Family Tree Template Requests and Remarks

# Remarks From the Original Template “PfingstenBook2023.pdf”

* Inconsistency between dates.
  + I notice there are some inconsistencies in the initial template. These inconsistencies such as no spaces between the dates and sometimes using the entire word for months that could be part of a name makes the approach of parsing each section more complicated.
* Handling additional partnerships.
  + In the template, I notice that some additional partnerships are handled by only writing the name of member again while leaving out the birth date or deceased date and some write everything. The perspective of parsing makes it more complicated to come up with patterns of handling additional partnerships.
* Inconsistency use of ordering types
  + In terms of parsing, OOP (Object-Orientated Programing) languages such as C# are used to the concept of Abstraction, which is a pillar of OOP that motives the idea of implementing generalizations of provided information. In the template, the pattern I’m noticing is Roman-Upper for 1st generation, Capital-Letter for 2nd Generation, Digit for 3rd Generation, Lowercase-Letter for 4th Generation, Parenthesized-Digit for 5th Generation, and Roman-Lower for 6th Generation.
  + Also, in the template, with handling additional partnerships, I find it easier the apply abstraction when after the additional partnership is written, the ordering type is copied and any additional children should be assigned the next enumeration rather than restarting it.

# Template Requests in The Parsing Perspective

* Names and Dates
  + Have one space between each token of the name.
  + Have at least 2 spaces between each name and date attribute
    - 1 space is used to separate by token.
    - 1 space is used to determine a transition between attributes.
    - Other spaces will be trimmed.
  + Since some of the months can be part of name such as June, July, Sept, and Jan, I request to write Jan as “Jan.” and the rest as the 1st 3 letters.
    - Ex: Jun rather than June, Jul rather than July, etc.
    - In this way, it would be very convenient for the parser to distinguish between a token being part of a name and being part of a date.
  + Have one space between the day, month, and year.
    - Ex: Rather than “31May2017”, write “31 May 2017”
* Handling Additional Partnerships
  + Present the ordering type, birth date, and deceased date again.
    - I. August Fred Pfingsten 26 Jun 1896 14 Sep 1921 24 Aug 1980
    - I. August F. Pfingsten 26 Jun 1896 5 Jan. 1940 24 Aug 1980
      * The ordering type will help detect the additional marriage.
      * The birth date and deceased date will help with comparing people. (In my Person model class, I implemented the IComparable<Person> Interface which has a method called this.CompareTo(Person other) which is the abstraction of comparing the people and describing sequential order.
* Ordering Types
  + Maintain consistency between writing ordering types
    - Ex: I.,A.,1.,a.,(1),i)
  + If the ordering type would need to be repeated based on the previous request, then write the ordering type again, in the forward generation, write the next ordering type in the sequence. For example,