

ΔΗΜΙΟΥΡΓΙΑ ΚΑΙ ΕΠΕΞΕΡΓΑΣΙΑ ΦΟΡΜΩΝ HTML

Θέματα σε αυτό το Κεφάλαιο

- Υποβολή δεδομένων από φόρμες
- Χειριστήρια κειμένου
- Κουμπιά
- Πλαίσια ελέγχου και ραδιοπλήκτρα
- Σύνθετα πλαίσια και πλαίσια καταλόγου
- Χειριστήρια "ανεβάσματος" αρχείων
- Χάρτες εικόνων στο διακομιστή
- Κρυφά αρχεία
- Ομάδες χειριστηρίων
- Σειρά ενεργοποίησης
- Διακομιστής Ιστού για την αποσφαλμάτωση φορμών

Κεφάλαιο

19

Οι φόρμες HTML παρέχουν μια απλή και αξιόπιστη διασύνδεση χρήστη για τη συλλογή δεδομένων από το χρήστη και τη μετάδοση δεδομένων σε μια μικρουπηρεσία ή σε κάποιο άλλο πρόγραμμα του διακομιστή με στόχο την περαιτέρω επεξεργασία. Σε αυτό το κεφάλαιο θα παρουσιάσουμε τα τυπικά χειριστήρια φόρμες που ορίζονται από τις προδιαγραφές της HTML 4.0. Πριν περιγράψουμε όμως τα χειριστήρια, θα εξηγήσουμε πρώτα πώς μεταδίδονται τα δεδομένα της φόρμας σε ένα διακομιστή όταν πραγματοποιείται μια αίτηση GET ή POST.

Θα παρουσιάσουμε επίσης ένα μίνι διακομιστή Ιστού που είναι χρήσιμος για την κατανόηση και την αποσφαλμάτωση των δεδομένων που στέλλονται μέσα από φόρμες HTML. Ο διακομιστής διαβάζει απλώς όλα τα δεδομένα HTML που του έχουν σταλεί από το φυλλομετρητή και επιστρέφει μια ιστοσελίδα όπου ενσωματώνονται όλες αυτές οι γραμμές σε ένα στοιχείο PRE. Χρησιμοποιούμε αυτόν το διακομιστή σε όλα τα παραδείγματα αυτού του κεφαλαίου για να δείξουμε τα δεδομένα που στέλλονται από χειριστήρια φόρμας στο διακομιστή όταν γίνεται η υποβολή της φόρμας HTML.

Για να χρησιμοποιήσετε φόρμες, θα πρέπει να θυμάστε πού τοποθετούνται τα κανονικά αρχεία HTML έτσι ώστε να είναι προσπελάσιμα στο διακομιστή Ιστού. Αυτή η θέση διαφέρει από διακομιστή σε διακομιστή, όπως περιγράφουμε στο Κεφάλαιο 2 και στο Παράρτημα. Στον επόμενο κατάλογο υπενθυμίζουμε τη θέση των αρχείων HTML για την προεπιλεγμένη εφαρμογή Ιστού στους διακομιστές Tomcat, JRun, και Resin.

Προεπιλεγμένη εφαρμογή Ιστού: Tomcat

- Κύρια θέση
κατάλογος εγκατάστασης webapps/ROOT
- Αντίστοιχη διεύθυνση URL
<http://υπολογιστής/ημερησία/ΚάποιαΑρχείο.html>
- Πιο συγκεκριμένη θέση
κατάλογος εγκατάστασης webapps/ROOT/ΚάποιαΚατάλογος
- Αντίστοιχη διεύθυνση URL
<http://υπολογιστής/ημερησία/ΚάποιαΚατάλογος/ΚάποιαΑρχείο.html>

Προεπιλεγμένη εφαρμογή Ιστού: JRun

- Κύρια θέση
κατάλογος εγκατάστασης servers/default/default-war
- Αντίστοιχη διεύθυνση URL
<http://υπολογιστής/ημερησία/ΚάποιαΑρχείο.html>
- Πιο συγκεκριμένη θέση
κατάλογος εγκατάστασης servers/default/default-war/ΚάποιαΚατάλογος
- Αντίστοιχη διεύθυνση URL
<http://υπολογιστής/ημερησία/ΚάποιαΚατάλογος/ΚάποιαΑρχείο.html>

Προεπιλεγμένη εφαρμογή Ιστού: Resin

- Κύρια θέση
κατάλογος εγκατάστασης doc
- Αντίστοιχη διεύθυνση URL
<http://υπολογιστής/ημερησία/ΚάποιαΑρχείο.html>
- Πιο συγκεκριμένη θέση
κατάλογος εγκατάστασης doc/ΚάποιαΚατάλογος
- Αντίστοιχη διεύθυνση URL
<http://υπολογιστής/ημερησία/ΚάποιαΚατάλογος/ΚάποιαΑρχείο.html>

Η προεπιλεγμένη εφαρμογή Ιστού του διακομιστή είναι χρήσιμη για εξέταση και εκμάθησή, αλλά όταν εκτυπώνετε πραγματικές εφαρμογές είναι σχεδόν σίγουρο ότι θα χρησιμοποιείτε προσαρμοσμένες εφαρμογές Ιστού για λεπτομέρειες διαβάστε ξανά την Ενότητα 2.11.

19.1 Πώς μεταδίδουν δεδομένα οι φόρμες HTML

Οι φόρμες HTML σας επιτρέπουν να δημιουργήσετε διάφορα χειριστήρια διασύνδεσης χρήστη για τη συλλογή της εισόδου σε μια ιστοσελίδα. Καθένα από τα χειριστήρια αυτά έχει συνήθως

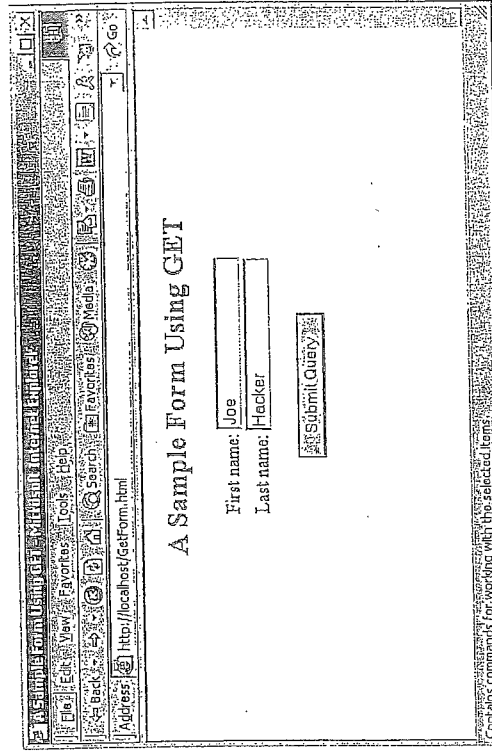
19.1 Πώς μεταδίδουν δεδομένα οι φόρμες HTML

ένα όνομα και μια τιμή, όπου το όνομα καθορίζεται στον κώδικα HTML και η τιμή προέρχεται είτε από είσοδο του χρήστη, είτε από μια προεπιλεγμένη τιμή που καθορίζεται στον κώδικα HTML. Ολόκληρη η φόρμα συσχετίζεται με το URL του προγράμματος που θα επεξεργαστεί τα δεδομένα, και όταν ο χρήστης υποβάλει τη φόρμα (συνήθως με το πάτημα ενός κουμπιού) τα ονόματα και οι τιμές των χειριστηρίων αποστέλλονται στην καθορισμένη διεύθυνση URL ως αλφαριθμητικό της μορφής

όνομα1=τιμή1&όνομα2=τιμή2...&όνομαN=τιμήN

Αυτό το αλφαριθμητικό μπορεί να σταλεί στο καθορισμένο πρόγραμμα με δύο μεθόδους: τις μεθόδους GET ή POST. Η πρώτη μέθοδος, η αίτηση HTTP GET, προσαρτά τα δεδομένα της φόρμας στο τέλος του καθορισμένου URL μετά από ένα ερωτηματικό. Η δεύτερη μέθοδος, η αίτηση HTTP POST, στέλνει τα δεδομένα μετά τις κεφαλίδες αίτησης HTTP και μια κενή γραμμή. Στα παραδείγματα που ακολουθούν δείχνουμε αναλυτικά τον τρόπο με τον οποίο στέλλονται τα δεδομένα στο διακομιστή, τόσο στην περίπτωση της αίτησης GET όσο και στην περίπτωση της αίτησης POST.

Για παράδειγμα, η Λίστα 19.1 (κώδικας HTML) και η Εικόνα 19-1 (τυπικό αποτέλεσμα) παρουσιάζουν μια απλή φόρμα με δύο πεδία κειμένου (text fields). Τα στοιχεία HTML που αποτελούν τη φόρμα εξετάζονται με λεπτομέρειες στη συνέχεια αυτού του κεφαλαίου, όμως προς το παρόν σημειώστε μερικά πράγματα. Πρώτον, παρατηρήστε ότι το ένα πεδίο κειμένου έχει το



Εικόνα 19-1 Αρχικό αποτέλεσμα της σελίδας GetForm.html.

όνομα firstName και το άλλο έχει το όνομα lastName. Δεύτερον, σημειώστε ότι τα χειριστήρια GUI θεωρούνται στοιχεία επιπέδου κειμένου (εμβόλιμα), οπότε πρέπει να χρησιμοποιήσετε ρητή μορφοποίηση HTML για να εξασφαλίσετε ότι τα χειριστήρια θα εμφανίζονται δίπλα από το κείμενο που τα περιγράφει. Τέλος, παρατηρήστε ότι το στοιχείο FORM ορίζει τη διεύθυνση `http://localhost:8088/SomeProgram` ως τη διεύθυνση URL στην οποία θα σταλούν τα δεδομένα.

Πριν από την υποβολή της φόρμας, ξεκίνησαμε ένα πρόγραμμα διακομιστή με το όνομα EchoServer στη θύρα 8088 του τοπικού μας μηχανήματος. Το πρόγραμμα EchoServer, που παρουσιάζεται στην Ενότητα 19.12, είναι ένας μίνι διακομιστής Ιστού που χρησιμοποιείται για αποσφαλμάτωση. Ανεξάρτητα από το URL που καθορίζεται και από τα δεδομένα που στέλνονται στο διακομιστή EchoServer, αυτός επιστρέφει μια ιστοσελίδα που δείχνει όλες τις πληροφορίες τις οποίες έλαβε από το φυλλομετρητή. Όπως φαίνεται στην Εικόνα 19-2, όταν η φόρμα υποβληθεί με την τιμή Joe στο πρώτο πεδίο κειμένου και Hacker στο δεύτερο πεδίο κειμένου, ο φυλλομετρητής απλώς ζητά τη διεύθυνση URL `http://localhost:8088/SomeProgram?firstName=Joe&lastName=Hacker`.

