书上的例子代码如下：

[复制代码](javascript:void(0);)

public class MyPair <T>{

private T first;

private T second;

public MyPair(){ first = null; second = null;}

public MyPair(T first, T second){ this.first = first; this.second = second;}

public T getFirst(){ return first;}

public T getSecond() {return second;}

public void setFirst(T value){ first = value;}

public void setSecond(T value) { second = value;}

}

[复制代码](javascript:void(0);)

public class DateInterval extends MyPair<Date> {

public void setSecond(Date second) {

if(second.compareTo(getFirst()) >= 0)

super.setSecond(second);

}

}

[复制代码](javascript:void(0);)

public static void main(String[] args) throws InterruptedException {

// TODO Auto-generated method stub

DateInterval interval = new DateInterval();

interval.setFirst(new Date());

interval.setSecond(new Date());

System.out.println("second value of interval: " + interval.getSecond().toString());

Thread.sleep(10);

MyPair<Date> datePair = interval;

datePair.setSecond(new Date());

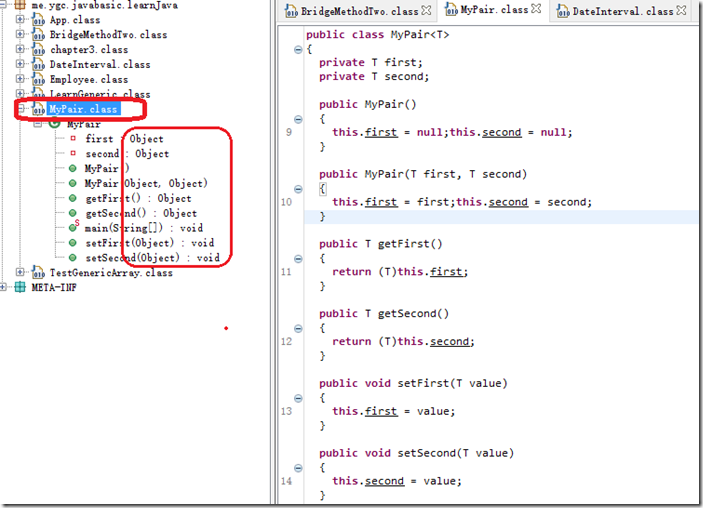
System.out.println("second value of interval: " + datePair.getSecond().toString());

}

[复制代码](javascript:void(0);)

一、通过jd-gui.exe来分析字节码，只能看到类型擦除信息

上网查阅了一些资料还是不明白，然后觉得应该可以通过反编译工具来看，于是找了jd-gui.exe来看，发现反编译出来的东西和原始的类基本相同的，如下，关于书上提到的类型擦除倒是确实存在，可以看到在字节码中其实没有泛型，而是做了类型擦除之后的类型。

[](http://images2015.cnblogs.com/blog/12617/201512/12617-20151204153316783-686707559.png)

[复制代码](javascript:void(0);)

public static void main(String[] args)

throws InterruptedException

{

DateInterval interval = new DateInterval();

interval.setFirst(new Date());

interval.setSecond(new Date());

System.out.println("second value of interval: " + ((Date)interval.getSecond()).toString());

Thread.sleep(10L);

MyPair<Date> datePair = interval;

datePair.setSecond(new Date());

System.out.println("second value of interval: " + ((Date)datePair.getSecond()).toString());

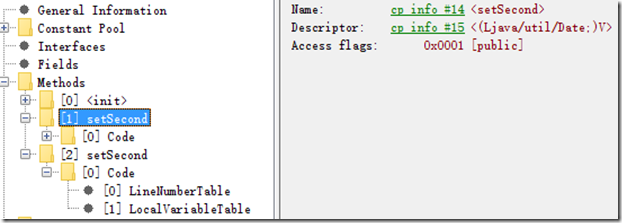
}

[复制代码](javascript:void(0);)

