<Next Date>

测试评估报告

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <11日/5月/2019年> | <1.0> | <测试报告> | <原帅> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 3

1.1 目的 3

1.2 范围 3

1.3 定义、首字母缩写词和缩略语 3

1.4 参考资料 3

1.5 概述 3

2. 测试结果摘要 3

3. 基于代码的测试覆盖 33

测试评估报告

# 简介

## 目的

根据NextDate程序源码，结合intellij idea的测试覆盖度报告，对测试情况作出评估。

## 范围

Nextdate程序及其测试代码

## 定义、首字母缩写词和缩略语

无

## 参考资料

Nextdate程序， 来源 github

NextDate测试代码， 本小组编写

NextDate程序测试计划， 本小组编写

## 概述

本次测试是基于类方法为粒度的单元测试，使用白盒测试中的路径覆盖的准则来设计测试用例。报告中展示了intellij idea生成的测试覆盖度，并分析了基于代码的测试覆盖程度。

# 测试结果摘要

本次测试通过率为100%，所有测试用例均顺利通过。

测试实现了条件覆盖，即Miller准则中的C1p

# 基于代码的测试覆盖

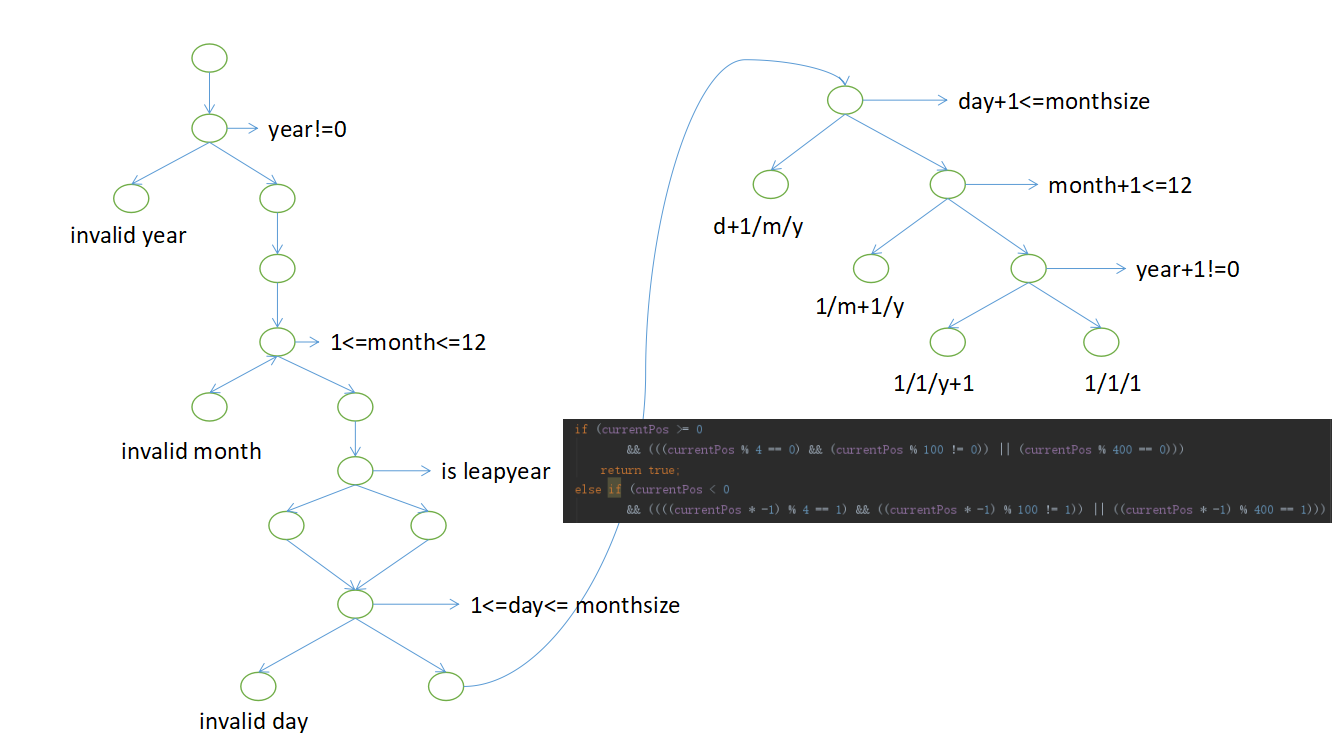
Intelij idea给出的覆盖度报告如下表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element | Class | Method | Line | Branch |
| CalendarUnit | 100% (1/1) | 100% (2/2) | 100% (3/3) | 100% (0/0) |
| Date | 100% (1/1) | 100% (8/8) | 100% (20/20) | 100% (6/6) |
| Day | 100% (1/1) | 100% (6/6) | 100% (20/20) | 100% (9/9) |
| Main | 100% (1/1) | 100% (1/1) | 100% (7/7) | 100% (0/0) |
| Month | 100% (1/1) | 100% (7/7) | 100% (25/25) | 100% (10/10) |
| Year | 100% (1/1) | 100% (7/7) | 100% (22/22) | 100% (13/13) |

其中，LIne列的覆盖度表示满足了语句覆盖标准，测试覆盖到了所有类的所有代码。

Branch列指的是条件覆盖，即每个类方法中的判断语句中，每个判断条件的正确和错误情况都覆盖到了。

其中，对于Main类，由于设计测试用例时，通过将调用变为inline的方式，跟踪了NextDate方法实际调用的代码路径（如下图）。因此对main的覆盖度不能通过上表体现出来，具体分析如下。



从上图看出，nextdate方法实际执行中涉及到7次判断语句，分别涉及到的条件为：

1. year=0， year!=0
2. Month<1, 1<= month <=12, month >=12
3. year>=0 时，

Year %400=0； year& 100=0且year%400!=0； year%4 =0且year %100!=0； year%4 !=0

Year<0时，

year%400=1; year%400!=1且year%100=1; year%4=1且year%100!=1; year%4!=1.

4.day<1; 1<=day<=monthsize; day>monthsize

5.day=monthsize; day<=monthsize

6. month=12; month<12

7.year=-1;year!=-1;

测试用例如下：

（month，day，year）

（1，1，0）

（0，1，1）

（13，1，1）

（1，0，1）

（2，28，1600）

（2，28，2100）

（2，14，114514）

（12，31，2004）

（12，31，-1）

（3，31，-16401）

（2，14，-301）

（2，28，-5）

结合上面的7个条件的分析，这些测试用例能够覆盖到所有条件，因此Main的测试也满足了 条件覆盖。