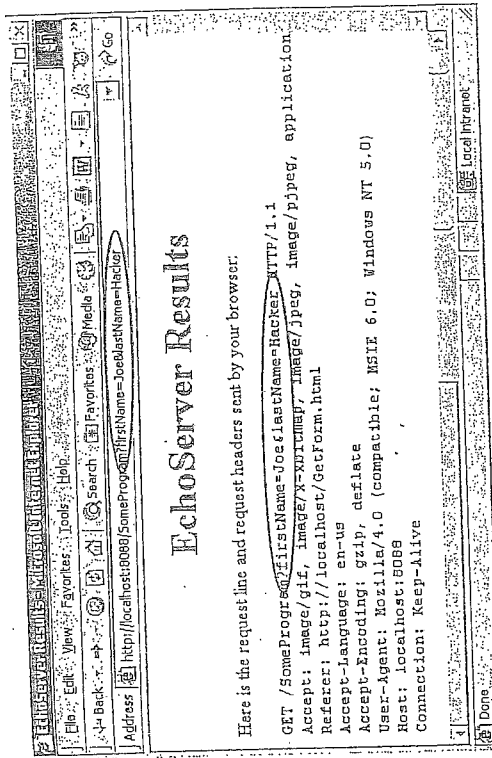
Λίστα 19.1 GetForm.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
  <TITLE>A Sample Form Using GET</TITLE>
</HEAD>
<BODY BGCOLOR="#FDF5E6">
  <CENTER>
<H2>A Sample Form Using GET</H2>
<FORM ACTION="http://localhost:8088/SomeProgram">
  First name:
  <INPUT TYPE="TEXT" NAME="firstName" VALUE="Joe"><BR>
  Last name:
  <INPUT TYPE="TEXT" NAME="lastName" VALUE="Hacker"><P>
  <INPUT TYPE="SUBMIT"> <!-- Press this button to submit form -->
</FORM>
</CENTER>
</BODY></HTML>
```

Η Λίστα 19.2 (κώδικας HTML) και η Εικόνα 19-3 (τυπικό αποτέλεσμα) παρουσιάζουν μία παραλλαγή που χρησιμοποιεί τη μέθοδο POST, αντί της μεθόδου GET. Όπως φαίνεται στην Εικόνα 19-4, η υποβολή της φόρμας με τις τιμές Joe και Hacker στα πεδία κειμένου δημιουργεί τη γραμμή `firstName=Joe&lastName=Hacker`, η οποία αποστέλλεται στο φυλλομετρητή σε ξεχωριστή γραμμή μετά τις κεφαλίδες αίτησης HTTP και μία κενή γραμμή.

Αυτή είναι και η γενική ιδέα πίσω από τις φόρμες HTML: χειριστήρια GUI συλλέγουν τα δεδομένα από το χρήστη, κάθε χειριστήριο έχει ένα όνομα και μια τιμή, και όταν υποβάλλεται η φόρμα στέλνεται στο διακομιστή ένα αλφαριθμητικό που περιέχει όλα τα ξενόγρια ονομάτων

19.1 Πώς μεταδίδουν δεδομένα οι φόρμες HTML



Εικόνα 19-2 Αίτηση HTTP που στάλθηκε από τον Internet Explorer 6.0 κατά την υποβολή της σελίδας GetForm.html.

τιμών. Η εξαγωγή των ονομάτων και των τιμών στο διακομιστή γίνεται εύκολα με τις μικροϋπηρεσίες: το θέμα αυτό περιγράφεται στο Κεφάλαιο 4 (Χειρισμός αιτήσεων πελάτη: Δεδομένα φόρμας). Στις ενότητες που ακολουθούν περιγράφουμε τα χειριστήρια φόρμας που χρησιμοποιούνται συχνότερα.

Λίστα 19.2 PostForm.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
  <TITLE>A Sample Form Using POST</TITLE>
</HEAD>
<BODY BGCOLOR="#FDF5E6">
  <CENTER>
<H2>A Sample Form Using POST</H2>
<FORM ACTION="http://localhost:8088/SomeProgram"
  METHOD="POST">
  First name:
  <INPUT TYPE="TEXT" NAME="firstName" VALUE="Joe"><BR>
  Last name:
  <INPUT TYPE="TEXT" NAME="lastName" VALUE="Hacker"><P>
  <INPUT TYPE="SUBMIT">
</FORM>
</CENTER>
</BODY></HTML>
```

A Sample Form Using POST

First name:

Last name:

Εικόνα 19-3 Αρχικό αποτέλεσμα της σελίδας PostForm.html.

EchoServer Results

Here is the request line and request headers sent by your browser.

```
POST /SomeProgram HTTP/1.1
Accept: image/gif, image/jpeg, image/jpg, image/png, application/vnd.ms-powerpoint
Accept-Charset: utf-8
Accept-Encoding: gzip, deflate
Accept-Language: en-us
Cache-Control: no-cache
Connection: Keep-Alive
Content-Length: 29
Host: localhost:8080
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)
X-Host-Name: Joe & Last Name: Hacker
```

Εικόνα 19-4 Αίτηση HTTP που στάλθηκε από τον Internet Explorer 6.0 κατά την υποβολή της σελίδας PostForm.html.

19.2 Το στοιχείο FORM

Οι φόρμες HTML σάς επιτρέπουν να δημιουργείτε ένα σύνολο στοιχείων εισόδου δεδομένων που σχετίζονται με ένα συγκεκριμένο URL. Σε καθένα από αυτά τα στοιχεία αποδίδεται ένα όνομα μέσα από τον πηγάτο κώδικα HTML, και καθένα από αυτά τα στοιχεία αποικτά μια τιμή είτε με βάση τον αρχικό κώδικα HTML, ή με βάση την είσοδο του χρήστη. Όταν υποβάλλεται η φόρμα, τα ονόματα και οι τιμές όλων των ενεργών στοιχείων συλλέγονται σε ένα αλφαριθμητικό με ένα σύμβολο = μεταξύ του ονόματος και της τιμής και ένα σύμβολο & ανάμεσα σε κάθε ξεχωριστό ονόματος/τιμής. Κατόπιν αυτό το αλφαριθμητικό μεταδίδεται στη διεύθυνση URL που καθορίζεται από το στοιχείο FORM. Το αλφαριθμητικό είτε προσαρτάται στο URL μετά από ένα ερωτηματικό, είτε αποστέλλεται σε ξεχωριστή γραμμή μετά από τις κεφαλίδες αίτησης HTTP και μια κενή γραμμή — ανάλογα με το αν χρησιμοποιήθηκε ως μέθοδος υποβολής η μέθοδος GET ή η μέθοδος POST. Οι ενότητες που ακολουθούν περιγράφουν τα διάφορα χαρακτηριστήρια διασύνδεσης χρήστη που μπορούν να χρησιμοποιηθούν σε φόρμες.

Στοιχείο HTML: <FORM ACTION="..." ...> ... </FORM>
 Ιδιότητες: ACTION, METHOD, ENCTYPE, TARGET, ONSUBMIT, ONRESET, ACCEPT, ACCEPT-CHARSET

Το στοιχείο FORM δημιουργεί μια περιοχή για στοιχεία εισόδου δεδομένων και καθορίζει τη διεύθυνση URL στην οποία θα μεταδοθούν τα δεδομένα τα οποία θα συλλεγούν. Για παράδειγμα:

```
<FORM ACTION="http://some.isp.com/someWebApp/SomeServlet">
  Στοιχεία εισόδου FORM και κανονική HTML
</FORM>
```

Το υπόλοιπο τμήμα αυτής της ενότητας επεξηγεί τις ιδιότητες που ισχύουν για το στοιχείο FORM: ACTION, METHOD, ENCTYPE, TARGET, ONSUBMIT, ONRESET, ACCEPT, και ACCEPT-CHARSET. Σημειώστε ότι δεν εξετάζουμε ιδιότητες όπως οι STYLE, CLASS, και LANG, που ισχύουν για γενικά στοιχεία HTML, αλλά μόνο τις ιδιότητες που αφορούν ειδικά το στοιχείο FORM.

ACTION

Η ιδιότητα ACTION καθορίζει τη διεύθυνση URL για το πρόγραμμα του διακομιστή που θα επεξεργαστεί τα δεδομένα FORM (για παράδειγμα, <http://www.whitehouse.gov/servlet/schedule-fund-raiser>). Αν το πρόγραμμα διακομιστή βρίσκεται στον ίδιο διακομιστή από τον οποίο προέρχεται και η φόρμα HTML, συνιστούμε τη χρήση σχετικού URL, αντί για απόλυτο URL. Αυτή η προσέγγιση σας επιτρέπει να μετακινήσετε τη φόρμα και τη μικροϋπηρεσία σε διαφορετικό υπολογιστή υπηρεσίας χωρίς να χρειάζεται να τις τροποποιήσετε. Αυτό είναι συνήθως σημαντικό ζήτημα, επειδή κατά κανόνα η ανάπτυξη και ο έλεγχος γίνονται σε ένα μηχάνημα και η εκκίνηση σε κάποιο άλλο μηχάνημα. Για παράδειγμα,

```
ACTION="/servlet/schedule-fund-raiser"
```



Η προσέγγιση του βιβλίου

Αν η μικρούτηρά σας ή η σελίδα JSP βρίσκεται στον ίδιο διακομιστή με τη φόρμα HTML, χρησιμοποιήστε σχετική διεύθυνση URL στην ιδιότητα ACTION.

Επιπρόσθετα, μπορείτε να καθορίσετε μια διεύθυνση ηλεκτρονικού ταχυδρομείου στην οποία θα αποσταλούν τα δεδομένα FORM (π.χ., mailto:audit@its.gov). Ορισμένοι φορείς παροχής υπηρεσιών Internet (ISP) δεν επιτρέπουν σε κανονικούς χρήστες τη δημιουργία προγραμμάτων στο διακομιστή, ή χρεώνουν επιπλέον για αυτό το προνόμιο. Σε μια τέτοια περίπτωση η αποστολή δεδομένων μέσω ηλεκτρονικού ταχυδρομείου αποτελεί βολική επιλογή όταν δημιουργείτε σελίδες που πρέπει να συλλέξουν δεδομένα αλλά δεν χρειάζεται να επιστρέψουν αποτελέσματα (π.χ., για την αποδοχή παραγγελιών για προϊόντα). Όταν χρησιμοποιείτε μια διεύθυνση URL mailto θα πρέπει να χρησιμοποιήσετε τη μέθοδο POST (διαβάστε σχετικά για την ιδιότητα METHOD στην υποενότητα που ακολουθεί).

