

Programowanie C#

Zadanie 2

Piotr Błaszyński

26 lutego 2019

Zadanie: **Zaimplementować własny serwer www (http):**

- odpowiadający na porcie 8000 + numer stanowiska,
- obsługujący tylko żądania typu GET,
- jako parametry startowe przyjmujący:
 - numer portu,
 - ścieżkę do katalogu z którego serwujemy pliki.

Pierwsza część zadania: po wejściu przeglądarką na adres `http://localhost:8000` (8000+numer stanowiska), pokazuje się jedna z czterech (losowo wybrana) stron generowanych w testach z poprzedniego zadania (Po wykonaniu tej części zadania proszę o jej pokazanie). **Uwaga:** niektóre przeglądarki wymagają podania informacji nagłówkowej (wszystkie „porządne” serwery taką informację zwracają) i bez tej informacji nie ma gwarancji, że przeglądarka poprawnie się zachowa. Nagłówek jest oddzielony od treści strony pustą linią. Informacja ta może mieć następującą formę:

```
HTTP/1.1 200 OK
Expires: -1
Date: Wed, 18 Oct 2017 09:47:45 GMT
Access-Control-Allow-Origin: *
Content-Type: text/html; charset=utf-8
Server: meinheld/0.6.1
x-xss-protection: 1; mode=block
Connection: close
```

Druga część zadania: Przygotować prosty interfejs graficzny pozwalający na:

- podanie numeru portu serwera,

- podanie ścieżki do katalogu na dysku, z którego mają być serwowane pliki,
- uruchomienie serwera,
- zatrzymanie serwera (należy pamiętać o wykorzystaniu wątku roboczego).

Trzecia część zadania:

- przygotować kilka (ponad 3) plików (wczytywanych kodem wziętym z trzeciego testu z poprzedniego zadania)
- serwować pliki w zależności od nazwy podanej w adresie:
http://localhost:8000/plik1.html serwuje plik o nazwie plik1.html,
- jeżeli nie zostanie podana żadna nazwa pliku, serwer zwraca zawartość pliku index.html,
- jeśli nie ma pliku to wysłać stronę z kodem błędu 404
- jeśli w trakcie przetwarzania żądania wystąpi jakikolwiek wyjątek to wysłać stronę z kodem błędu 500.

Klasy potrzebne do realizacji zadania: **TcpListener** (System.Net, **Thread** (System.Threading), oczywiście Forms.

Dla „szybkich”: przygotować wersję pozwalającą na jednoczesne serwowanie wielu plików (wielowątkową), przygotować metodę testowania takiego sposobu serwowania z poziomu pojedynczego komputera.