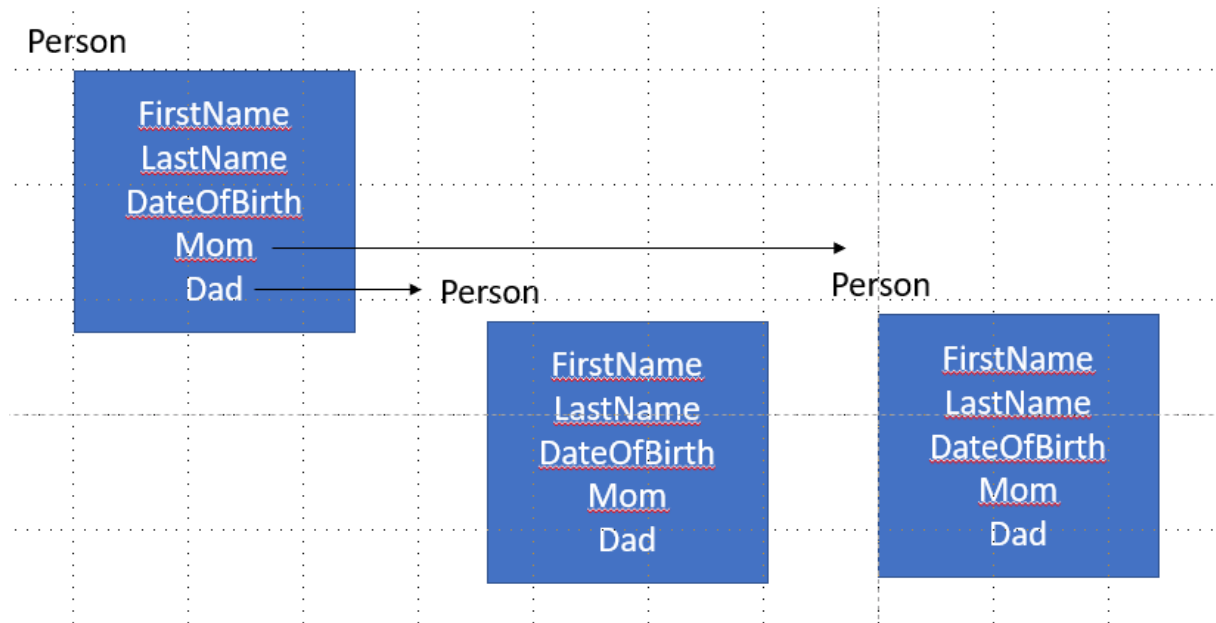


Übungsaufgabe - Protokoll

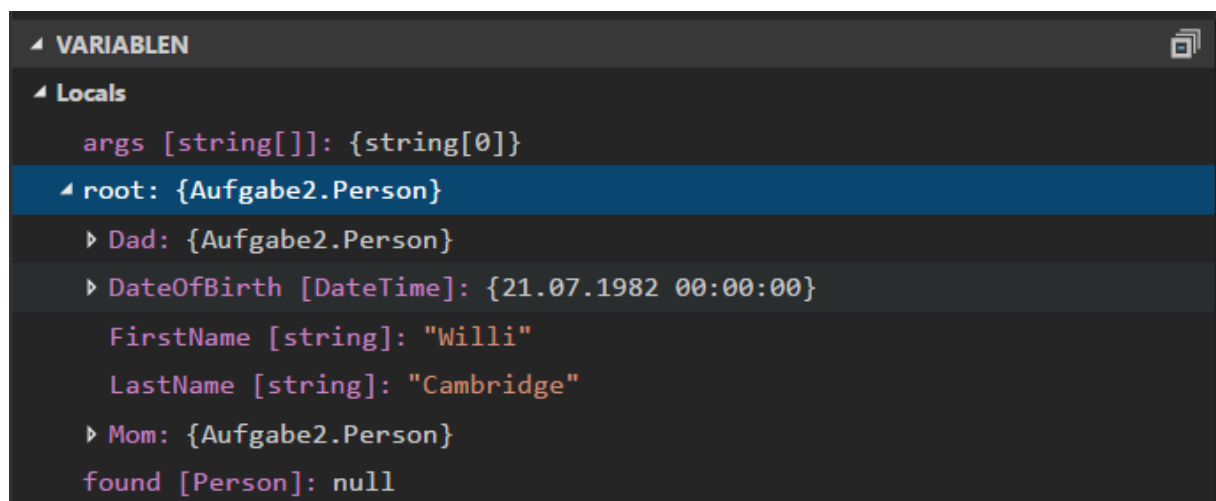
- Neuer Ordner (Aufgabe2) angelegt und in VisualStudioCode die Dateien Program.cs und FamilyTree.cs abgelegt
- Die Datenstruktur Person ist rekursiv, denn jedes Objekt vom Typ Person referenziert zwei weitere Objekte vom Typ Person, nämlich Mom und Dad:



- Breakpoint in Zeile 19 von Program.cs gesetzt und Debugger gestartet
- Der Debugger durchläuft das Programm und hält in Zeile 18 an, d.h. die Methode Build.Tree(); in FamilyTree.cs ist durchlaufen und wurde in Person root gespeichert.

```
Person root = Familytree.BuildTree();
```

- Im Debugger Fenster sieht man woraus die erste Person (Willi Cambrigde) besteht (Dad, DateOfBirth, FirstName, LastName, Mom)