Θα πρέπει επίσης να σημειώσουμε ότι η ιδιότητα ACTION δεν είναι απαραίτητη για το στοιχείο FORM. Αν παραλείψετε την ιδιότητα ACTION, τα δεδομένα της φόρμας θα σταλούν στο ίδιο URL με αυτό της φόρμας. Για ένα παράδειγμα: χρήσης της προσέγγισης αυτο-υποβολής φόρμας δείτε την Ενότητα 4.8.

METHOD

Η ιδιότητα METHOD καθορίζει πώς θα μεταδοθούν τα δεδομένα στο διακομιστή HTTP. Όταν χρησιμοποιείτε τη μέθοδο GET, τα δεδομένα προσαρτώνται στο τέλος της καθορισμένης διεύθυνσης URL, μετά από ένα ερωτηματικό. Για ένα σχετικά παράδειγμα δείτε την Ενότητα 19.1 (Πώς μεταδίδουν δεδομένα οι φόρμες HTML). Η προσεγγισμένη μέθοδος είναι η GET, που είναι επίσης η μέθοδος η οποία χρησιμοποιείται όταν ο χρήστης πληκτρολογεί κάποιο URL στη γραμμή διεύθυνσης ή πατά σε κάποιο σύνδεσμο υπερ-κειμένου. Όταν χρησιμοποιείτε τη μέθοδο POST, τα δεδομένα στέλνονται σε ξεχωριστή γραμμή. Ανάλογα με την περίπτωση, μπορεί να είναι προτιμότερη είτε η μέθοδος GET, είτε η μέθοδος POST.

Επειδή με τη μέθοδο GET τα δεδομένα αποτελούν τμήμα του URL, το πλεονέκτημα της GET είναι ότι μπορείτε να κάνετε τα παρακάτω:

- Αποθήκευση των αποτελεσμάτων της υποβολής μιας φόρμας. Για παράδειγμα, μπορείτε να υποβάλετε τα δεδομένα και να αποθηκεύσετε το URL που θα προκύψει ως σελίδα, να το σταλείτε σε κάποιο συνάδελφο μέσω ηλεκτρονικού ταχυδρομείου, ή να το τοποθετήσετε σε έναν κανονικό σύνδεσμο υπερ-κειμένου. Η δυνατότητα δημιουργίας σελιδοδείκτη με τη σελίδα που προκύπτει είναι ο κύριος λόγος για τον οποίο οι τοποθεσίες google.com, yahoo.com, και άλλες μηχανές αναζήτησης χρησιμοποιούν τη μέθοδο GET.
- Πληκτρολόγηση δεδομένων με το χέρι. Μπορείτε να ελέγξετε τις μικρουπηρεσίες ή τις σελίδες JSP που χρησιμοποιούν τη μέθοδο GET πληκτρολογώντας απλώς μια διεύ-

θυνα URL με προσωπτημένα τα κατάλληλα δεδομένα. Αυτή η δυνατότητα είναι ιδιαίτερα χρήσιμη κατά την αρχική ανάπτυξη.

Επειδή τα δεδομένα POST δεν αποτελούν τμήμα του URL, τα πλεονεκτήματα της μεθόδου POST είναι ότι μπορείτε να κάνετε τα παρακάτω:

- Μετάδοση μεγάλου όγκου δεδομένων. Πολλοί φυλλομετρητές περιορίζουν τα URL σε μερικές χιλιάδες χαρακτήρες, οπότε η GET είναι ακατάλληλη όταν η φόρμα πρέπει να στείλει μεγάλο όγκο δεδομένων. Επειδή οι φόρμες HTML σάς επιτρέπουν να ανεβάσετε (upload) αρχεία από το μηχάνημα του πελάτη (δείτε την Ενότητα 19.7), η αποστολή πολλών megabyte δεδομένων είναι αρκετά συνηθισμένη υπόθεση. Μόνο η μέθοδος POST μπορεί να πραγματοποιήσει την εργασία αυτή.
- Αποστολή δυσδικών δεδομένων. Τα διαστήματα, οι επαναφορές κεφαλής, οι σπληνικές θέτες, και πολλοί άλλοι χαρακτήρες δεν είναι έγκυροι χαρακτήρες στις διευθύνσεις URL. Αν ανεβάσετε ένα μεγάλο δυαδικό αρχείο στο διακομιστή, θα είναι χρονοβόρα η διαδικασία της κωδικοποίησης όλων των χαρακτήρων πριν από τη μετάδοση και της αποκωδικοποίησής τους στην άλλη πλευρά.
- Διατήρηση του ιδιαιδικού χαρακτήρα των δεδομένων από κάποιον που παρακολουθεί το χρήστη. Οι φόρμες HTML σάς επιτρέπουν να δημιουργήσετε πεδία κωδικών πρόσβασης, όπου τα δεδομένα αντικαθίστανται από αστέρισκους στην οθόνη. Όμως η χρήση ενός πεδίου κωδικού πρόσβασης είναι άσκοπη όταν τα δεδομένα εμφανίζονται ως κείμενο στη διεύθυνση URL — κάτι τέτοιο θα επέτρεπε στους "αδιάκριτους" να κοιτούν πίσω από την πλάτη του χρήστη ή να ψάχνουν τη λίστα του ιστορικού όταν ο χρήστης δεν είναι μισοστός στον υπολογιστή. Πρέπει να σημειώσουμε, όμως, ότι η απλή χρήση της μεθόδου POST δεν παρέχει προστασία απέναντι σε κάποιον που χρησιμοποιεί κάποιο "πρόγραμμα λαχνοκούπακτών" (packet sniffer) στη δικτυακή σύνδεση. Για προστασία απέναντι σε αυτόν τον τύπο επίθεσης, χρησιμοποιήστε ασφαλείς υποδοχές SSL (σύνδεση https://) για την κρυπτογράφηση της κυκλοφορίας στο δίκτυο. Για περισσότερες πληροφορίες σχετικά με τη χρήση υποδοχών SSL σε εφαρμογές Ιστού δείτε τα κεφάλαια σχετικά με την ασφάλεια εφαρμογών Ιστού στο βιβλίο More Servlets and JavaServer Pages. Για την ανάγνωση δεδομένων GET ή POST από μια μικρούτηρά σας, καλείτε τη μέθοδο

```
request.getParameter("name")
```

όπου name είναι η τιμή της ιδιότητας NAME για το στοιχείο εισόδου της φόρμας HTML. Για περισσότερες λεπτομέρειες δείτε το Κεφάλαιο 4 (Χειρισμός αιτήσεων πελάτη: Δεδομένα φόρμας). Σημειώνουμε όμως ότι, αν είναι απαραίτητο, μπορείτε να χρησιμοποιήσετε τη μέθοδο request.getInputStream για να διαβάσετε άμεσα τα δεδομένα POST, όπως φαίνεται εδώ

```
int length = request.getContentLength();
if (length > SOME_MAXSIZE) {
    throw new IOException("Possible denial of service attack");
}
byte[] data = new byte[length];
```

```
ServletInputStream inputStream = request.getInputStream();
int read = inputStream.readLine(data, 0, length);
```

ENCTYPE

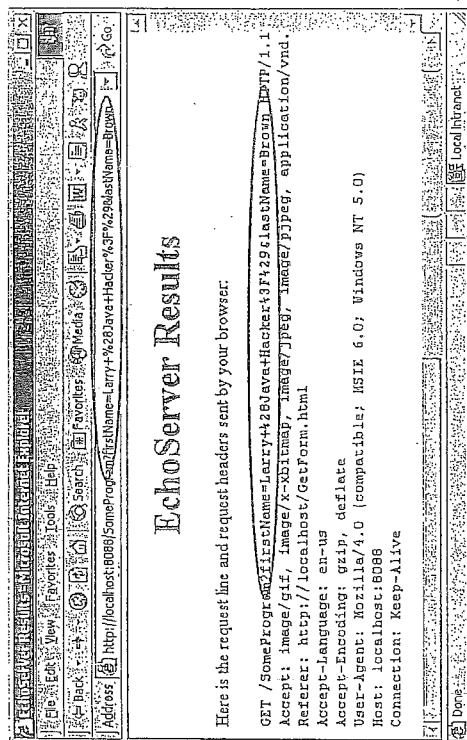
Αυτή η ιδιότητα καθορίζει τον τρόπο με τον οποίο θα κωδικοποιηθούν τα δεδομένα πριν μεταδοθούν. Η προεπιλεγμένη τιμή είναι η `application/x-www-form-urlencoded`. Η κωδικοποίηση, όπως καθορίστηκε από την Κοινότητα του Παγκόσμιου Ιστού (World Wide Web Consortium), είναι η UTF-8 με τη διαφορά ότι ο πελάτης μετατρέπει τα διαστήματα σε σύμβολα συν (+) και όλους τους άλλους μη αλφαριθμητικούς χαρακτήρες σε σύμβολο ποσοστού (%) ακολουθούμενο από δύο δεκαδικά ψηφία που αντιπροσωπεύουν το χαρακτήρα στο σύνολο χαρακτήρων του φυλλομετρητή. Αυτές οι μετατροπές γίνονται επιρόσθετα από την τοποθέτηση του συμβόλου ίσον (=) μετάξυ του ονόματος κάθε καταχώρισης και της αντίστοιχης τιμής, καθώς και την τοποθέτηση του εμπορικού και (ε) μετάξυ των ζευγαριών ονομάτων/τιμών.

Για παράδειγμα, η Εικόνα 19-5 παρουσιάζει μια έκδοχή της σελίδας `GetForm.html` (Λίστα 19.1) όπου για το όνομα πληκτρολογήσαμε "Larry (Java Hacker)". Όπως μπορείτε να δείτε στην Εικόνα 19-6, αυτή η καταχώριση στέλνεται ως `"Larry+%28Java+Hacker%3F%29"`. Αυτό γίνεται επειδή τα διαστήματα γίνονται σύμβολα συν, το 28 είναι η τιμή ASCII για την αριστερή παρένθεση (στο δεκαεξαδικό σύστημα), το 3F είναι η τιμή ASCII για το αγγλικό ερωτηματικό, και το 29 είναι η τιμή ASCII για τη δεξιά παρένθεση.

Εικόνα 19-5 Προσρμοσμένο αποτέλεσμα της σελίδας `GetForm.html`.

19.2 Το στοιχείο FORM

Σημειώστε ότι, αν δεν καθοριστεί κάτι διαφορετικό, τα δεδομένα POST κωδικοποιούνται και αυτά ως `application/x-www-form-urlencoded`.



