* 二进制兼容性 : 低版本编译的代码可在高版本JDK运行，反之不行
* 源代码兼容性 : 使用高版本JDK新特性不能再低版本上编译。
* 行为兼容性 : 代码里有bug，如果JDK版本变了，行为改变
* 类文件变化 : JDK7编译的类文件版本号为51.0，JDK8编译的为52.0
* Java SE 与上一版本相比不兼容的地方
* JDK 与上一版本相比不兼容的地方
* Java SE 中移除的特性
* JDK 中移除的特性
* 不建议使用的 API

**Jdk6-jdk7改动的地方：**

异常抛出不同

javax.lang.model.type.MirroredTypeException

 java.lang.Float.parseFloat(String) /parseDouble

java.util.TreeMap/TreeSet

Formatter.format()

javax.lang.model.type.TypeVisitor继承结构发生了变化

 java.lang.Character.isLowerCase/isUpperCase 变成遵循Unicode Standard

Io相关

设置java.io.File.setReadOnly  时不再支持目录的设置

setWriteable

Nio相关

java.nio.channels.\*

java.nio.channels.DatagramChannel()

awt相关

 java.awt.Color

MouseEvent.getButton()

Windows.setBackground

Toolkit.getPrintJob(Frame, String, Properties)

sun.awt.exception.handler

com.sun.image.codec.jpeg

Locale.getDefault(Locale.Category)

Jmx相关

————增加可靠事件处理

javax.management.event

————访问控制

jmx.remote.x.access.file

新增com.sun.management.jmxremote.local.only 在jdk6中默认接受本地远程连接请求，JDK7需设置此属性为false才能接收远程请求。

JDBC相关

新增 RowSetFactory

新增ava.sql.Connection, java.sql.Driver,

javax.sql.CommonDatasource, java.sql.Statement

JAXP相关

移除了 XSLTProcessorApplet

SOAP消息中，不再支持DTD

Arrays和Collections 所有子类的iterator()的实现发生了改变，可能导致IllegalArgumentException

编译相关

java.lang.Class 新增 getMethods getDeclaredMethods getDeclaredConstructors

泛型方法推断：1 根据实际参数类型推断方法参数类型 2根据边界和上下文的约束推断没有传入前一阶段没有推断出来的变量

Test<?> t = new Test<>(); //不再支持

class X<Y extends Integer> { //内部类的泛型,T不是Integer子类就报错

class Y {}

class Y1<T extends X<Y>> { }

}

public class A { // 泛型的擦除，JDK7开始不支持用泛型来区分参数

int m(List<String> ls) { return 0;}

long m(List<Integer> ls) {return 1; }

}

//泛型与重载结合使用，发生了微妙的变化

class A {private int val = 42; } //泛型的继承，不再使用父类的private属性

class Test<X extends A> {

void test(X x) {int i = x.val; //error in JDK 7 }}

可能会出问题的地方：

泛型的推断、异常处理、JDK8不兼容 spring2.5.6