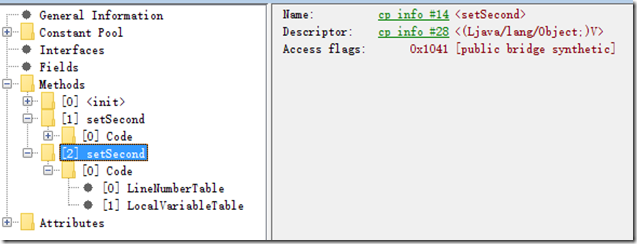
二、用jclasslib来看字节码

    感觉可能是jd-gui.exe太高级了，反编译做过头了，结果把我需要的信息都过滤掉了，所以找了稍微更原始反编译工具，jclasslib，使用它打开DateInternal.class文件之后，可以看到如下信息。在上面的源码中我们实际上只给DateInternal添加了一个setSecon方法，但是在反编译之后发现可以看到两个setSecond方法，两个方法的信息分别如下。

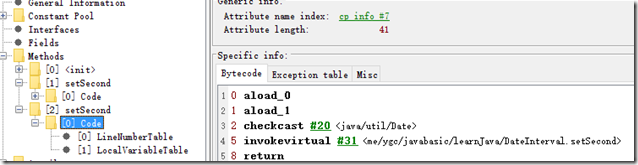
1）第一个就是我们在源码中定义的setSecond，入参为Date类型的setSecond方法；

[](http://images2015.cnblogs.com/blog/12617/201512/12617-20151204153328471-502898808.png)

2）第二个就是书上所说的桥接方法，可以看到这个方法的flag中，除了有public，还有bridge，synthetic两个标志，这表示这个是由编译器自动生成的桥接方法。

[](http://images2015.cnblogs.com/blog/12617/201512/12617-20151204153330877-2135235722.png)

3）在看看方法的内容，其实内部调用了DateInterval.setSecond方法，并且在

[](http://images2015.cnblogs.com/blog/12617/201512/12617-20151204153339596-142886002.png)

三、也可以使用javap命令来查看字节码信息

在命令行输入javap -c -v DateInternal.class

则会输出如下信息，这里看到的信息和jclaslib看到的类似。

[复制代码](javascript:void(0);)

Classfile /D:/java/eclipse/learnJava/target/classes/me/ygc/javabasic/learnJava/DateInterval.class

Last modified 2015-12-1; size 771 bytes

MD5 checksum f8d67b651cd0aa143e3fbe03c5edd519

Compiled from "DateInterval.java"

public class me.ygc.javabasic.learnJava.DateInterval extends me.ygc.javabasic.learnJava.MyPair<java.util.Date>

minor version: 0

major version: 49

flags: ACC\_PUBLIC, ACC\_SUPER

Constant pool:

#1 = Class #2 // me/ygc/javabasic/learnJava/DateInterval

#2 = Utf8 me/ygc/javabasic/learnJava/DateInterval

#3 = Class #4 // me/ygc/javabasic/learnJava/MyPair

#4 = Utf8 me/ygc/javabasic/learnJava/MyPair

#5 = Utf8 <init>

#6 = Utf8 ()V

#7 = Utf8 Code

#8 = Methodref #3.#9 // me/ygc/javabasic/learnJava/MyPair."<init>":()V

#9 = NameAndType #5:#6 // "<init>":()V

#10 = Utf8 LineNumberTable

#11 = Utf8 LocalVariableTable

#12 = Utf8 this

#13 = Utf8 Lme/ygc/javabasic/learnJava/DateInterval;

#14 = Utf8 setSecond

#15 = Utf8 (Ljava/util/Date;)V

#16 = Methodref #1.#17 // me/ygc/javabasic/learnJava/DateInterval.getFirst:()Ljava/lang/Object;

#17 = NameAndType #18:#19 // getFirst:()Ljava/lang/Object;

#18 = Utf8 getFirst

#19 = Utf8 ()Ljava/lang/Object;