Εικόνα 19-6 Αίτηση HTTP που στάλθηκε από τον Internet Explorer 6.0 κατά την υποβολή της φόρμας `GetForm.html` με τα δεδομένα που φαίνονται στην Εικόνα 19-5.

Οι περισσότεροι σύγχρονοι φυλλομετρητές υποστηρίζουν μια επιπλέον κωδικοποίηση `ENTYPE`, την `multipart/form-data`. Αυτή η κωδικοποίηση μεταδίδει κάθε πεδίο ως ξεχωριστό τμήμα ενός συμβατού με τους τύπους MIME εγγράφου. Για να χρησιμοποιήσετε αυτή την κωδικοποίηση `ENTYPE`, θα πρέπει να καθορίσετε ως μέθοδο την `POST`. Αυτή η κωδικοποίηση κάνει μερικές φορές ευκολότερο το χειρισμό πολύπλοκων δεδομένων από το πρόγραμμα στο διακομιστή, και είναι απαραίτητη όταν χρησιμοποιείτε χειριστήρια ανεβασματος αρχείων για την αποστολή ολόκληρων εγγράφων (δείτε την Ενότητα 19.7). Για παράδειγμα, η Λίστα 19.3 παρουσιάζει μια φόρμα που διαφέρει από την `GetForm.html` (Λίστα 19.1) στο ότι το στοιχείο

```
<FORM ACTION="http://localhost:8088/SomeProgram">
```

έχει γίνει

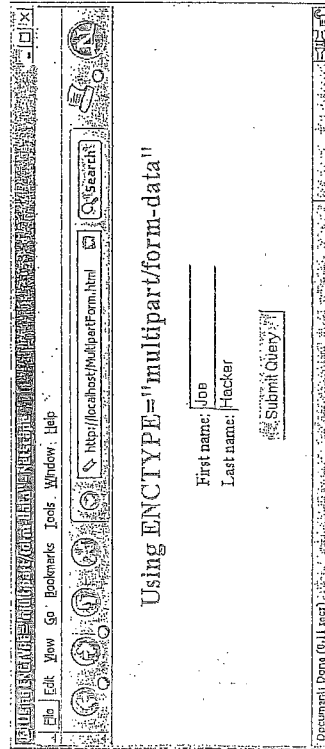
```
<FORM ACTION="http://localhost:8088/SomeProgram"
```

```
ENCTYPE="multipart/form-data" METHOD="POST">
```

Οι Εικόνες 19-7 και 19-8 δείχνουν το αποτέλεσμα.

Windows 9x
MultipartForm.html

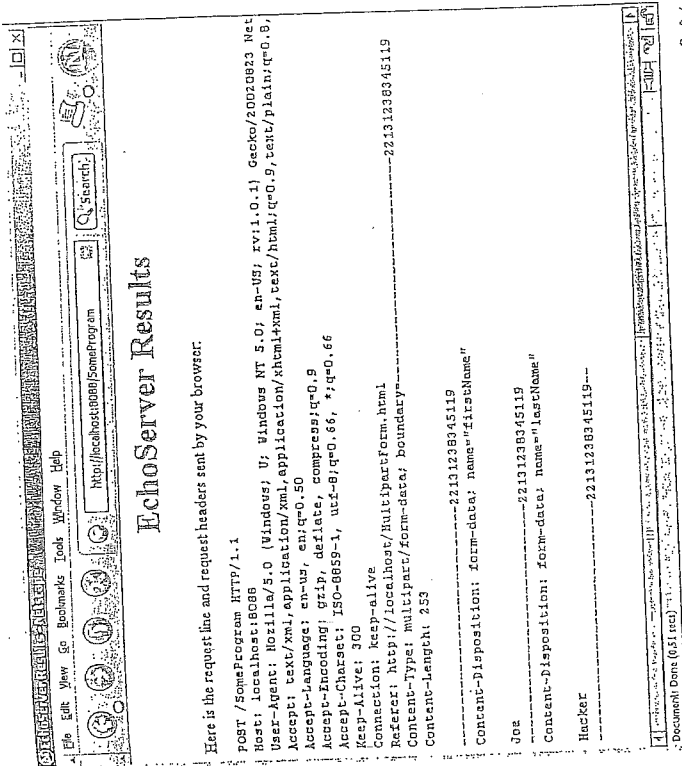
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>Using ENCTYPE="multipart/form-data"</TITLE>
</HEAD>
<BODY BGCOLOR="#FDF5E6">
<CENTER>
<H2>Using ENCTYPE="multipart/form-data"</H2>
<FORM ACTION="http://localhost:8088/SomeProgram"
ENCTYPE="multipart/form-data" METHOD="POST">
First name:
<INPUT TYPE="TEXT" NAME="firstName" VALUE="Joe"><BR>
Last name:
<INPUT TYPE="TEXT" NAME="lastName" VALUE="Hacker"><P>
<INPUT TYPE="SUBMIT">
</FORM>
</CENTER>
</BODY></HTML>
```



Εικόνα 19-7 Αρχικό αποτέλεσμα της σελίδας MultipartForm.html.

TARGET.

Η ιδιότητα TARGET χρησιμοποιείται από τους φυλλομετρητές που έχουν δυνατότητα επεξεκόνστις πλαίσιων (frame-compatible) για να προσδιορίσουν ποιο από τα τμήματα του κειμένου πλαίσιων θα πρέπει να χρησιμοποιηθεί για την εμφάνιση των αποτελεσμάτων της μικροεπιθεώρησης, της σελίδας USP, ή του όποιου άλλου προγράμματος που χειρίζεται την υποβολή της φόρμας. Η προεπιλεγμένη τιμή είναι η εμφάνιση των αποτελεσμάτων στο τμήμα πλαίσιων που περιέχει τη φόρμα η οποία υποβάλλεται.



Εικόνα 19-8 Αίτηση HTTP που στάλθηκε από το Netscape 7.0 κατά την υποβολή της σελίδας MultipartForm.html.

ONCEBIMTTI KAI ONRESET

ONSUBMIT και ONKBESE1

Αντές οι ιδιότητες χρησιμοποιούνται από την JavaScript για την προσάρτηση κώδικα. Αντές οι ιδιότητες χρησιμοποιούνται κατά την υποβολή της φόρμας ή κατά την επαναφορά της που πρέπει να εκτελεστεί ανή την υποβολή της φόρμας. Για την UNSUBMIT, αν η παράσταση πάρει φόρμας στην αρχική κατάσταση, αντίστοιχα. Αν το σος επιτρέπεται να καλέσετε κώδικα την τιμή false η φόρμα δεν υποβάλλεται. Αν το σος επιτρέπεται να καλέσετε κώδικα JavaScript στον πεδίο το οποίο θα ελέγξει τη μορφή των τιμών των πεδίων της φόρμας πριν από την υποβολή, ενημερώνοντας το χρήστη για ελλείψεις ή ακατάλληλες καταχωρίσεις.

ACCEPT-CHARSET

ACCEPT και ACCEPT-CHARSET. Αυτές οι ιδιότητες είναι νέες στον HTML 4.0, και καθορίζουν τους τύπους MIME (ACCEPT) και τις κωδικοποιήσεις χαρακτήρων (ACCEPT-CHARSET) που θα πρέπει να γίνουν αποδεκτές από τη μικρουπηρεσία ή το οποίο πρόγραμμα που επεξεργάζεται τα δεδομένα της φόρμας. Οι τύποι MIME που περιλαμβάνονται στην ιδιότητα ACCEPT

μπορούν επίσης να χρησιμοποιηθούν από τον πελάτη έτσι ώστε να περιορίσει τους τύπους αρχείων που θα εμφανίζονται στο χρήστη σε πλαίσια ανεβάσιματος αρχείων.

19.3 Χειριστήρια κειμένου

Η HTML υποστηρίζει τρεις τύπους στοιχείων εισόδου κειμένου: τα πεδία κειμένου, τα πεδία κωδικών πρόσβασης, και τις περιοχές κειμένου. Σε όλα αυτά τα στοιχεία αποδίδετε εσείς ένα όνομα και η τιμή λογβάνεται από τα περιεχόμενα του χειριστήριου. Το όνομα και η τιμή αποστέλλονται στο διακομιστή κατά την υποβολή της φόρμας, η οποία γίνεται συνήθως μέσω ενός κουμπιού υποβολής (δείτε την Ενότητα 19.4).

Πεδία κειμένου

Στοιχείο HTML: `<INPUT TYPE="TEXT" NAME="..." ...>`
(Χωρίς επικέφαλο)

Ιδιότητες: NAME (υποχρεωτική), VALUE, SIZE, MAXLENGTH, ONCHANGE, ONSELECT, ONFOCUS, ONBLUR, ONKEYDOWN, ONKEYPRESS, ONKEYUP

Αυτό το στοιχείο δημιουργεί ένα πεδίο εισόδου μίας γραμμής στην οποία ο χρήστης μπορεί να πληκτρολογήσει κείμενο, όπως είδαμε στις Λίστες 19.1, 19.2, και 19.3. Αν θέλετε να δημιουργήσετε πεδία κειμένου με πολλές γραμμές, δείτε την ιδιότητα TEXTAREA στην υποενότητα που ακολουθεί. Αν και στις φόρμες εισόδου (INPUT) ο προεπιλεγμένος τύπος στοιχείου είναι ο TEXT, στις συνιστούμε (για λόγους σαφήνειας) να ορίζετε ρητά τον τύπο TEXT. Θα πρέπει να έχετε υπόψη σας ότι στο εσωτερικό των στοιχείων FORM ισχύει η κανονική αναδίπλωση κειμένου που εφαρμόζουν οι φυλλομετρητές, έτσι θα πρέπει χρησιμοποιείτε κατάλληλες εντολές HTML για να διασφαλίσετε ότι ο φυλλομετρητής δεν θα διαχωρίσει το περιγραφικό κείμενο από το αντίστοιχο πεδίο κειμένου.

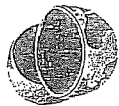
Η προσέγγιση του βιβλίου



Χρησιμοποιήστε ρητές εντολές HTML για να ομαδοποιήσετε τα πεδία κειμένου και το περιγραφικό κείμενό τους.

Αν η φόρμα έχει κάποιο κουμπί SUBMIT (για λεπτομέρειες σχετικά με το κουμπί SUBMIT δείτε την Ενότητα 19.4), το Netscape 7.0 και ο Internet Explorer 6.0 υποβάλλουν τη φόρμα όταν ο χρήστης πατά το πλήκτρο Enter με το δρομέα μέσα σε κάποιο πεδίο κειμένου. Αυτή η συμπεριφορά, όμως, δεν επιβάλλεται από τις προδιαγραφές της HTML, και άλλοι φυλλομετρητές μπορεί να συμπεριφέρονται διαφορετικά.

Προειδοποίηση



Μη βασίζεστε στο φυλλομετρητή για την υποβολή της φόρμας όταν ο χρήστης πατά το πλήκτρο Enter ενώ βρίσκεται σε κάποιο πεδίο κειμένου. Θα πρέπει πάντα να συμπεριλαμβάνετε ένα κουμπί ή μια εικόνα που να υποβάλλει ρητά τη φόρμα.

Για να παρεμποδίσετε την υποβολή της φόρμας όταν ο χρήστης πατά το πλήκτρο Enter σε ένα πεδίο κειμένου, χρησιμοποιήστε ένα στοιχείο BUTTON με το χειριστή συμβάντος onClick αντί για ένα κουμπί SUBMIT. Για παράδειγμα, μπορείτε να χρησιμοποιήσετε το ακόλουθο στοιχείο

