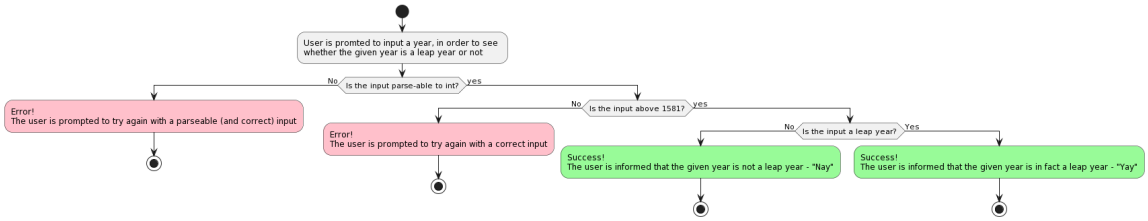
Assignment 00
Benjamin Millant



Figur 1: Activity Diagram of `IsInputLeapYear`, link to graph

The above graph is a representation of how the algorithm in Assignment00 works. For a better resolution of the graph follow the link in the figure text. The algorithm's purpose is to decide whether a given input is a leap year or not. In order to asses this, the algorithm performs certain checks in order to find the answer. A total of 3 checks are made:

1. Is the input parse-able?
2. Is the input inside the given range?
3. Is the input a leap year?

1: In order to check for leap year, the algorithm has to handle integers - since it is not possible to do arithmetic calculations on other types, like strings. And since the algorithm's purpose is to handle a *user's input*, it need to be able to parse the input into an integer. Therefore, the very first check is to see whether the given input *is* parse-able. This is decided by a function called `int.TryParse()` - if this method returns false, then the user id prompted to try again with a better input. If the method returns true, it moves forward to *2:* This check is necessary since the range for calculating leap years start at year 1582 - per definition of the assignment. This is a simple check handled by the method `isLeapYearWithinRange`. If the check return false, the user is prompted to try again with a year inside the range. If the check return true, the algorithm goes to the next and last check *3:* The final check is whether the input is a leap year or not. This is decided by the method `isLeapYear`. Depending on the return value, the user is informed either *"Yay"* or *"Nay"* and thus have a better understanding of whether their year is a leap year or not.