

## Softwaredesign, Abgabe 2, Übungsaufgabe

Program.cs und FamilyTree.cs wurden in Visual Studio Code angelegt.

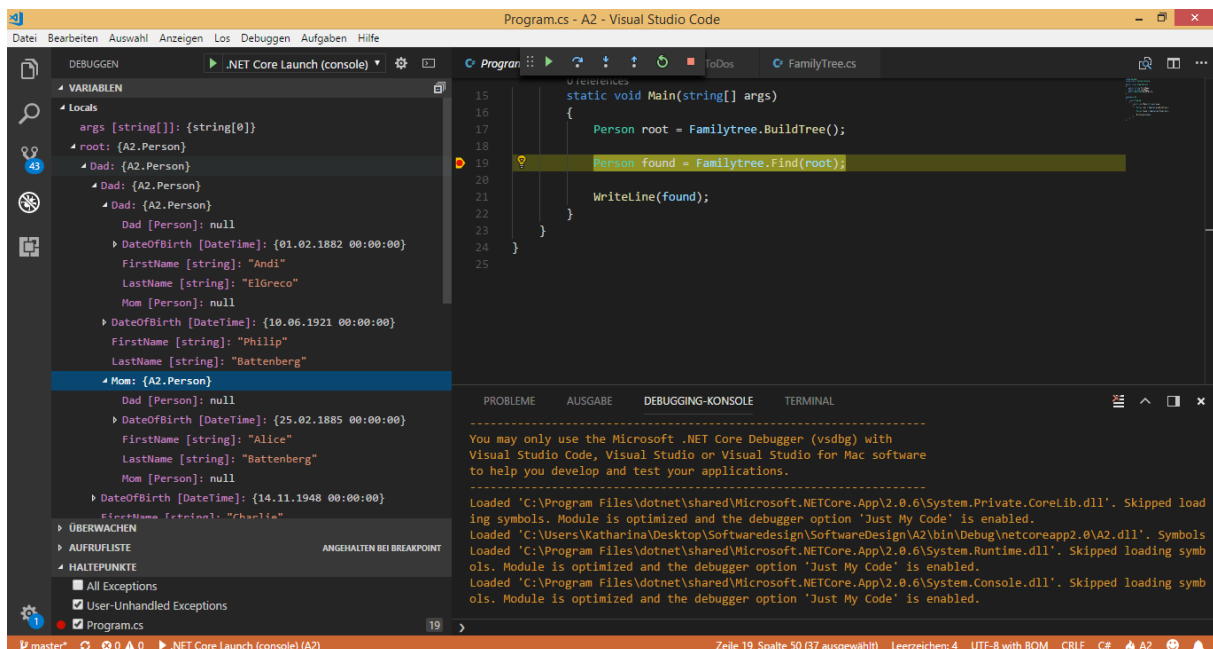
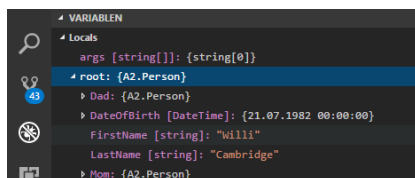
Die Datenstruktur Person ist rekursiv (Selbstaufbau), denn jedes Objekt vom Typ Person referenziert zwei weitere Objekte vom Typ Person (Mom und Dad):

```
public class Person
{
    public string FirstName;
    public string LastName;
    public DateTime DateOfBirth;

    public Person Mom;
    public Person Dad;
}
```

Das bedeutet, dass auch Mom und Dad so aufgebaut sind wie das vorherige Objekt vom Typ Person und auch zwei weitere Objekte vom Typ Person (Mom und Dad) referenzieren.

Breakpoint wurde gesetzt in Zeile 19 von Program.cs. Inhalt von root im Debugger: Aufgesetzter Beispielbaum vom Typ Person.



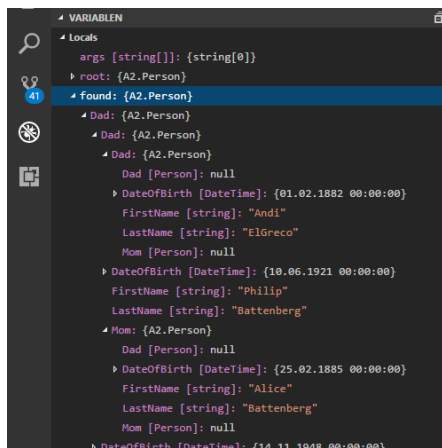
Breakpoint wurde wieder entfernt.