#20 = Class #21 // java/util/Date

#21 = Utf8 java/util/Date

#22 = Methodref #20.#23 // java/util/Date.compareTo:(Ljava/util/Date;)I

#23 = NameAndType #24:#25 // compareTo:(Ljava/util/Date;)I

#24 = Utf8 compareTo

#25 = Utf8 (Ljava/util/Date;)I

#26 = Methodref #3.#27 // me/ygc/javabasic/learnJava/MyPair.setSecond:(Ljava/lang/Object;)V

#27 = NameAndType #14:#28 // setSecond:(Ljava/lang/Object;)V

#28 = Utf8 (Ljava/lang/Object;)V

#29 = Utf8 second

#30 = Utf8 Ljava/util/Date;

#31 = Methodref #1.#32 // me/ygc/javabasic/learnJava/DateInterval.setSecond:(Ljava/util/Date;)V

#32 = NameAndType #14:#15 // setSecond:(Ljava/util/Date;)V

#33 = Utf8 SourceFile

#34 = Utf8 DateInterval.java

#35 = Utf8 Signature

#36 = Utf8 Lme/ygc/javabasic/learnJava/MyPair<Ljava/util/Date;>;

{

public me.ygc.javabasic.learnJava.DateInterval();

descriptor: ()V

flags: ACC\_PUBLIC

Code:

stack=1, locals=1, args\_size=1

0: aload\_0

1: invokespecial #8 // Method me/ygc/javabasic/learnJava/MyPair."<init>":()V

4: return

LineNumberTable:

line 6: 0

LocalVariableTable:

Start Length Slot Name Signature

0 5 0 this Lme/ygc/javabasic/learnJava/DateInterval;

public void setSecond(java.util.Date);

descriptor: (Ljava/util/Date;)V

flags: ACC\_PUBLIC

Code:

stack=2, locals=2, args\_size=2

0: aload\_1

1: aload\_0

2: invokevirtual #16 // Method getFirst:()Ljava/lang/Object;

5: checkcast #20 // class java/util/Date

8: invokevirtual #22 // Method java/util/Date.compareTo:(Ljava/util/Date;)I

11: iflt 19

14: aload\_0

15: aload\_1

16: invokespecial #26 // Method me/ygc/javabasic/learnJava/MyPair.setSecond:(Ljava/lang/Object;)V

19: return

LineNumberTable:

line 8: 0

line 9: 14

line 10: 19

LocalVariableTable:

Start Length Slot Name Signature

0 20 0 this Lme/ygc/javabasic/learnJava/DateInterval;

0 20 1 second Ljava/util/Date;

public void setSecond(java.lang.Object);

descriptor: (Ljava/lang/Object;)V

flags: ACC\_PUBLIC, ACC\_BRIDGE, ACC\_SYNTHETIC

Code:

stack=2, locals=2, args\_size=2

0: aload\_0

1: aload\_1

2: checkcast #20 // class java/util/Date

5: invokevirtual #31 // Method setSecond:(Ljava/util/Date;)V

8: return

LineNumberTable:

line 1: 0

LocalVariableTable:

Start Length Slot Name Signature

}

SourceFile: "DateInterval.java"

Signature: #36 // Lme/ygc/javabasic/learnJava/MyPair<Ljava/util/Date;>;

[复制代码](javascript:void(0);)

四、通过代码来验证桥接方法的存在

如果编写如下代码：

public static void main(String[] args) throws InterruptedException {

// TODO Auto-generated method stub

MyPair datePair = new DateInterval();

datePair.setSecond(new Object());

}

运行之后会提示如下：

Exception in thread "main" java.lang.ClassCastException: java.lang.Object cannot be cast to java.util.Date

    at me.ygc.javabasic.learnJava.DateInterval.setSecond(DateInterval.java:1)

    at me.ygc.javabasic.learnJava.MyPair.main(MyPair.java:26)

说明他实际上是去调用了一个setSecond(Object)的方法，然后在内部做了从Object到Date的转换，然后转换失败了。