Department of Computer Engineering
est. 1989
Kasetsart University

Software Test

Lesson 7
Homework – Feedback

Uwe Gühl
Winter 2015 / 2016

Homework

1. Write 5 unit test cases for a program to determine whether a given year is a leap year or not.
   Please explain why you have chosen the tests.
   Preferred tool is JUnit, but any other tool is accepted as well.
2. Send them in a neutral format (.txt, .pdf) until 16.03.2016 to uweguehl@hotmail.com

Homework – Discussion

- Which years to be considered?
  Recommended: Don’t do “implementation related tests” like ..
  - If February has 29 days, than the year must be a leap year
  - If the year has 366 days, than the year must be a leap year
- Is year 0 a valid year?
  If so:
  Is year 0 a leap year?
- Could negative years be leap years?

Homework – Discussion

Method to determine whether a year is a leap year*

- In the Gregorian calendar, first established in 1582 by Pope Gregory XIII, a normal year consists of 365 days. Because the actual length of a sidereal year (the time required for the Earth to revolve once about the Sun) is actually 365.25635 days.
- A "leap year" of 366 days is used once every four years to eliminate the error caused by three normal (but short) years. Any year that is evenly divisible by 4 is a leap year: for example,
  1988, 1992, and 1996 are leap years.
- However, there is still a small error that must be accounted for. To eliminate this error, the Gregorian calendar stipulates that a year that is evenly divisible by 100 (for example, 1900) is a leap year only if it is also evenly divisible by 400.
  - For this reason, the following years are not leap years:
    1700, 1800, 1900, 2100, 2200, 2300, 2500, 2600
    This is because they are evenly divisible by 100 but not by 400.
  - The following years are leap years:
    1600, 2000, 2400
    This is because they are evenly divisible by both 100 and 400.

* Source: https://support.microsoft.com/en-us/kb/214019

Homework – Discussion

Could negative years be leap years?**

- The year 0 is that in which one supposes that Jesus Christ was born, which several chronologists mark 1 before the birth of Jesus Christ and which we marked 0, so that the sum of the years before and after Jesus Christ gives the interval which is between these years, and where numbers divisible by 4 mark the leap years as so many before or after Jesus Christ.

— *Jacques Cassini, Tables astronomiques, 5, translated from French*

** https://en.wikipedia.org/wiki/0_%28year%29

Homework – Proposal

Proposal: Testing years that could be divided by 4
2020, 0, -2016

```
@Test
public void testDetermineLeapYearPositive4_2020() {
        assertTrue(“2020 is a leap year as divisible by 4",
        year.isLeapYear(2020));
}
@Test
public void testDetermineLeapYearPositive_0() {
        assertTrue("0 is a leap year as divisible by 4",
        year.isLeapYear(0));
}

@Test
public void testDetermineLeapYearPositive4_NegYear() {
        assertTrue(“-2016 is a leap year as divisible by 4",
        year.isLeapYear(-2016));
}
```

Homework – Proposal

Proposal: Testing years that could be not be divided by 4
2017, -2017

```
@Test
public void testDetermineLeapYearNegative_2017() {
        assertFalse("2017 is not a leap year",
        year.isLeapYear(2017));
}
@Test
public void testDetermineLeapYearNegative_NegYear() {
        assertFalse("-2017 is not a leap year",
        year.isLeapYear(-2017));
}
```

Homework – Proposal

Proposal: Testing years that are not leap years as they

- Could be divided by 4
- But could be divided by 100 AND **NOT** by 400.

-1900, 2100

```
@Test
public void testDetermineLeapYearNegative_2100() {
         assertFalse("2100 is not a leap year; divisible by 4, but by 100
and not by 400",
         year.isLeapYear(2100));
}

@Test
public void testDetermineLeapYearNegative_NegYear1900() {
         assertFalse("-1900 is not a leap year; divisible by 4, but by 100
and not by 400",
         year.isLeapYear(-1900));
}
```

Homework – Proposal

Proposal: Testing years that are leap years as they

- Could be divided by 4
- But could be divided by 100 AND by 400.

2000, -10.000

```
@Test
public void testDetermineLeapYearNegative100AND400_2000() {
         assertTrue("2000 is a leap year; divisible by 4, by 100, and by
         400",
         year.isLeapYear(2000));
}

@Test
public void testDetermineLeapYearPositive400_NegYear() {
         assertTrue("-100000 is a leap year; divisible by 4, by 100, and by
400",
         year.isLeapYear(-100000));
}
```

Homework
Possible implementation (1)

```
@package LeapYear;

public class LeapYearDetermination {

          public boolean isLeapYear (int year) {
                  boolean isLeapYear = false;

                  if(year % 400 == 0) {
                      isLeapYear = true;
                  }
                  else if (year % 100 == 0) {
                      isLeapYear = false;
                  }
                  else if(year % 4 == 0) {
                      isLeapYear = true;
                  }
                  else {
                      isLeapYear = false;
                  }
                  return isLeapYear;
          }
}
```

Homework
Possible implementation (2)

```
@package LeapYear;
import java.util.GregorianCalendar;

public boolean isLeapYear (int year) {
   // create a new calendar
   GregorianCalendar cal = (GregorianCalendar)
        GregorianCalendar.getInstance();
   return cal.isLeapYear(year);
}
```

Homework – Feedback

Additional ideas:

- Testing if input values are floats, textual characters instead of digits
  → If program reads string