Kierunki Rozwoju Telekomunikacji - projekt

Dokumentacja projektowa

Skrypt interpretujący adres IP

Projekt wykonali:

Mateusz Furgała

Adam Czerwiec

Paweł Kuźniar

Prowadzący: mgr inż. Natalia Cieplińska

1. Spis Treści

2. Opis ogólny

Skrypt jest narzędziem w Pythonie służącym do analizy i przetwarzania adresów IP w różnych formatach, w tym CIDR i maski dziesiętne. Program umożliwia:

- Obliczenie i wyświetlenie szczegółowych informacji o adresie IP, takich jak adres sieci, maska sieci, adres rozgłoszeniowy, liczba hostów w sieci, oraz reprezentacje binarne i szesnastkowe adresu IP.
- Sprawdzenie, czy dany adres IP należy do specjalnych zakresów (np. adresy prywatne, multicast, loopback, itp.).
- Sprawdzenie, czy adres IP należy do zadanej podsieci.

Skrypt obsługuje różne formaty wejściowe, umożliwiając użytkownikowi wygodne podanie adresu w formacie CIDR (np. 192.168.1.1/24) lub z maską dziesiętną (np. 192.168.1.1 255.255.255.0). Dodatkowo, wyświetla reprezentację binarną i szesnastkową adresu IP.

3. Wymagania

Skrypt wymaga Pythona w wersji 3.x oraz modułu ipaddress, który jest standardowym modułem w Pythonie do pracy z adresami IP.

4. Użyte moduły

- ipaddress – Wbudowany moduł w Pythonie do pracy z adresami IP. Używany do walidacji adresów IP, konwersji do różnych formatów oraz obliczania informacji o sieciach.
- argparse – Moduł do obsługi argumentów wiersza poleceń, pozwala na łatwe zarządzanie wejściami użytkownika.

5. Opis funkcji

a. validate_netmask(mask)

- Opis: Sprawdza poprawność maski sieci w formacie dziesiętnym.
- Argumenty: mask – Maska sieci w formacie dziesiętnym (np. 255.255.255.0).
- Zwraca: Wartość prefiksu sieci (np. 24).
- Podnosi wyjątek: ValueError w przypadku nieprawidłowej maski.

b. parse_ip_and_mask(ip, mask=None)

- Opis: Rozpoznaje, czy podano CIDR (np. 192.168.1.1/24), czy adres z maską dziesiętną (np. 192.168.1.1 255.255.255.0) oraz przetwarza je na obiekt ip_interface.
- Argumenty:
  - ip – Adres IP.
  - mask – Opcjonalna maska sieci.
- Zwraca: Obiekt ip_interface (np. IPv4Interface('192.168.1.1/24')).
- Podnosi wyjątek: ValueError w przypadku nieprawidłowego formatu adresu.

c. check_special_address(ip_obj)

- Opis: Sprawdza, czy adres IP należy do specjalnych kategorii adresów, takich jak: loopback, multicast, prywatny (RFC1918), link-local (APIPA), zarezerwowany, czy nieokreślony.
- Argumenty: ip_obj – Obiekt reprezentujący adres IP.
- Zwraca: String z nazwą kategorii, do której należy adres IP (np. "Prywatny (RFC1918)").

d. ip_info(ip_obj)

- Opis: Wyświetla szczegółowe informacje o adresie IP, w tym adres sieciowy, maskę sieci, adres rozgłoszeniowy, liczbę hostów w sieci oraz reprezentację binarną i szesnastkową.
- Argumenty: ip_obj – Obiekt reprezentujący adres IP.
- Zwraca: Brak, funkcja jedynie wyświetla informacje.

e. is_ip_in_subnet(ip, subnet)

- Opis: Sprawdza, czy adres IP należy do zadanej podsieci.
- Argumenty:
  - ip – Adres IP.
  - subnet – Podsiec w formacie CIDR (np. 192.168.1.0/24).
- Zwraca: True jeśli adres IP należy do podsieci, False w przeciwnym razie.

f. main()

- Opis: Główna funkcja, która przetwarza argumenty wejściowe, wywołuje odpowiednie funkcje oraz wyświetla wyniki.
- Argumenty: Brak, argumenty są przekazywane przez wiersz poleceń.
- Zwraca: Brak, funkcja jedynie wyświetla informacje.

