

## Programmazione di Sistemi in Rete (A.A. 2017/18)

Esercitazione n. 8 del 7 novembre 2017

Prof. Eugenio Zimeo

### Esercizio 8.1

Si estenda l'implementazione del server Web, prevista dall'esercitazione 7, con l'elaborazione concorrente delle richieste HTTP.

### Esercizio 8.2

Si utilizzi il server Web per visualizzare su standard output il contenuto di un messaggio di richiesta con metodo POST. Si usi il file Form.html di esempio.

### Esercizio 8.3

Si modifichi l'implementazione del server Web in modo da: (1) usare lo stream di input associato ad un handler per costruire un oggetto di tipo `HttpRequestImpl` (classe che implementa l'interfaccia `HttpRequest` riportata di seguito), (2) usare un oggetto di tipo `HttpResponseImpl` (classe che implementa l'interfaccia `HttpResponse` riportata di seguito) per inviare sullo stream di output associato all'handler il contenuto del messaggio di risposta.

L'interfaccia `HttpRequest` è la seguente:

```
public interface HttpRequest {
    // Restituisce lo stream di input associato al messaggio
    public InputStream getInputStream() throws IOException;
    // Restituisce lo stream orientato ai caratteri associato al messaggio
    public BufferedReader getReader() throws IOException;
    // Restituisce il metodo usato per la richiesta
    public String getMethod();
    // Restituisce la stringa contenente i parametri di Query estratti da un form
    public String getQueryString();
    // Restituisce i formati MIME accettati
    public String getAccept();
    // Restituisce il tipo e la versione dello User Agent usato dal lato client
    public String getUserAgent();
    // Restituisce il valore del parametro specificato
    public String getParameter(String param);
}
```

L'interfaccia `HttpResponse` è la seguente:

```
public interface HttpResponse {
    // Restituisce lo stream associato al messaggio
    public OutputStream getOutputStream() throws IOException;
    // Restituisce il PrintWriter associato al messaggio
    public PrintWriter getWriter() throws IOException;
    // Specifica il tipo di contenuto del messaggio
    public void setContentType(String cType);
    // Specifica la lunghezza del body del messaggio
    public void setContentLength(int cLength);
    // Invia nello stream di output le intestazioni del messaggio
    public void flushHeader() throws IOException;
}
```