Mandatory Assignment 0
Exercise 9

Gustav Metnik-Beck

September 9, 2021

1 The Leap Year Algorithm

1.1 Introduction

Figure 1 depicts the current implementation of the Leap Year algorithm used for telling whether a given year is a leap year or not.

A simplified definition of a Leap Year is: "*Every year that is exactly divisible by four is a leap year, except for years that are exactly divisible by 100, but these centurial years are leap years if they are exactly divisible by 400. For example, the years 1700, 1800, and 1900 are not leap years, but the years 1600 and 2000 are.*".

This definition has been transformed into the algorithm depicted as the last two diamonds in the diagram of figure 1. Whereas the first two diamonds represent error handling for wrong input years.

1.2 Explaining the Diagram

First, the algorithm needs input, which is depicted as a parallelogram. This parameter is an integer representing the year to be determined if is a leap year or not. This parameter is simply called "*year*" in the algorithm.

The next step is checking the input for errors. The first diamond depicts an if statement to check whether or not the input year is less than or equal to zero to accommodate the divisible by four rule, moreover we do not have negative years.

If the algorithm receives a zero or a negative year (the statement is *true*), it will throw an *ArgumentException* terminating the algorithm.

Else it will keep going to the next if statement if it comes out as *false*.

Next up, it is checked whether or not the input year meets the requirement that the algorithm should only apply to years from 1852. The requirement is depicted as the variable name "*minYear*".

If the input year is less than the minimum year (1852) it will throw an *ArgumentException* terminating the algorithm.

Else it will go to the next step. Another if statement is then checking if the first two rules of Leap Year are followed: "*Every year that is exactly divisible by four is a leap year, except for years that are exactly divisible by 100*".

This means that the year has to be divisible by four, which is done by the modulus operation by checking if it equals to zero. The modulus operation returns an integer which represents the remainder of the divide operation. Therefore, if it equals to zero, it is exactly divisible by four.

Moreover, the year cannot be exactly divisible by 100, which is depicted by the other modulus operation.

If the whole expression is true, the algorithm returns **true**, meaning that the input year is a Leap Year.

Else if it is false, it either means that the input year is not divisible by four or exactly divisible by 100 or both. This brings us to the last rule of Leap Year: "*Centurial years are leap years if they are exactly divisible by 400.*".

Therefore, it has to do a last check to see if it is a centurial year that has to be checked. This is depicted as the last diamond in the chain.

If the year is exactly divisible by 400 (again solved by modulus) the algorithm will return **true** else it will return **false**.

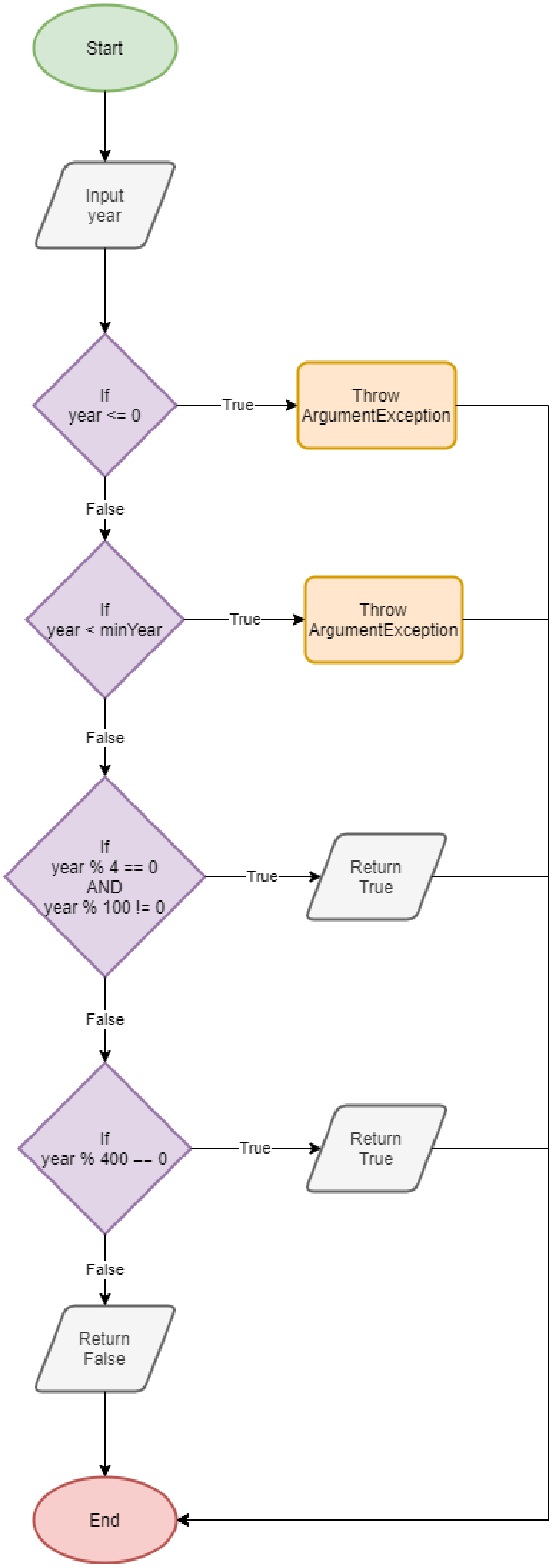

Figure 1: Flowchart over the "*IsLeapYear algorithm*", showing the progress from giving an input year to outputting either an exception or true/false depending on the year entered.