6. Opis argumentów wejściowych

Skrypt obsługuje następujące argumenty wejściowe:

- ip (wymagane): Adres IP w jednym z następujących formatów:
  - Adres IP: 192.168.1.1 (Podejście klasowe – adres traktowany jako Klasy C).
  - CIDR: 192.168.1.1/24 (adres IP z CIDR).
  - Adres IP i maska dziesiętna: 192.168.1.1 255.255.255.0 (adres IP z maską).
  - CIDR z odstępem: 192.168.1.1 /24 (adres IP z prefiksem po odstępie).
- -n, --network (opcjonalnie): Określa adres podsieci, do której należy sprawdzić przynależność adresu IP (np. 192.168.1.0/24).
- -h, –help: wyświetla pomoc opisując użycie funkcji w sposób przedstawiony poniżej:

```
C:\Users\pkuzniar\PycharmProjects\IPInfo>python ip.py -h
usage: ip.py [-h] [-n ADRES [MASKA ...]] ip [ip ...]

Podręczne narzędzie sieciowe - sprawdzanie adresów IP i przynależności do sieci.
Użycie: <ADRES IP SPRAWDZANY> <ADRES DOCELOWY/CIDR> lub <ADRES DOCELOWY> <MASKA>

positional arguments:
  ip                    Adres IP w jednym z formatów:
                          • IP (Podejście klasowe):  192.168.1.1
                          • CIDR:                    192.168.1.1/24
                          • IP + maska dziesiętna:   192.168.1.1 255.255.255.0

options:
  -h, --help            show this help message and exit
  -n ADRES [MASKA ...], --network ADRES [MASKA ...]
                        Sprawdź przynależność podanego adresu IP do danej sieci:

                        Sieć docelową można określić za pomocą:
                          • CIDR:              192.168.1.0/24
                          • ADRES + MASKA:     192.168.1.0 255.255.255.0

                        Przykłady użycia:
                          python ip.py 192.168.1.100 -n 192.168.1.0/24
                          python ip.py 192.168.1.100 -n 192.168.1.0 255.255.255.0

                        Jeśli adres sieci jest nieprawidłowy, zgłoszony zostanie błąd.
```

7. Opis wyników

Skrypt wyświetla następujące informacje na temat adresu IP:

- Typ adresu: IPv4 lub IPv6.
- Adres IP: Wartość adresu IP.
- Adres sieci: Adres sieciowy wynikający z podanego adresu IP i maski.
- Maska sieci: Maska sieciowa.
- Adres rozgłoszeniowy: Adres broadcastowy dla sieci IPv4, dla IPv6 jest to "N/A".
- Liczba hostów w sieci: Liczba hostów dostępnych w danej sieci (dla IPv4 z wyłączeniem adresów sieciowego i rozgłoszeniowego).
- Reprezentacja binarna: Reprezentacja binarna adresu IP.
- Reprezentacja szesnastkowa: Reprezentacja szesnastkowa adresu IP.
- Kategoria adresu: Określenie, czy adres IP należy do specjalnej kategorii (loopback, multicast, prywatny, itp.).

Ponadto, skrypt wyświetla informację, czy dany adres IP należy do wskazanej podsieci (jeśli opcja --network została użyta).

