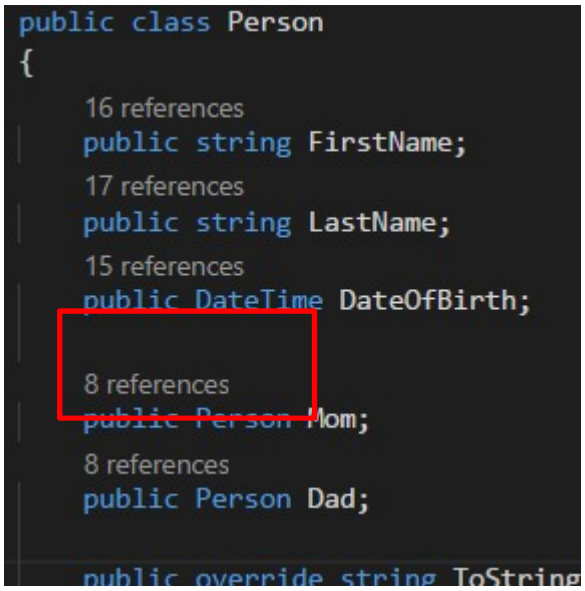
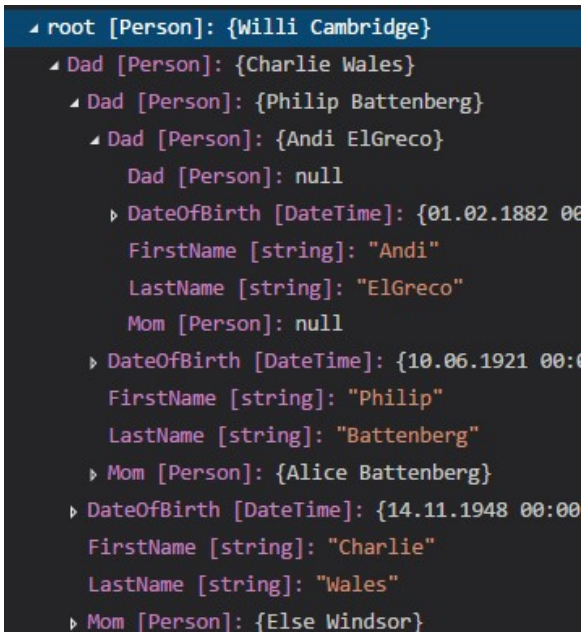
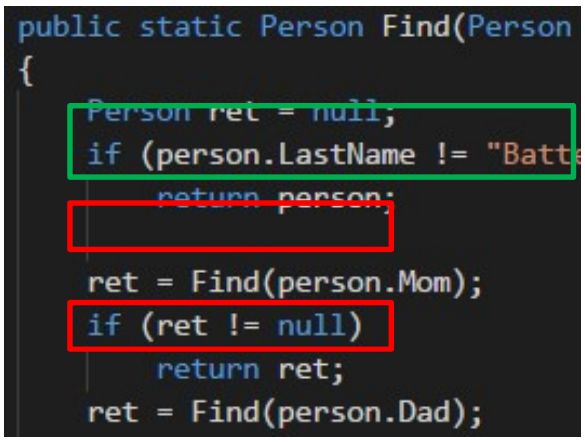
Softwaredesign Aufgabe 3 – Debugging



Hier wird für jede Instanz der Klasse Person eine Mom und ein Dad vom Typ Person angelegt – rekursive Klasse.



Es wird im Debugger die Person root angezeigt. Wenn man diese aufklappt erhält man eine Übersicht über alle ihre Eigenschaften, zu denen auch die oben beschriebenen Personen Mom und Dad gehören. Auch diese kann man wiederum weiter aufklappen um deren Eltern zu sehen. So lässt sich im Debugger der gesamte Stammbaum öffnen und untersuchen.



In der Funktion Find() wird auf eine Bedingung geprüft (grün). Ist diese falsch so ruft die Funktion sich selbst wieder auf, entweder mit der Mom oder der Dad Person (rot). So wird der ganze Stammbaum durchsucht, bis eine Person gefunden wird, auf die die Bedingung zutrifft.

```
public static Person Find(Person
{
    Person ret = null;
    if (person.LastName == "Batte
        return person;

    ret = Find(person.Mom);
    if (ret != null)
        return ret;
    ret = Find(person.Dad);
```

Ich habe nun die Bedingung in der Find() Funktion so geändert, dass nach einer Person gesucht wird, deren Nachname „Battenberg“ ist.

