的属性，必须创建了实例对象，其中的实例变量才会被分配空间，才能使用这个实例变量。
2. 静态变量不属于某个实例对象，而是属于类，所以也称为类变量，只要程序加载了类的字节码，不用创建任何实例对象，静态变量就会被分配空间，静态变量就可以被使用了。
3. 总之，实例变量必须创建对象后才可以通过这个对象来使用，静态变量则可以直接使用类名来引用。

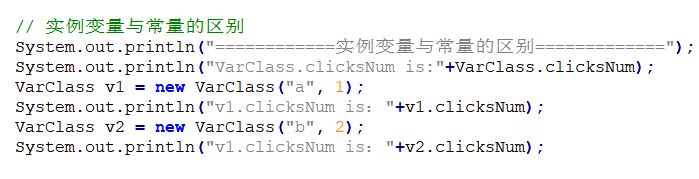
例如，对于下面的程序，无论创建多少个实例对象，

永远都只分配了一个 clicksNum变量，并且每创建一个实例对象，这个 clicksNum就会加 1；但是，每创建一个实例对象，就会分配一个 str变量，

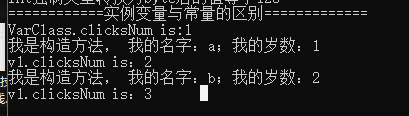












局部变量：

1. 局部变量是在栈上分配的。
2. 局部变量没有默认值，所以局部变量被声明后，必须经过初始化，才可以使用。
3. 局部变量不设置默认值（不初始化）会报错。

实例变量：

1. 实例变量声明在一个类中，但在方法、构造方法和语句块之外；
2. 当一个对象被实例化之后，每个实例变量的值就跟着确定；
3. 实例变量在对象创建的时候创建，在对象被销毁的时候销毁；
4. 实例变量的值应该至少被一个方法、构造方法或者语句块引用，使得外部能够通过这些方式获取实例变量信息；
5. 实例变量可以声明在使用前或者使用后；
6. 访问修饰符可以修饰实例变量；
7. 实例变量对于类中的方法、构造方法或者语句块是可见的。一般情况下应该把实例变量设为私有。通过使用访问修饰符可以使实例变量对子类可见；
8. 实例变量具有默认值。数值型变量的默认值是0，布尔型变量的默认值是false，引用类型变量的默认值是null。变量的值可以在声明时指定，也可以在构造方法中指定；
9. 实例变量可以直接通过变量名访问。但在静态方法以及其他类中，就应该使用完全限定名：ObejectReference.VariableName。

类变量（静态变量）

1. 类变量也称为静态变量，在类中以 static 关键字声明，但必须在方法之外。
2. 无论一个类创建了多少个对象，类只拥有类变量的一份拷贝。
3. 静态变量除了被声明为常量外很少使用，静态变量是指声明为 public/private，final 和 static 类型的变量。静态变量初始化后不可改变。
4. 静态变量储存在静态存储区。经常被声明为常量，很少单独使用 static 声明变量。
5. 静态变量在第一次被访问时创建，在程序结束时销毁。
6. 与实例变量具有相似的可见性。但为了对类的使用者可见，大多数静态变量声明为 public 类型。
7. 默认值和实例变量相似。数值型变量默认值是 0，布尔型默认值是 false，引用类型默认值是 null。变量的值可以在声明的时候指定，也可以在构造方法中指定。此外，静态变量还可以在静态语句块中初始化。
8. 静态变量可以通过：ClassName.VariableName的方式访问。
9. 类变量被声明为 public static final 类型时，类变量名称一般建议使用大写字母。如果静态变量不是 public 和 final 类型，其命名方式与实例变量以及局部变量的命名方式一致。