- Dad und Mom bestehen wiederum aus DateOfBirth, FirstName, LastName und den Objekten Mom und Dad.
- Dies wird so lange fortgeführt bis die Objekte Mom und Dad auf null zeigen.

```
args [string[]]: {string[0]}
└─ root: {Aufgabe2.Person}
  └─ Dad: {Aufgabe2.Person}
    └─ Dad: {Aufgabe2.Person}
      └─ Dad: {Aufgabe2.Person}
        Dad [Person]: null ←
        ▶ DateOfBirth [DateTime]: {01.02.1882 00:00:00}
        FirstName [string]: "Andi"
        LastName [string]: "ElGreco"
        Mom [Person]: null ←
        ▶ DateOfBirth [DateTime]: {10.06.1921 00:00:00}
        FirstName [string]: "Philip"
        LastName [string]: "Battenberg"
        ▶ Mom: {Aufgabe2.Person}
        ▶ DateOfBirth [DateTime]: {14.11.1948 00:00:00}
```

- Die Methode Find() durchläuft rekursiv den Baum und prüft alle Person-Objekte darauf, ob die Bedingung in [Zeile 22](#) gegeben ist. Die erste Person, die die Bedingung erfüllt, wird zurückgeliefert

```
if (person.LastName != "Battenberg")
    return person;
```

→ In diesem Fall wird die Person „Cambridge“ zurückgegeben, weil er die erste Person im Baum ist und die Bedingung erfüllt.

Um nicht gleich die erste Person zurückzugeben, wird die Bedingung geändert:

Variante 1:

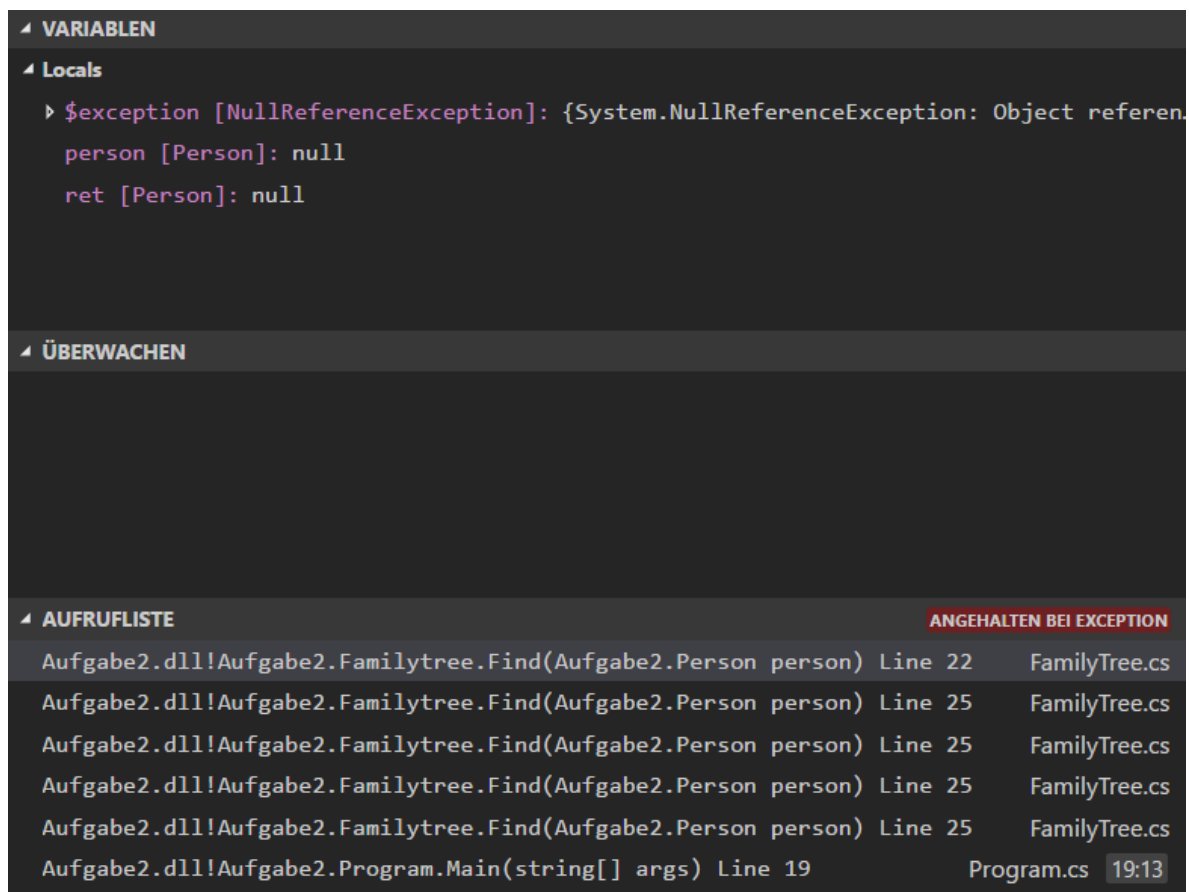
```
if (person.LastName != "Cambrigde")
    return person;
```

Der Code wirft keine Exception und es wird „Spencer“ zurückgegeben.

Variante 2:

```
if (person.LastName == "Battenberg")
    return person;
```

- Dieser Code wirft eine Exception:



Der Code `ret = Find(person.Dad);` wird nie ausgeführt und dadurch dass „Battenberg“ nur in Dad vorkommt, kann die Bedingung nie erfüllt werden.

Lösungsvorschlag: „Battenberg“ kommt nur in der väterlichen Seite von Willi, daher wird Find(person.Mom); mit Find(person.Dad) ausgetauscht:

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

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

    ret = Find(person.Mom);
    return ret;
}
```

Es wird keine Exception geworfen.