

java实现的18位身份证格式验证算法 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/144/2021\\_2022\\_java\\_E5\\_AE\\_9E\\_E7\\_8E\\_B0\\_c104\\_144454.htm](https://www.100test.com/kao_ti2020/144/2021_2022_java_E5_AE_9E_E7_8E_B0_c104_144454.htm) 公民身份号码是特征组合码,由十七位数字本体码和一位数字校验码组成.排列顺序从左至右依次为:六位数字地址码,八位数字出生日期码,三位数字顺序码和一位数字校验码。 1、地址码 表示编码对象常住户口所在县（市、旗、区）的行政区划代码，按 GB/T 2260 的规定执行。 2、出生日期码 表示编码对象出生的年、月、日，按 GB/T 7408 的规定执行。年、月、日代码之间不用分隔符。

例：某人出生日期为 1966年10月26日，其出生日期码为 19661026。 3、顺序码 表示在同一地址码所标识的区域范围内，对同年、同月、同日出生的人编定的顺序号，顺序码的奇数分配给男性，偶数千分配给女性。 4、校验码 校验码采用ISO 7064：1983，MOD 11-2 校验码系统。（1）十七位数字本体码加权求和公式  $S = \text{Sum}(A_i * W_i), i = 0, \dots, 16$ ，先对前17位数字的权求和  $A_i$ :表示第*i*位置上的身份证号码数字值  $W_i$ :表示第*i*位置上的加权因子  $W_i$ : 7 9 10 5 8 4 2 1 6 3 7 9 10 5 8 4

2 （2）计算模  $Y = \text{mod}(S, 11)$  （3）通过模得到对应的校验码  $Y$ : 0 1 2 3 4 5 6 7 8 9 10 校验码: 1 0 X 9 8 7 6 5 4 3 2 下面是java实现的

```
代码 /* * IDCard.java Created on 2004-11-5 17:03:37 * */
package org.yz21.study.idcard. /** * @author violin 2004-11-5
17:03:37 * Copyright www.yz21.org 2003-2004 */ public class
IDCard { // wi =2(n-1)(mod 11) final int[] wi =
{7,9,10,5,8,4,2,1,6,3,7,9,10,5,8,4,2,1}. // verify digit final int[] vi =
{1,0,X,9,8,7,6,5,4,3,2}. private int[] ai = new int[18]. public
```

```

IDCard() { } //verify public boolean Verify(String idcard) { if
(idcard.length() == 15) { idcard = uptoeighteen(idcard). } if
(idcard.length() != 18) { return false. } String verify =
idcard.substring(17, 18). if (verify.equals(getVerify(idcard))) {
return true. } return false. } //get verify public String getVerify(String
eightcardid) { int remaining = 0. if (eightcardid.length() == 18) {
eightcardid = eightcardid.substring(0, 17). } if (eightcardid.length()
== 17) { int sum = 0. for (int i = 0. i String k =
eightcardid.substring(i, i + 1). ai = Integer.parseInt(k). } for (int i = 0. i
sum = sum + ai * i. } remaining = sum % 11. } return remaining == 2
? "X" : String.valueOf(vi[remaining]). } //15 0update to 18 public
String uptoeighteen(String fifteencardid) { String eightcardid =
fifteencardid.substring(0,6). eightcardid = eightcardid + "19".
eightcardid = eightcardid + fifteencardid.substring(6,15). eightcardid =
eightcardid + getVerify(eightcardid). return eightcardid. } } 测试代码
: 使用的单元测试工具是junit /* * IDCardTest.java Created on
2004-11-5 17:32:12 * */ package org.yz21.study.idcard. import
junit.framework.Test. import junit.framework.TestCase. import
junit.framework.TestSuite. /** * @author violin 2004-11-5 17:32:12 *
Copyright www.yz21.org 2003-2004 */ public class IDCardTest
extends TestCase { private String idcard1 = "11010519491231002X".
private String idcard2 = "440524188001010014". public void
testVerify() { IDCard idcard = new IDCard().
this.assertTrue(idcard.Verify(idcard1)).
this.assertTrue(idcard.Verify(idcard2)). } public static Test suite() {
return new TestSuite(IDCardTest.class). } public static void

```

```
main(String[] args) { junit.textui.TestRunner.run(suite()). } }
```

100Test 下载频道开通，各类考试题目直接下载。详细请访问

[www.100test.com](http://www.100test.com)