

IPV4 Calculator

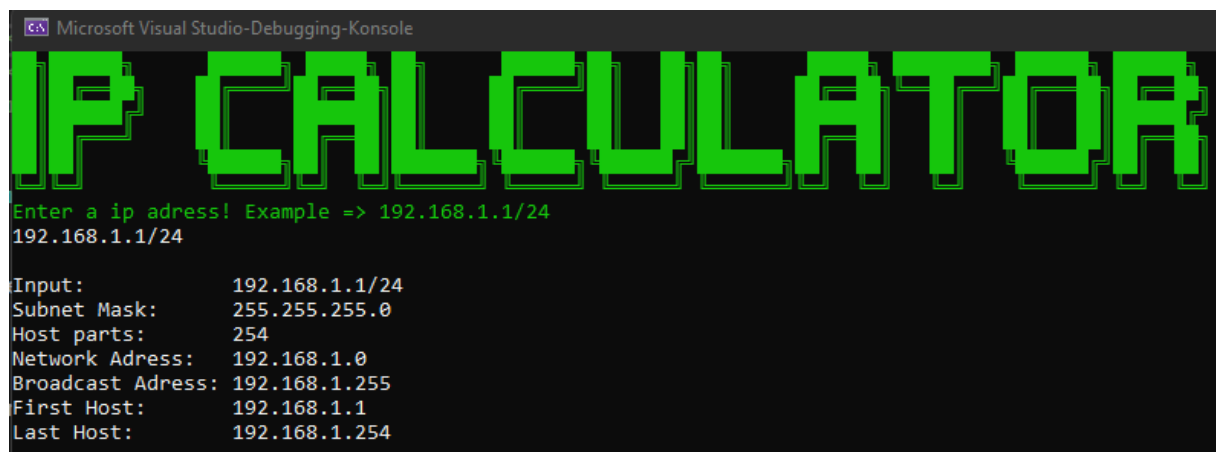
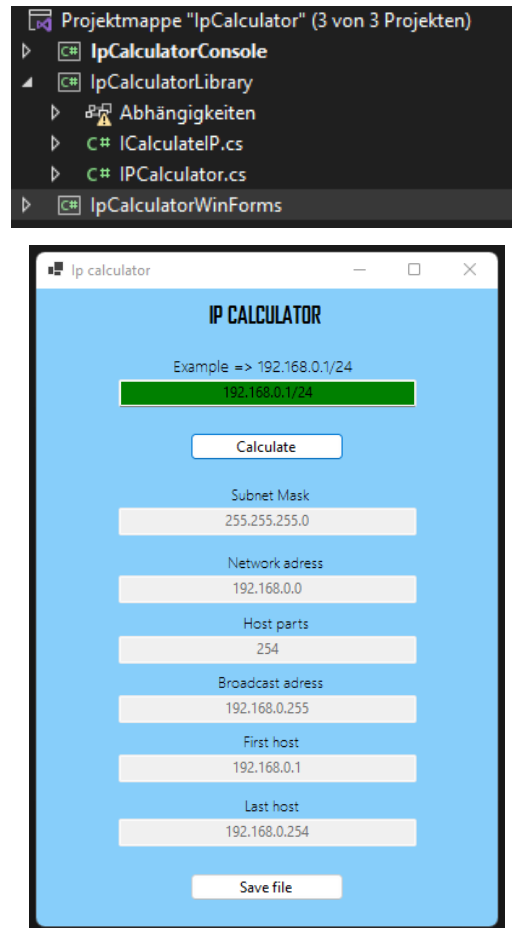
Eray Bayram

Inhalt

Einleitung.....	2
Ablauf	3
Software Code	3
Schwierigkeiten	4
Fehler.....	4
Persönliche Meinung.....	4

Einleitung

Die Logik der IP-Kalkulation habe in einer Library zusammengefasst, auf diese greife ich mit 2 verschiedenen UI's zu einmal mit Windows Forms und einmal mit der Konsole. Die Logik, die angewendet wurde, nennt sich Business Logik. Hierbei geht es darum, die UI mit der Logik zu trennen.