```
<INPUT TYPE="BUTTON" VALUE="Check Values"
onClick="submit()">
```

αντί του στοιχείου

```
<INPUT TYPE="SUBMIT">
```

για την υποβολή της φόρμας σας.

Οι υποενότητες που ακολουθούν περιγράφουν τις ιδιότητες που ισχύουν ειδικά για πεδία κειμένου. Δεν εξετάζουμε ιδιότητες που ισχύουν σε γενικά στοιχεία HTML (όπως STYLE, CLASS, ID). Η ιδιότητα TABINDEX, η οποία ισχύει για όλα τα στοιχεία φόρμας, εξετάζεται στην Ενότητα 19.11 (Χειριστήρια σειράς ενεργοποίησης).

NAME

Η ιδιότητα NAME προσδιορίζει το πεδίο κειμένου κατά την υποβολή της φόρμας. Στην τυπική HTML, η ιδιότητα αυτή είναι υποχρεωτική. Επειδή τα δεδομένα αποστέλλονται πάντοτε στο διακομιστή με τη μορφή ξενιγίων ονομάτων/τιμών, δεν αποστέλλονται δεδομένα για τα χειριστήρια φόρμας στα οποία δεν έχει οριστεί η ιδιότητα NAME.

VALUE

Η ιδιότητα VALUE, αν έχει οριστεί, καθορίζει τα αρχικά περιεχόμενα του πεδίου κειμένου. Όταν υποβάλλεται η φόρμα αποστέλλονται τα τρέχοντα περιεχόμενα· αυτά μπορεί να αντανakούν την είσοδο του χρήστη. Αν το πεδίο κειμένου είναι κενό κατά την υποβολή της φόρμας, τα δεδομένα φόρμας αποστέλλονται απλά από το όνομα και ένα σύμβολο ισότητας (π.χ., name1=value1 & textfieldname=εname3= value3).

SIZE

Αυτή η ιδιότητα καθορίζει το πλάτος του πεδίου κειμένου, με βάση το μέσο όρο πλάτους των χαρακτήρων της γραμματοσειράς που χρησιμοποιείται. Αν το κείμενο είναι μεγαλύτερο από το μέγεθος που έχει οριστεί, πραγματοποιείται αντόματα κύλιση στο πεδίο κειμένου. Αυτό θα μπορούσε να συμβεί στην περίπτωση που ο χρήστης έχει πλη-

κτρολογήσει περισσότερους χαρακτήρες από τον αριθμό SIZE, ή αν έχει πληκτρολογήσει συνολικά SIZE ευρείς χαρακτήρες (π.χ., κεφαλαίο W) σε μια γραμματοσειρά αναλογικού πλάτους. Το Netscape και ο Internet Explorer 6.0 χρησιμοποιούν αυτόματα αναλογική γραμματοσειρά για τα πεδία κειμένου. Δυστυχώς, δεν μπορείτε να αλλάξετε τη γραμματοσειρά ενσωματώνοντας ένα στοιχείο INPUT σε ένα στοιχείο FONT ή CODE. Μπορείτε όμως να χρησιμοποιήσετε φύλλα στύλ (cascading style sheets, CSS) για να αλλάξετε τη γραμματοσειρά των στοιχείων εισόδου. Για παράδειγμα, το παρακάτω φύλλο στύλ (που είναι τοποθετημένο στην ενότητα HEAD της σελίδας HTML) θα εμφανίσει το κείμενο όλων των στοιχείων INPUT σε μορφή Futura 12 στίγγιών (με την προϋπόθεση ότι η γραμματοσειρά Futura είναι εγκατεστημένη στο μηχάνημα του πελάτη).

```
<style type="text/css">
INPUT {
  font-size : 12pt;
  font-family : Futura;
}
</style>
```



Η προσέγγιση του βιβλίου

Εξ ορισμού, το Netscape και ο Internet Explorer εμφανίζουν τα στοιχεία INPUT σε ανάλογη γραμματοσειρά. Για να αλλάξετε τη γραμματοσειρά για τα στοιχεία INPUT, χρησιμοποιήστε φύλλα στύλ.

MAXLENGTH

Η ιδιότητα MAXLENGTH καθορίζει το μέγιστο αριθμό των επιτρεπόμενων χαρακτήρων. Αυτός ο αριθμός διαφέρει από τον αριθμό των ορατών χαρακτήρων, ο οποίος καθορίζεται μέσω της ιδιότητας SIZE. Πρέπει όμως να σημειώσουμε ότι οι χρήστες μπορούν να τον παρακάμνουν για τις αιτήσεις GET μπορούν να πληκτρολογήσουν δεδομένα κατευθείαν στο URL, και για τις αιτήσεις POST μπορούν να γράψουν τη δική τους φόρμα HTML. Έτσι το πρόγραμμα στο διακομιστή δεν θα πρέπει να θεωρεί ότι η αίτηση περιέχει την κατάλληλη ποσότητα δεδομένων.

ONCHANGE, ONSELECT, ONFOCUS, ONBLUR, ONDBLDOWN, ONKEYPRESS, και ONKEYUP

Αυτές οι ιδιότητες χρησιμοποιούνται μόνο από τους φυλλομετρητές που υποστηρίζουν την JavaScript. Καθορίζουν την ενέργεια που θα πραγματοποιηθεί όταν ο δρομέας βγει από το πεδίο κειμένου μετά την πραγματοποίηση κάποιας αλλαγής, όταν ο χρήστης επλάξει κείμενο στο πεδίο κειμένου, όταν το πεδίο κειμένου αποκτήσει την εστίαση εισόδου (input focus), όταν το πεδίο κειμένου πάψει να έχει την εστίαση εισόδου, και όταν θα πατηθούν ή θα αποδεσμευτούν, αντίστοιχα, μειονομμένα πλήκτρα.

Πεδία κωδικού πρόσβασης

Στοιχείο HTML: `<INPUT TYPE="PASSWORD" NAME="..." ...>`
(Χωρίς ετικέτα τέλους)

Ιδιότητες: NAME (υποχρεωτική), VALUE, SIZE, MAXLENGTH, ONCHANGE, ONSELECT, ONFOCUS, ONBLUR, ONKEYDOWN, ONKEYPRESS, ONKEYUP

Τα πεδία κωδικού πρόσβασης δημιουργούνται και χρησιμοποιούνται όπως και τα πεδία κειμένου, με τη διαφορά ότι όταν ο χρήστης πληκτρολογεί κείμενο η είσοδος δεν εμφανίζεται στην οθόνη. Αντίθετα, εμφανίζεται κάποιος χαρακτήρας μάσκας, όπως ο αστερίσκος (δείτε την Εικόνα 19-9). Η απόκρυψη εισόδου είναι χρήσιμη για τη συλλογή δεδομένων όπως οι αριθμοί των πιστωτικών καρτών ή οι κωδικοί πρόσβασης, τους οποίους ο χρήστης δεν θα ήθελε να είναι ορατοί από άλλους που μπορεί να βρίσκονται δίπλα στον υπολογιστή του. Όταν υποβάλλεται η φόρμα, ως τιμή του πεδίου μεταδίδεται το κανονικό (και όχι το μασκαρισμένο) κείμενο.

Επειδή τα δεδομένα GET προσαρτάνται στη διεύθυνση URL μετά από ένα ερωτηματικό, θα πρέπει πάντοτε να χρησιμοποιείτε τη μέθοδο POST όταν έχετε πεδία κωδικού πρόσβασης, έτσι ώστε να μην μπορούν οι υπόλοιποι να διαβάσουν τον κωδικό πρόσβασης από το URL που εμφανίζεται στη γραμμή διευθύνσεων του φυλλομετρητή. Επιπλέον, για λόγους ασφαλείας κατά τη μετάδοση, θα πρέπει να εξετάσετε τη δυνατότητα χρήσης υποδοχών SSL, με τις οποίες κρυπτογραφούνται τα δεδομένα. Για περισσότερες πληροφορίες σχετικά με τη χρήση υποδοχών SSL δείτε τα κεφάλαια σχετικά με την ασφάλεια των εφαρμογών Ιστού, στο βιβλίο More Servlets and JavaServer Pages.

Η προσέγγιση του βιβλίου



Για να προστατεύσετε το απόρρητο των πληροφοριών του χρήστη, να χρησιμοποιείτε πάντοτε τη μέθοδο POST όταν δημιουργείτε φόρμες που περιέχουν πεδία κωδικού πρόσβασης. Για πρόσθετη ασφάλεια μεταδώστε τα δεδομένα με το πρωτόκολλο https://, όπου χρησιμοποιείται κρυπτογράφηση SSL για τα δεδομένα.

NAME, VALUE, SIZE, MAXLENGTH, ONCHANGE, ONSELECT, ONFOCUS, ONBLUR, ONKEYDOWN, ONKEYPRESS, και ONKEYUP

Οι ιδιότητες των πεδίων κωδικού πρόσβασης χρησιμοποιούνται ακριβώς όπως και εκείνες για τα πεδία κειμένου.

Enter Password:

Εικόνα 19-9 Ένα πεδίο κωδικού πρόσβασης που έχει δημιουργηθεί με την εντολή
 <INPUT TYPE="PASSWORD" ...>.

ΠΕΡΙΟΧΕΣ ΚΕΙΜΕΝΟΥ

Παράδειγμα HTML: <TEXTAREA NAME="..."
 ROWS=xxx COLS=yyy> ...
 </TEXTAREA>

Ιδιότητες: NAME (υποχρεωτική), ROWS (υποχρεωτική), COLS (υποχρεωτική),
 WRAP (μη τυποποιημένη), ONCHANGE, ONSELECT, ONFOCUS,
 ONBLUR, ONKEYDOWN, ONKEYPRESS, ONKEYUP

Η ιδιότητα TEXTAREA δημιουργεί μια περιοχή κειμένου με πολλές γραμμές· δείτε σχετικά την Εικόνα 19-10. Το στοιχείο αυτό δεν διαθέτει ιδιότητα VALUE· αντίθετα, το κείμενο μεταξύ των επικεφαλών αρχής και τέλους χρησιμοποιείται ως αρχικό περιεχόμενο της περιοχής κειμένου. Ο χειρισμός του αρχικού κειμένου μεταξύ των ετικετών <TEXTAREA ...> και </TEXTAREA> γίνεται όπως και με το κείμενο που βρίσκεται στο εσωτερικό του (παρωχημένου πλέον) στοιχείου XMP. Με άλλα λόγια, διατηρούνται τα κενά διαστήματα σε αυτό το αρχικό κείμενο, και οι σή-
 μανσεις HTML μεταξύ των επικεφαλών αρχής και τέλους ερμηνεύονται "καριολεξικά", εκτός από τις αντίστροφες γραμμάτιες (όπως <, ©, κ.ο.κ.) που ερμηνεύονται κανονικά. Αν στη φόρ-
 μα δεν χρησιμοποιείται κάποια προσαρμοσμένη κωδικοποίηση ENCTYPE (δείτε σχετικά την Ενό-
 τητα 19.2, "Το στοιχείο FORM"), τότε στους χαρακτήρες, συμπεριλαμβανομένων και εκείνων που παράγονται από οντότητες χαρακτήρων, εφαρμόζεται κωδικοποίηση URL (URL-encoded) πριν από τη μετάδοσή τους. Έτσι τα διαστήματα γίνονται σύμβολα συν και οι άλλοι μη αλφαριθ-
 μητικοί χαρακτήρες γίνονται %xx, όπου xx είναι η αριθμητική τιμή του χαρακτήρα στο δεκαεξα-
 δικό σύστημα.

NAME

Αυτή η ιδιότητα καθορίζει το όνομα που θα αποσταλεί στο διακομιστή.

ROWS

Η ιδιότητα ROWS καθορίζει τον αριθμό των ορατών γραμμών κειμένου. Αν εισαχθούν περισσότερες γραμμές κειμένου, προστίθεται στην περιοχή κειμένου μια κατακόρυφη ράβδος κύλισης.

COLS

Η ιδιότητα COLS καθορίζει το ορατό πλάτος της περιοχής κειμένου, με βάση το μέσο πλάτος των χαρακτήρων της γραμματοσειράς που χρησιμοποιείται. Στο Netscape 7.0 και στον Internet Explorer 6.0, αν το κείμενο κάποιας γραμμής περιέχει περισσότερους χαρακτήρες από όσους επιτρέπει το καθορισμένο πλάτος, τότε το κείμενο αναδιπλώνε-

19.3 Χειριστήρια κειμένου

τα στην επόμενη γραμμή. Αν όμως μια λέξη έχει περισσότερους χαρακτήρες από το καθορισμένο πλάτος, ο Internet Explorer 6.0 αναδιπλώνει τη λέξη στην επόμενη γραμμή ενώ το Netscape 7.0 προσθέτει μια οριζόντια ράβδο κύλισης για να διατηρήσει τη λέξη στην ίδια γραμμή. Άλλοι φυλλομετρητές μπορεί να συμπεριφέρονται διαφορετικά.

ONCHANGE, ONSELECT, ONFOCUS, ONBLUR, ONKEYDOWN, ONKEYPRESS, και ONKEYUP

Αυτές οι ιδιότητες εφαρμόζονται μόνο σε φυλλομετρητές που υποστηρίζουν την JavaScript· καθορίζουν τον κώδικα που θα εκτελεστεί αν προκύψουν κάποιες συγκε-
 κριμένες συνθήκες. Η ONCHANGE χειρίζεται την περίπτωση που η περιοχή κειμένου χά-
 νει την εστίαση εισόδου μετά από αλλαγή του περιεχομένου της, η ONSELECT περιγρά-
 φει τι θα γίνει όταν επιλεγεί από το χρήστη το κείμενο της περιοχής κειμένου, οι ONFO-
 CUS και ONBLUR καθορίζουν τι θα γίνει αν η περιοχή κειμένου αποκτήσει ή χάσει την
 εστίαση εισόδου, και οι υπόλοιπες ιδιότητες καθορίζουν τι θα γίνει αν πατηθούν
 μεμονομένα πλήκτρα.

Η Λίστα 19.4 δημιουργεί μια περιοχή κειμένου με 5 ορατές γραμμές που μπορεί να περιέχει 30 χαρακτήρες ανά γραμμή. Το αποτέλεσμα φαίνεται στην Εικόνα 19-10.

Λίστα 19.4

Παράδειγμα χειριστήριου φόρμας TEXTAREA