8. Przykłady użycia

```
============================================================
Uruchamianie: C:\Python312\python.exe C:\Users\pkuzniar\PycharmProjects\IPInfo\ip.py 192.168.1.100
------------------------------------------------------------
============ Informacje o adresie 192.168.1.100 ============
Klasa adresu:               Klasa C
Adres prywatny:             Tak
Typ adresu:                 IPv4
Maska sieci:                255.255.255.0
Adres sieci:                192.168.1.0
Adres rozgłoszeniowy:       192.168.1.255
Liczba hostów w sieci:      254
Liczba adresów w sieci:     256
Reprezentacja binarna:      11000000101010000000000101100100
Reprezentacja szesnastkowa: 0xc0a80164
Kategoria adresu:           Prywatny (RFC1918)
============================================================
============================================================
```

```
============================================================
Uruchamianie: C:\Python312\python.exe C:\Users\pkuzniar\PycharmProjects\IPInfo\ip.py 169.254.0.1
------------------------------------------------------------
============ Informacje o adresie 169.254.0.1 ============
Klasa adresu:               Klasa B
Adres prywatny:             Tak
Typ adresu:                 IPv4
Maska sieci:                255.255.0.0
Adres sieci:                169.254.0.0
Adres rozgłoszeniowy:       169.254.255.255
Liczba hostów w sieci:      65534
Liczba adresów w sieci:     65536
Reprezentacja binarna:      10101001111111100000000000000001
Reprezentacja szesnastkowa: 0xa9fe0001
Kategoria adresu:           APIPA (Link-local)
============================================================
============================================================
```

```
============================================================
Uruchamianie: C:\Python312\python.exe C:\Users\pkuzniar\PycharmProjects\IPInfo\ip.py 100.230.8.73/17
------------------------------------------------------------
============ Informacje o adresie 100.230.8.73 ============
Klasa adresu:               Brak
Adres prywatny:             Nie
Typ adresu:                 IPv4
Maska sieci:                255.255.128.0
Adres sieci:                100.230.0.0
Adres rozgłoszeniowy:       100.230.127.255
Liczba hostów w sieci:      32766
Liczba adresów w sieci:     32768
Reprezentacja binarna:      01100100111001100000100001001001
Reprezentacja szesnastkowa: 0x64e60849
Kategoria adresu:           Brak specjalnej kategorii
============================================================
============================================================
```

```
============================================================
Uruchamianie: C:\Python312\python.exe C:\Users\pkuzniar\PycharmProjects\IPInfo\ip.py 74.125.200.100 255.255.252.0
------------------------------------------------------------
============ Informacje o adresie 74.125.200.100 ============
Klasa adresu:               Brak
Adres prywatny:             Nie
Typ adresu:                 IPv4
Maska sieci:                255.255.252.0
Adres sieci:                74.125.200.0
Adres rozgłoszeniowy:       74.125.203.255
Liczba hostów w sieci:      1022
Liczba adresów w sieci:     1024
Reprezentacja binarna:      01001010011111011100100001100100
Reprezentacja szesnastkowa: 0x4a7dc864
Kategoria adresu:           Brak specjalnej kategorii
============================================================
============================================================
```

```
============================================================
Uruchamianie: C:\Python312\python.exe C:\Users\pkuzniar\PycharmProjects\IPInfo\ip.py 2001:db8::1/64
------------------------------------------------------------
============ Informacje o adresie 2001:db8::1 ============
Klasa adresu:               Brak
Adres prywatny:             Tak
Typ adresu:                 IPv6
Maska sieci:                /64
Adres sieci:                N/A [IPv6]
Adres rozgłoszeniowy:       N/A [IPv6]
Liczba hostów w sieci:      18446744073709551616
Liczba adresów w sieci:     18446744073709551616
Reprezentacja binarna:      00100000000000010000110110111000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001
Reprezentacja szesnastkowa: 2001:0db8:0000:0000:0000:0000:0000:0001
Kategoria adresu:           Documentation (2001:db8::/32)
============================================================
============================================================
```

```
============================================================
Uruchamianie: C:\Python312\python.exe C:\Users\pkuzniar\PycharmProjects\IPInfo\ip.py ff02::1/128
------------------------------------------------------------
============ Informacje o adresie ff02::1 ============
Klasa adresu:               Brak
Adres prywatny:             Nie
Typ adresu:                 IPv6
Maska sieci:                /128
Adres sieci:                N/A [IPv6]
Adres rozgłoszeniowy:       N/A [IPv6]
Liczba hostów w sieci:      N/A
Liczba adresów w sieci:     1
Reprezentacja binarna:      11111111000000100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001
Reprezentacja szesnastkowa: ff02:0000:0000:0000:0000:0000:0000:0001
Kategoria adresu:           Multicast
============================================================
============================================================
```

```
============================================================
Uruchamianie: C:\Python312\python.exe C:\Users\pkuzniar\PycharmProjects\IPInfo\ip.py fe80::1/64
------------------------------------------------------------
============ Informacje o adresie fe80::1 ============
Klasa adresu:               Brak
Adres prywatny:             Tak
Typ adresu:                 IPv6
Maska sieci:                /64
Adres sieci:                N/A [IPv6]
Adres rozgłoszeniowy:       N/A [IPv6]
Liczba hostów w sieci:      18446744073709551616
Liczba adresów w sieci:     18446744073709551616
Reprezentacja binarna:      11111110100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001
Reprezentacja szesnastkowa: fe80:0000:0000:0000:0000:0000:0000:0001
Kategoria adresu:           APIPA (Link-local)
============================================================
============================================================
```

```
============================================================
Uruchamianie: C:\Python312\python.exe C:\Users\pkuzniar\PycharmProjects\IPInfo\ip.py 192.168.1.100 -n 192.168.1.0/24
------------------------------------------------------------
Adres IP 192.168.1.100 należy do sieci 192.168.1.0/24
============================================================


============================================================
Uruchamianie: C:\Python312\python.exe C:\Users\pkuzniar\PycharmProjects\IPInfo\ip.py 192.168.1.100 -n 10.0.0.0/8
------------------------------------------------------------
Adres IP 192.168.1.100 NIE należy do sieci 10.0.0.0/8
============================================================


============================================================
Uruchamianie: C:\Python312\python.exe C:\Users\pkuzniar\PycharmProjects\IPInfo\ip.py 192.168.1.100 -n 192.168.1.0 255.255.255.0
------------------------------------------------------------
Adres IP 192.168.1.100 należy do sieci 192.168.1.0 255.255.255.0
============================================================


============================================================
Uruchamianie: C:\Python312\python.exe C:\Users\pkuzniar\PycharmProjects\IPInfo\ip.py 192.168.1.100 -n 192.168.0.0 255.255.0.0
------------------------------------------------------------
Adres IP 192.168.1.100 należy do sieci 192.168.0.0 255.255.0.0
============================================================
```