Ablauf

- Zuerst habe ich das Interface definiert.
- Dann habe ich angefangen Schritt für Schritt die Eigenschaften der IP-Adresse zu berechnen.
- Wenn eine Funktion fertig war, habe ich das Summary dazu geschrieben und versucht den Code zu optimieren.

Software Code

- Hier liste ich alle Funktionen der IpCalculator.cs Klasse.

```
#region Functions
/// <summary>
/// Splits the ip adress and subnet prefix with following seperators [".", "/"] and saves it to an array.
/// </summary>
2 Verweise
public void SplitIpAddress(string ipAndSubnet, ref int subnetSuffix, ref byte[] ipOctetsByte)[...]

/// <summary>
/// Calculates the hosts from a network range and returns it back.<br/>
/// networkID and broadcastID is subtracted <br/>
/// </summary>
/// <returns> number of hosts within a network range</returns>
2 Verweise
public int CalculateHosts(int subnetSuffix) => (int)Math.Pow(2, 32 - subnetSuffix) - 2;

/// <summary>
/// Converts a subnet suffix to a subnetmask
/// </summary>
/// <returns>string subnetmask</returns>
2 Verweise
public string GetSubnetMask(int subnetSuffix, ref string binarySubnetMask, ref byte[] subnetMaskByteArray)[...]

/// <summary>
/// Calculates the first network address <br/>
/// If you write 0 as add parameter, you will get the network address exmaple 192.168.1.0 <br/>
/// If you write 1 as add parameter, you will get the first usable hostadress from a network example 192.168.1.1
/// </summary>
/// <returns>string ip adress</returns>
3 Verweise
public string GetFirstNetworkAdress(int add, byte[] ipOctetsByte, byte[] subnetMaskByte)[...]

/// <summary>
/// Calculates the last network address <br/>
/// If you write 0 as subtract parameter, you will get the broadcast adress exmaple 192.168.1.255 <br/>
/// If you write 1 as subtract parameter, you will get the last usable hostadress from a network example 192.168.1.254
/// </summary>
/// <returns>string ip adress</returns>
3 Verweise
public string GetLastNetworkAdress(int subtract, byte[] ipOctetsByte, byte[] subnetMaskByte)[...]

/// <summary>
/// Checks if ip is valid and returns bool
/// </summary>
/// <returns>If ip is valid then return true else false</returns>
2 Verweise
public static bool CheckIfIpIsValid(string? userInput)[...]

/// <summary>
/// Creates a .txt file and write ip properties in file
/// </summary>
2 Verweise
public void CreateAndWriteFile(string path)[...]

#endregion
```

Schwierigkeiten

- Es gab immer wieder gewisse Punkte wo man sich anfangs ein bisschen unsicher war wie man das Problem angeht aber dies hat sich dann auch gleich wieder erledigt.
- Für mich persönlich war diese Aufgabe nicht wirklich schwer, der Bit oder und der Tilde Operator |~ war für mich was Neues, weshalb ich erst mal ein paar Sachen darüber lernen musste bis ich es implementieren konnte.

Fehler

- Anfangs dachte ich das die Broadcast Adresse immer .255 ist und der letzte Host immer .254. Dies war aber nicht so, dafür musste ich eine neue Lösung finden.
- Als ich mit dem Lehrer meinen Code besprochen habe, bemerkten wir das ich zu oft werte konvertiere anstatt diese in Variablen zwischenspeichern. Dies führt zu Performance Problemen bei großen Anwendungen. Dieses Problem habe ich danach behoben.

Persönliche Meinung

- Mir persönlich hat das Projekt Spaß gemacht, währenddessen konnte ich ein vertieftes Wissen über IP-Adressen sammeln
- Zudem habe ich meine Programmierkenntnisse erweitert.