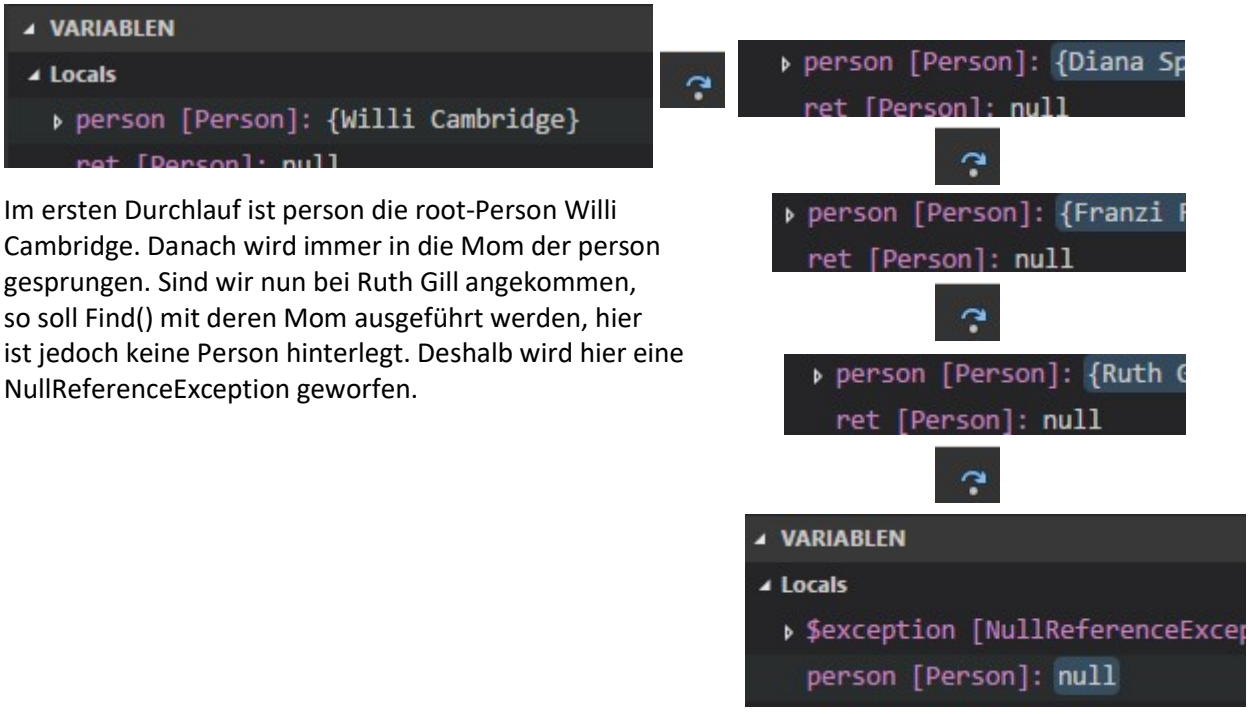
Wird das Programm nun ausgeführt, so tritt eine NullReferenceException auf.

```
PS D:\GitHub\SoftwaredesignMaterial\L03_Debugging> dotnet run

Unhandled Exception: System.NullReferenceException: Object reference not set to an instance of a
   at Debugging.Familytree.Find(Person person) in D:\GitHub\SoftwaredesignMaterial\L03_Debugging
   at Debugging.Familytree.Find(Person person) in D:\GitHub\SoftwaredesignMaterial\L03_Debugging
   at Debugging.Familytree.Find(Person person) in D:\GitHub\SoftwaredesignMaterial\L03_Debugging
   at Debugging.Familytree.Find(Person person) in D:\GitHub\SoftwaredesignMaterial\L03_Debugging
   at Debugging.Familytree.Find(Person person) in D:\GitHub\SoftwaredesignMaterial\L03_Debugging
```

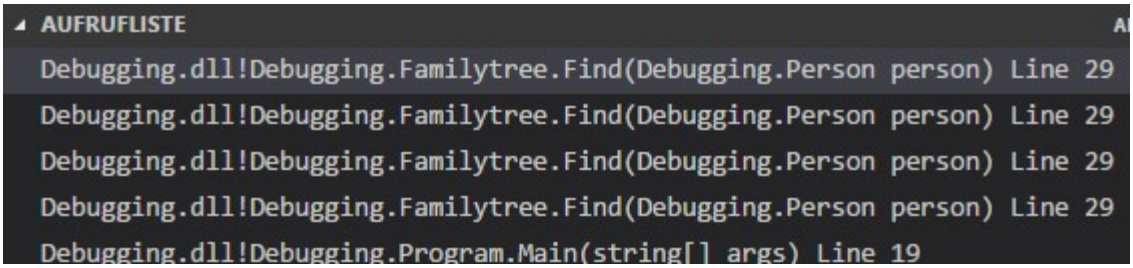
```
23          public static Person Find(Per
24          {
25              Person ret = null;
26              if (person.LastName == "B
27                  return person;
28
29              ret = Find(person.Mom);
30              if (ret != null)
31                  return ret;
32              ret = Find(person.Dad);
```

Um den Fehler zu finden habe ich in der FamilyTree.cs bei Zeile 29 einen Breakpoint gesetzt. Nun kann im Debugger beobachtet werden, welche Werte der Input-Parameter person und die Variable ret in jedem Durchlauf haben.



Im ersten Durchlauf ist person die root-Person Willi Cambridge. Danach wird immer in die Mom der person gesprungen. Sind wir nun bei Ruth Gill angekommen, so soll Find() mit deren Mom ausgeführt werden, hier ist jedoch keine Person hinterlegt. Deshalb wird hier eine NullReferenceException geworfen.

Call-Stack (Aufrufliste) beim Debuggen. Hier wird die Rekursivität der Funktion Find() nochmals deutlich:




```
public static Person Find(Person
{
    Person ret = null;
    if (person.LastName == "Batte
        return person;
    if(person.Mom != null)
        ret = Find(person.Mom);
    if (ret != null)
        return ret;
    if(person.Dad != null)
```

Um den Bug zu beheben habe ich vor dem rekursiven Aufruf der Methode eine Abfrage eingefügt, welche prüft und sicherstellt, dass der Parameter für die Find() Funktion nicht null ist. So kann eine NullReferenceException vermieden werden.

Die komplexere Bedingung zur Suche einer Person habe ich wie folgt umgesetzt: Man kann nun zusätzlich ein „minage“ und ein „maxage“ als Parameter beim Funktionsaufruf angeben und es wird die erste Person im Stammbaum zurückgegeben, deren Alter in diesem Bereich liegt.

```
public static Person FindPersonByAge(Person person, int minag
{
    int age = DateTime.Now.Year - person.DateOfBirth.Year;
    if (age >= minage && age <= maxage)
        return person;
    if(person.Mom != null)
        ret = FindPersonByAge(person.Mom, minage, maxage);
    if (ret != null)
        return ret;
    if(person.Dad != null)
```