Die Methode Find() durchläuft rekursiv den Baum und prüft alle Person-Objekte auf die Bedingung in Zeile 22:

```
public static Person Find(Person person)
{
    Person ret = null;
    if (person.LastName != "Battenberg") //<- Bedingung Zeile 22
        return person;

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

Die erste Person, die die Bedingung erfüllt, wird zurückgeliefert: Willi Cambridge.



Änderung der Bedingung in Zeile 22, sodass nicht gleich die erste Person (Willi Cambridge) zurückgegeben wird:

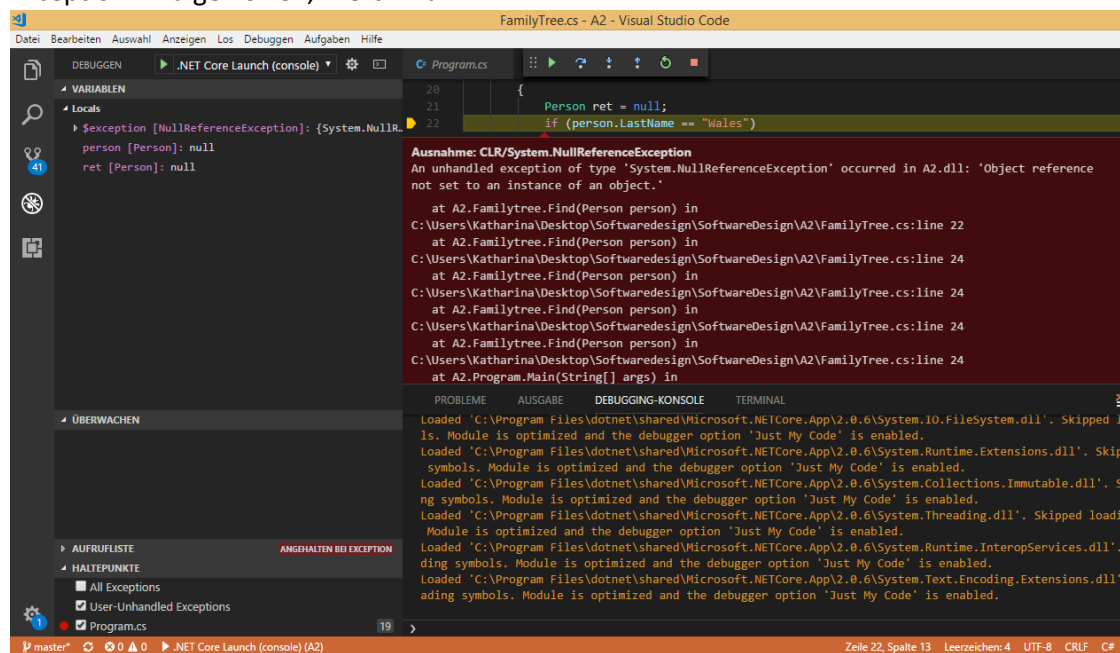
```
public static Person Find(Person person)
{
    Person ret = null;
    if (person.LastName != "Cambridge" & person.LastName != "Spencer")
        return person;
}
```

Ohne Abstürze, 3. Person wird zurückgegeben (Mom), es wird zuerst auf mütterlicher Seite geprüft.

(2.Änderung):

```
public static Person Find(Person person)
{
    Person ret = null;
    if (person.LastName == "Wales")
        return person;
    ret = Find(person.Mom);
    if (ret != null)
        return ret;
    ret = Find(person.Dad);           /*wird nicht ausgeführt*/
    return ret;
}
```

Exception wird geworfen, Wert = null:



Austausch von Find(person.Mom) und Find(person.Dad):

```
public static Person Find(Person person)
{
    Person ret = null;
    if (person.LastName == "Wales")
        return person;
    ret = Find(person.Dad);
    if (ret != null)
        return ret;
    ret = Find(person.Mom);           /*wird nicht ausgeführt*/
    return ret;
}
```

Keine Exception wurde geworfen.

Folgerung: Es wird eine NullReferenceException geworfen, weil die letzten Personen im Stammbaum keine Mom und keinen Dad haben. Eigenschaften haben demnach den Wert null.

Komplexere Bedingung siehe FamilyTree.cs.