```
<P>
<P>
Enter some HTML:<BR>
<TEXTAREA NAME="HTML" ROWS=5 COLS=30>
Delete this text and replace
with some HTML to validate.
</TEXTAREA>
<CENTER>
```

Enter some HTML:

Delete this text and replace
 with some HTML to validate.

Εικόνα 19-10 Μια περιοχή κειμένου στο Netscape 7.0.

19.4 Κουμπιά

Τα κουμπιά (push buttons) χρησιμοποιούνται κυρίως για δύο λόγους στις φόρμες HTML: για την υποβολή φορμών και για την επαναφορά των χειριστηρίων στις τιμές που καθορίστηκαν στον αρχικό κώδικα HTML. Οι φυλλομετρητές που χρησιμοποιούν JavaScript μπορούν επίσης να χρησιμοποιήσουν τα κουμπιά για έναν τρίτο λόγο: για την ενεργοποίηση κώδικα JavaScript.

Κατά παράδοση, τα κουμπιά δημιουργούνται από στοιχεία INPUT στα οποία η ιδιότητα TYPE έχει τιμή SUBMIT, RESET, ή BUTTON. Στην HTML 4.0 προστέθηκε ένα στοιχείο BUTTON, που υποστηρίζεται από τον Internet Explorer 6.0 και το Netscape 7.0. Αυτό το νέο στοιχείο σας επιτρέπει να δημιουργείτε κουμπιά τα οποία έχουν ετικέτες πολλών γραμμών, εικόνες, κείμενο σε διάφορες γραμματοσειρές, και πολλά άλλα. Όμως οι παλαιότεροι φυλλομετρητές πιθανόν να μην υποστηρίζουν το στοιχείο BUTTON.

Κουμπιά υποβολής

Στοιχείο HTML: <INPUT TYPE="SUBMIT" ...> (Χωρίς ετικέτα τέλους)

Ιδιότητες: NAME, VALUE, ONCLICK, ONDBLCLICK, ONFOCUS, ONBLUR

Όταν ο χρήστης πατά σε ένα κουμπί υποβολής (submit), η φόρμα αποστέλλεται στη μικροϋπηρεσία ή το πρόγραμμα του διακομιστή που καθορίζεται από την παράμετρο ACTION του στοιχείου FORM. Αν και η ενέργεια αυτή μπορεί να πραγματοποιηθεί και με άλλους τρόπους, όπως με το πλάτημα που χρήστη σε ένα χάρτη εικόνας (image map), οι περισσότερες φόρμες διαθέτουν τουλάχιστον ένα κουμπί υποβολής. Τα κουμπιά υποβολής, όπως και τα υπόλοιπα χειριστήρια φόρμας, έχουν την εμφάνιση και τη λειτουργικότητα που καθορίζεται από το λειτουργικό σύστημα του πελάτη, έτσι μπορεί να φαίνονται διαφορετικά στις διάφορες πλατφόρμες. Η Εικόνα 19-11 παρουσιάζει ένα κουμπί υποβολής στα Windows 2000 Professional, το οποίο δημιουργήθηκε με την εντολή

```
<INPUT TYPE="SUBMIT">
```

Submit Query

Εικόνα 19-11 Ένα κουμπί υποβολής με την προεπιλεγμένη ετικέτα.

NAME και VALUE

Τα περισσότερα στοιχεία εισόδου έχουν ένα όνομα και μια αντίστοιχη τιμή. Όταν υποβάλλεται η φόρμα, τα ονόματα και οι τιμές των ενεργών στοιχείων σενανώνονται για να σχηματίσουν ένα αλφαριθμητικό δεδομένων. Αν το κουμπί υποβολής χρησιμοποιείται μόνο για την έναρξη της υποβολής της φόρμας, το όνομα του κουμπιού μπορεί να παραλειφθεί αφού δεν συμμετέχει στο αλφαριθμητικό δεδομένων που αποστέλλεται. Αν δοθεί το όνομα, τότε αποστέλλονται μόνο το όνομα και η τιμή του κουμπιού που πραγματικά πατήθηκε. Αυτή η δυνατότητα σας επιτρέπει να χρησιμοποιήσετε περισσότερα

από ένα κουμπιά υποβολής και να ανιχνεύσετε ποιο από αυτά πατήθηκε. Η ετικέτα χρησιμοποιείται ως μεταδιδόμενη τιμή. Η χρήση μιας ρητής ιδιότητας VALUE θα αλλάξει την προεπιλεγμένη ετικέτα.

Για παράδειγμα, η λίστα 19.5 δημιουργεί ένα πεδίο κειμένου και δύο κουμπιά υποβολής, όπως φαίνεται στην Εικόνα 19-12. Αν, για παράδειγμα, πατήσει το πρώτο κουμπί, το αλφαριθμητικό δεδομένων που θα σταλεί στο διακομιστή θα είναι το Item=256MB+SIMM&Add=Add+Item+to+Cart.

Λίστα 19.5 Παράδειγμα χειριστηρίων εισόδου SUBMIT

```
<CENTER>
Item:
<INPUT TYPE="TEXT" NAME="Item" VALUE="256MB SIMM"><BR>
<INPUT TYPE="SUBMIT" NAME="Add"
VALUE="Add Item to Cart">
<INPUT TYPE="SUBMIT" NAME="Delete"
VALUE="Delete Item from Cart">
</CENTER>
```

Item: 256MB SIMM

Add Item to Cart Delete Item from Cart

Εικόνα 19-12 Κουμπιά υποβολής με προσαρμοσμένες ετικέτες.

Πρέπει να σημειώσουμε ότι, όταν τα δεδομένα της φόρμας υποβάλλονται σε μια μικροϋπηρεσία, η μέθοδος request.getParameter επιστρέφει τιμή null για τα κουμπιά που δεν έχουν πατηθεί. Έτσι θα μπορούσατε να χρησιμοποιήσετε έναν απλό έλεγχο για την τιμή null, όπως φαίνεται εδώ, για να προσδιορίσετε ποιο κουμπί πατήθηκε.

