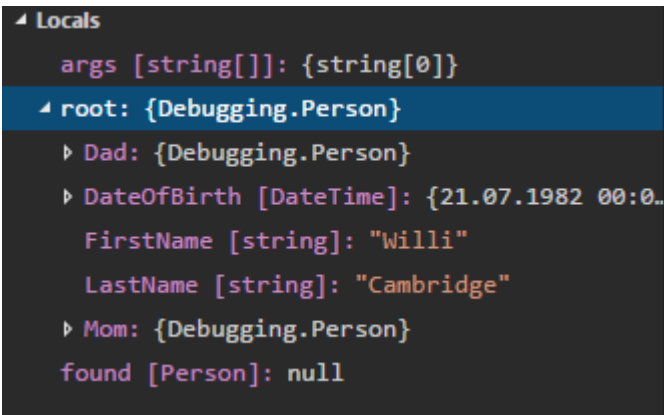


Sehr geehrte Leserinnen und Leser,
in dem Praktikum Aufgabe 2 Behandeln die Java Programme Program.cs und FamilyTree.cs.

Rekursion:

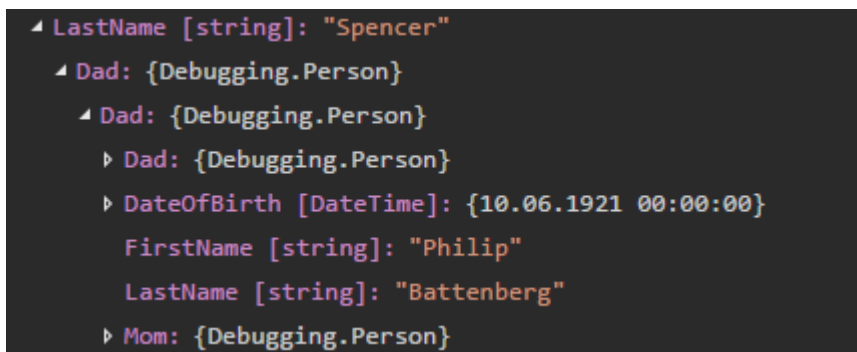
Die Programme werden als Rekursiv beschreiben da sie in sich selbst immer wieder aufrufen können.

Halte Punkt ab Zeile 19:



Der Code von Programm wird nun bis zu Zeile 18 ausgeführt.
Der Code, der bis jetzt ausgeführt wurde ist bis jetzt nur eine Root zu einer Person die aus der Datei FamilyTree.

Dad:



Klickt man im Debugger nun auf Dad so sehen wir jetzt schon wie der Code im Programm sich mindestens einem wieder aufruft also rekursiv ist.



Wenn wir nun den Code bis Zeile 22 laufen lassen sehen wir das die erste Ausgabe für Dad: Charie Wales ist. Um nicht die erste Person auszugeben, können wir durch die Änderung der Parameter erreichen. Tauschen wir das `!=` durch ein `==` und erstellen einem Try und Catch block zur Sicherheit, falls der Name nicht vorhanden seien sollte, können wir das Programm vor dem Absturz schützen.

```
└─ person: {Debugging.Person}
  └─ Dad: {Debugging.Person}
    └─ Dad: {Debugging.Person}
      └─ DateOfBirth [DateTime]: {14.11.1948 00:00:00}
        FirstName [string]: "Charlie"
        LastName [string]: "Wales"
      └─ Mom: {Debugging.Person}
        └─ DateOfBirth [DateTime]: {21.07.1982 00:00:00}
          FirstName [string]: "Willi"
          LastName [string]: "Cambridge"
```

Altersabfrage:

um das Alter einer Person Abzufragen reicht eine einfaches Abfragen des Alters einer Person die gerade Abgefragt wird.

```
{
    3 references
    public static Person Find(Person person)
    {
        var age = DateTime.Now.Year - person.DateOfBirth.Year;
        if ( age > 40)
        {
            WriteLine("First person in range of age >40: " + person.LastName);
            return person;
        }
    }
}
```