Obsługa błędów:

```
============================================================
Uruchamianie: C:\Python312\python.exe C:\Users\pkuzniar\PycharmProjects\IPInfo\ip.py 300.300.300.300
------------------------------------------------------------
Błąd: Nieprawidłowy adres IP: 300.300.300.300
============================================================


============================================================
Uruchamianie: C:\Python312\python.exe C:\Users\pkuzniar\PycharmProjects\IPInfo\ip.py 192.168.1.100 -n 192.168.1.0 255.300.255.0
------------------------------------------------------------
Nieprawidłowa sieć: 192.168.1.0 255.300.255.0 (Nieprawidłowa sieć lub maska: 192.168.1.0 255.300.255.0)
============================================================


============================================================
Uruchamianie: C:\Python312\python.exe C:\Users\pkuzniar\PycharmProjects\IPInfo\ip.py 192.168.1.100 -n 100.100.100.300 255.255.265.255
------------------------------------------------------------
Nieprawidłowa sieć: 100.100.100.300 255.255.265.255 ('100.100.100.300 255.255.265.255' does not appear to be an IPv4 or IPv6 network)
============================================================
```

9. Wnioski

Skrypt jest elastycznym narzędziem, które umożliwia szybkie analizowanie adresów IP, sprawdzanie ich przynależności do specjalnych zakresów oraz wykonywanie obliczeń na temat podsieci. Dzięki zastosowaniu modułów ipaddress i argparse, program jest łatwy w użyciu i może być wykorzystany zarówno przez administratorów sieci, jak i programistów zajmujących się sieciami komputerowymi