```
if (request.getParameter("Add") != null) {
doCartAdditionOperation(...);
} else if (request.getParameter("Delete") != null) {
doCartDeletionOperation(...);
}
```

ONCLICK, ONDBLCLICK, ONFOCUS, και ONBLUR

Αυτές οι μη τυπικές ιδιότητες χρησιμοποιούνται από φυλλομετρητές που έχουν δυνατότητα χρήσης JavaScript για τη συσχέτιση κώδικα JavaScript με το κουμπί. Ο κώδικας ONCLICK και ONDBLCLICK εκτελείται όταν ο χρήστης πατά (ή διπλοπατά) στο κουμπί, ο κώδικας ONFOCUS εκτελείται όταν το κουμπί λαμβάνει την εστίαση εισόδου, και ο κώδικας ONBLUR εκτελείται όταν το κουμπί χάνει την εστίαση εισόδου. Αν ο κώδικας που έχει προσωρθηθεί στο κουμπί επιστρέφει τιμή false, αναστέλλεται η υποβολή της

φόρμας. Οι ιδιότητες HTML δεν κάνουν διάκριση πηλών και κεφαλαίων χαρακτήρων, και οι προγραμματιστές JavaScript συνήθως ονομάζουν τις ιδιότητες αυτές `onClick`, `ondblClick`, `onFocus`, και `onBlur`.

ΣΤΟΙΧΕΙΟ HTML: <BUTTON TYPE="SUBMIT" ...>
ΚΩΔΙΚΟΣ HTML
</BUTTON>

bioinfores: NAME, VALUE, ONCLICK, ONDBLCLICK, ONFOCUS, ONBLUR

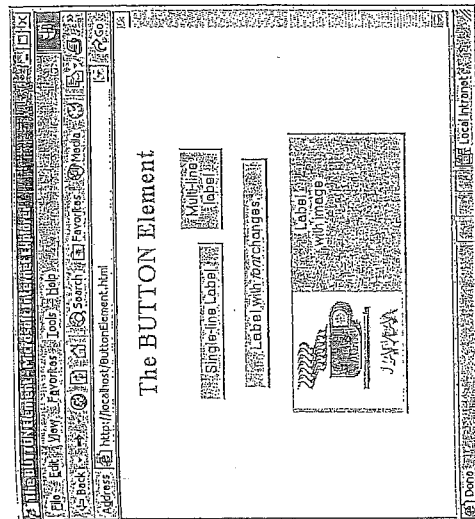
Αυτός ο εναλλακτικός τρόπος δημιουργίας κουμπιών υποβολής σας επιτρέπει να χρησιμοποιήσετε το δικό σας κώδικα HTML για τα περιεχόμενα του κουμπιού. Αυτό το στοιχείο σας επιτρέπει να έχετε επικείμες κουμπιό που καταλαμβάνουν περισσότερες από μία γραμμές κειμένου, επικείμες κουμπιό με τις διάφορες γραμματοσειρές, κουμπιό εικόνας, κ.ο.κ. Η Λίστα 19.6 παρουσιάζει μερικά παραδείγματα και η Εικόνα 19-13 δείχνει τα αποτελέσματα.

NAME, VALUE, ONCLICK, ONDBLCLICK, ONFOCUS, ONBLUR

Αυτές οι ιδιότητες χρησιμοποιούνται με τον ίδιο τρόπο όπως στο στοιχείο

```
<INPUT TYPE="SUBMIT" ...>
```

ButtonElement.html

[illegible]

ΕΚΔΗΛΩΣΗ 19-13 Κουμπιέρ υποβολής που δημιουργούνται με το στοιχείο EUTTON.

Κουιτιά επαναφοράς

Στοίσις,ο HTML: <INPUT TYPE="RESET" ...> (Χωρίς ετικέτα τέλους)

ISóntec	VALUE	NAME	ONCLICK	ONDBCLICK	ONFOCUS	ONBLUR
1	1	1	1	1	1	1

Με τα κουμπιά επαναφοράς (reset buttons) ο χρήστης μπορεί να επαναφέρει τις τιμές όλων των στοιχείων της φόρμας σε εκείνες που είχαν καθοριστεί από τις αρχικές παραμέτρους VALUE. Η τιμή των κομπιτών αυτών δεν μεταδίδεται ποτέ ως τιμήμα των περιεχόμενων της φόρμας.

VALLE

Η ιδιότητα VALUE καθορίζει την επικέτα του κουμπιού· η προεπιλεγμένη τιμή είναι "Reset".

NAME:

Επειδή τα cookies επαναφοράς δεν συμπεριέχουν στο αλφριθμητικό δεδομένων που μεταδίδεται κατά την υποβολή της φόρμας, στην τυπική HTML τα cookies αυτά δεν μεταδίδονται. Όμως η JavaScript επιτρέπει τη χρήση μιας ιδιότητας NAME έτσι ώστε να σταλτοποιηθεί η αναφορά στο στοιχείο αυτό.

ONCLICK, ONDBLCLICK, ONFOCUS, and ONBLUR

Ανέσεις οι μη τυπικές ιδιότητες χρησιμοποιούνται από φυλλομετρήτες που έχουν δυνατότητα JavaScript για τη συσχέτιση κώδικα. JavaScript με το κουμπί. Ο κώδικας των ON-CLICK και ONDCLICK εκτελείται όταν ο χρήστης πατά (ή διπλοπατά) στο κουμπί, ο

κώδικας ONFOCUS εκτελείται όταν το κουμπί λαμβάνει την εστίαση εισόδου, και ο κώδικας ONBLUR εκτελείται όταν το κουμπί χάνει την εστίαση εισόδου. Οι ιδιότητες HTML δεν κάνουν διάκριση πεζών-κεφαλαίων, έτσι οι προγραμματιστές JavaScript συνήθως ονομάζουν αυτές τις ιδιότητες onclick, ondblclick, onfocus, και onblur.

Στοιχείο HTML: <BUTTON TYPE="RESET" ...>
 Κώδικας HTML
 </BUTTON>

Ιδιότητες: VALUE, NAME, ONCLICK, ONDBLCLICK, ONFOCUS, ONBLUR

Αυτός ο εναλλακτικός τρόπος δημιουργίας κουμπιών επαναφοράς σας επιτρέπει να χρησιμοποιήσετε το δικό σας κώδικα HTML για τα περιεχόμενα του κουμπιού. Οι ιδιότητες χρησιμοποιούνται με ίδιο τρόπο όπως και στο στοιχείο <INPUT TYPE="SUBMIT" ...>.

Κουμπί JavaScript

Στοιχείο HTML: <INPUT TYPE="BUTTON" ...> (Χωρίς επικέτα-πέλους)

Ιδιότητες: NAME, VALUE, ONCLICK, ONDBLCLICK, ONFOCUS, ONBLUR

Αυτό το στοιχείο αναγνωρίζεται μόνο από φυλλομετρήτες που υποστηρίζουν την JavaScript. Δημιουργεί ένα κουμπί με ίδια οπτική εμφάνιση όπως και τα κουμπιά SUBMIT ή RESET, και επιτρέπει στο δημιουργό της σελίδας να προσαρτήσει κώδικα JavaScript στις ιδιότητες ONCLICK, ONDBLCLICK, ONFOCUS, ή ONBLUR. Το ζευγάρι ονόματος/τιμής που συσχετίζεται με το κουμπί JavaScript δεν μεταδίδεται ως τμήμα των δεδομένων κατά την υποβολή της φόρμας. Μπορείτε να συσχετίσετε με το κουμπί όποιον κώδικα θέλετε, αλλά μια από τις πιο συνηθισμένες χρήσεις είναι όταν θέλετε να βεβαιωθείτε ότι όλα τα στοιχεία εισόδου έχουν την κατάλληλη μορφή πριν από την υποβολή της φόρμας στο διακομιστή. Για παράδειγμα, ο κώδικας που ακολουθεί δημιουργεί ένα κουμπί το οποίο, όταν ο χρήστης πατά σε αυτό, καλεί τη συνάρτηση validateForm.

<INPUT TYPE="BUTTON" VALUE="Check Values"
 onclick="validateForm()">

Στοιχείο HTML: <BUTTON TYPE="BUTTON" ...>
 Κώδικας HTML
 </BUTTON>

Ιδιότητες: NAME, VALUE, ONCLICK, ONDBLCLICK, ONFOCUS, ONBLUR

Αυτός ο εναλλακτικός τρόπος δημιουργίας κουμπιών JavaScript σας επιτρέπει να χρησιμοποιήσετε κώδικα HTML για τα περιεχόμενα του κουμπιού. Όλας οι ιδιότητες χρησιμοποιούνται όπως και στο στοιχείο <INPUT TYPE="BUTTON" ...>.

19.5 Πλαίσια ελέγχου και ραδιοπλήκτρα

Τα πλαίσια ελέγχου (check boxes) και τα ραδιοπλήκτρα ή κουμπιά επιλογής (radio buttons) είναι χρήσιμα χειριστήρια που επιτρέπουν στο χρήστη να κάνει την επιλογή του από ένα σύνολο προ-καθορισμένων επιλογών. Αν και στα πλαίσια ελέγχου η ενεργοποίηση ή η απενεργοποίηση μπορεί να γίνει μεμονωμένα για το κάθε πλαίσιο, τα ραδιοπλήκτρα μπορούν να είναι ομαδοποιημένα έτσι ώστε να μπορεί να επιλεγεί κάθε φορά μόνο ένα από τα μέλη της ομάδας.

Πλαίσια ελέγχου

Στοιχείο HTML: <INPUT TYPE="CHECKBOX" NAME="..." ...>
 (Χωρίς επικέτα-πέλους)

Ιδιότητες: NAME (υποχρεωτική), VALUE, CHECKED, ONCLICK, ONFOCUS, ONBLUR

Αυτό το στοιχείο εισόδου δημιουργεί ένα πλαίσιο ελέγχου του οποίου το ζευγάρι όνομα/τιμή μεταδίδεται κατά την υποβολή της φόρμας μόνο αν το πλαίσιο ελέγχου είναι ενεργοποιημένο. Για παράδειγμα, ο κώδικας που ακολουθεί δημιουργεί το πλαίσιο ελέγχου που φαίνεται στην Εικόνα 19-14.

<P>
 <INPUT TYPE="CHECKBOX" NAME="noEmail" CHECKED>
 Check here if you do <I>not</I> want to
 get our email newsletter

☒ Check here if you do *not* want to get our email newsletter

Εικόνα 19-14 Ένα πλαίσιο ελέγχου HTML.

Σημειώστε ότι το περιγραφικό κείμενο που συγγίζεται με το πλαίσιο ελέγχου είναι σε κανονική γλώσσα HTML, και θα πρέπει να φροντίσετε εσείς να εξασφαλίσετε ότι θα εμφανιστεί δίπλα από το πλαίσιο ελέγχου. Έτσι, η επικέτα <P> στο προηγούμενο παράδειγμα διασφαλίζει ότι το πλαίσιο ελέγχου δεν θα αποτελεί τμήμα της προηγούμενης παραγράφου.



Η προσέγγιση του βιβλίου

Οι παράγραφοι στο εσωτερικό ενός στοιχείου FORM συμπληρώνονται και αναδίδονται όπως και οι κανονικές παράγραφοι της ιστοσελίδας. Έτσι, θα πρέπει να φροντίσετε να χρησιμοποιήσετε ρητές εντολές HTML για να διατηρήσετε τα στοιχεία εισόδου μαζί με το κείμενο που τα περιγράφει.

Προειδοποίηση

Βεβαιωθείτε ότι η ιδιότητα NAME κάθε ραδιοπλήκτρου μιας λογικής ομάδας είναι ακριβώς ίδια με εκείνη των υπολοίπων μελών της ομάδας.



VALUE

Η ιδιότητα VALUE παρέχει την τιμή που θα μεταδοθεί με τη NAME κατά την υποβολή της φόρμας. Η ιδιότητα αυτή δεν επιτρέπει την εμφάνιση του ραδιοπλήκτρου. Αντίθετα, γύρω από το ραδιοπλήκτρο τοποθετούνται κανονικό κείμενο και κώδικας HTML, όπως συμβαίνει και στα πλαίσια ελέγχου.

CHECKED

Αν καθοριστεί η ιδιότητα CHECKED, τότε το ραδιοπλήκτρο θα είναι αρχικά ενεργοποιημένο όταν φορτωθεί η αντίστοιχη ιστοσελίδα. Διαφορετικά δεν θα είναι καταρχήν ενεργοποιημένο.

ONCLICK, ONFOCUS, και ONBLUR

Αυτές οι ιδιότητες καθορίζουν τον κώδικα JavaScript που θα εκτελεστεί όταν ο χρήστης πατήσει στο ραδιοπλήκτρο, όταν το ραδιοπλήκτρο λάβει την εστίαση εισόδου, και όταν το ραδιοπλήκτρο χάσει την εστίαση εισόδου, αντίστοιχα.

19.6 Σύνθετα πλαίσια και πλαίσια καταλόγου

Το στοιχείο SELECT παρουσιάζει στο χρήστη ένα σύνολο επιλογών. Αν ο χρήστης μπορεί να επιλέξει μόνο μία από τις καταχωρίσεις του καταλόγου αυτού και δεν έχει καθοριστεί συγκεκριμένο μέγεθος εμφάνισης, τότε οι επιλογές παρουσιάζονται σε ένα σύνθετο πλαίσιο (combo box) ή πτυσσόμενο μενού (drop-down menu). Τα πλαίσια καταλόγου (list box) χρησιμοποιούνται όταν ο χρήστης επιτρέπεται να διαλέξει περισσότερες από μία επιλογές ή όταν έχει καθοριστεί συγκεκριμένο μέγεθος εμφάνισης του καταλόγου. Οι επιλογές καθορίζονται με καταχωρίσεις OPTION που είναι ενσωματωμένες στο στοιχείο SELECT. Η τυπική μορφή είναι η εξής:

```
<SELECT NAME="Όνομα" ...>
  <OPTION VALUE="Τιμή1">Κε(μενο Επιλογής 1
  <OPTION VALUE="Τιμή2">Κε(μενο Επιλογής 2
  ...
  <OPTION VALUE="ΤιμήN"> Κε(μενο Επιλογής N
</SELECT>
```

19.6 Σύνθετα πλαίσια και πλαίσια καταλόγου

Οι προδιαγραφές της HTML 4.0 ορίζουν επίσης το στοιχείο OPGROUP (με μία μόνο ιδιότητα, την LABEL) όπου εσωκλείονται στοιχεία OPTION για τη δημιουργία επάλληλων μενού.

Στοιχείο HTML: <SELECT NAME = "... " ...> </SELECT>

Ιδιότητες: NAME (υποχρεωτική), SIZE, MULTIPLE, ONCLICK, ONFOCUS, ONBLUR, ONCHANGE

Το στοιχείο SELECT δημιουργεί ένα σύνθετο πλαίσιο ή ένα πλαίσιο καταλόγου για την επιλογή μεταξύ των διαφόρων καταχωρίσεων. Κάθε επιλογή καθορίζεται με ένα στοιχείο OPTION το οποίο περικλείεται μεταξύ των ετικετών <SELECT ...> και </SELECT>.

NAME

Η ιδιότητα NAME προσδιορίζει το όνομα στοιχείου που θα σταλεί κατά την υποβολή της φόρμας.

SIZE

Η ιδιότητα SIZE καθορίζει τον αριθμό γραμμών που θα είναι ορατές. Αν χρησιμοποιηθεί η ιδιότητα SIZE, το στοιχείο SELECT παρουσιάζεται συνήθως ως πλαίσιο καταλόγου, αντί για σύνθετο πλαίσιο. Το σύνθετο πλαίσιο είναι η κανονική αναπαράσταση όταν δεν έχει καθοριστεί ούτε η ιδιότητα SIZE, ούτε η ιδιότητα MULTIPLE.

MULTIPLE

Η ιδιότητα MULTIPLE καθορίζει ότι μπορούν να επιλεγούν ταυτόχρονα περισσότερες από μία καταχωρίσεις. Αν δεν καθοριστεί η ιδιότητα MULTIPLE, τότε επιτρέπεται η επιλογή μόνο μίας καταχώρισης. Από μια μικροϋπηρεσία, θα πρέπει να χρησιμοποιήσετε τη μέθοδο request.getParameterValues για να πάρετε έναν πίνακα με τις καταχωρίσεις που έχουν επιλεγεί. Για παράδειγμα, ο κώδικας

```
String[] listValues = request.getParameterValues("language");
if (listValues != null) {
    for (int i=0; i<listValues.length; i++) {
        String value = listValues[i];
        ...
    }
}
```

θα σας επιτρέψει να επεξεργαστείτε όλες τις τιμές που έχουν επιλεγεί σε ένα πλαίσιο καταλόγου με το όνομα language (δείτε τη Λίστα 19.8). Θα πρέπει να θυμάστε ότι η σειρά με την οποία εμφανίζονται οι τιμές στον πίνακα μπορεί να μην αντιστοιχεί με τη σειρά με την οποία εμφανίζονται οι τιμές στη λίστα.

Η προσέγγιση του βιβλίου



Αν σε μία λίστα `SELECT` είναι δυνατή η επιλογή πολλών καταχωρήσεων, χρησιμοποιήστε τη μέθοδο `request.getParameterValues` για να πάρετε έναν πίνακα με τα επιλεγμένα στοιχεία.

ONCLICK, ONFOCUS, ONBLUR, και ONCHANGE

Αυτές οι μη τυπικές ιδιότητες υποστηρίζονται από φυλλομετρητές που επιτρέπουν κώδικα JavaScript. Οι ιδιότητες αυτές προορίζουν τον κώδικα που θα εκτελεστεί όταν ο χρήστης πατά σε μια καταχώριση, όταν το στοιχείο αποικτά την εστίαση εισόδου, όταν το στοιχείο χάνει την εστίαση εισόδου, και όταν το στοιχείο χάνει την εστίαση εισόδου μετά από τροποποίηση της επιλεγμένης καταχώρισης, αντίστοιχα.

Στοιχείο HTML: `<OPTION ...>` <Προαιρετική ετικέτα τέλους>

Ιδιότητες: `SELECTED`, `VALUE`

Αυτή η ιδιότητα καθορίζει τις επιλογές του καταλόγου· είναι έγκυρη μόνο μέσα σε ένα στοιχείο `SELECT`.

VALUE

Η ιδιότητα `VALUE` καθορίζει την τιμή που θα μεταδοθεί μαζί με το όνομα (`NAME`) του στοιχείου `SELECT` αν επιλεγεί η τρέχουσα καταχώριση. Δεν είναι το κείμενο που εμφανίζεται στο χρήστη· το κείμενο αυτό καθορίζεται με ξεχωριστό κώδικα HTML, μετά από την ετικέτα `OPTION`.

SELECTED

Αν καθοριστεί, η ιδιότητα `SELECTED` προσδιορίζει ότι το συγκεκριμένο στοιχείο του μενού θα είναι επιλεγμένο κατά τη φόρτωση της σελίδας.

Η Λίστα 19.8 δημιουργεί ένα πτυσσόμενο μενού με επιλογές γλωσσών προγραμματισμού. Επειδή επιτρέπεται μόνο μία επιλογή από τον κατάλογο αυτόν και δεν έχει καθοριστεί ιδιότητα `SIZE`, το μενού εμφανίζεται ως σύνθετο πλαίσιο. Οι Εικόνες 19-16 και 19-17 δείχνουν την αρχική εμφάνιση του στοιχείου και την εμφάνιση του στοιχείου αφού ο χρήστης ενεργοποιήσει το μενού, πατώντας σε αυτό. Αν κατά τη στιγμή της υποβολής της φόρμας ο χρήστης έχει διαλέξει την καταχώριση `java`, τότε στο πρόγραμμα στο διακομιστή θα σταλεί η καταχώριση `language=java`. Παρατηρήστε ότι μεταδίδεται η ιδιότητα `VALUE`, και όχι το περιγραφικό κείμενο.

19.6 Σύνθετα πλαίσια και πλαίσια καταλόγου

Λίστα 19.8 Παράδειγμα μενού `SELECT`

```
Favorite language:
<SELECT NAME="language">
<OPTION VALUE="c">c
<OPTION VALUE="c++">c++
<OPTION VALUE="java" SELECTED>java
<OPTION VALUE="lisp">lisp
<OPTION VALUE="perl">perl
<OPTION VALUE="smalltalk">smalltalk
</SELECT>
```

Favorite language:

Εικόνα 19-16 Ένα στοιχείο `SELECT` που εμφανίζεται ως σύνθετο πλαίσιο (πτυσσόμενο μενού).

Favorite language:

Java
C
C++
Java
Lisp
Perl
Smalltalk

Εικόνα 19-17 Επιλογή μιας καταχώρισης από ένα στοιχείο `SELECT`.

Το δεύτερο παράδειγμα δείχνει ένα στοιχείο `SELECT` που εμφανίζεται ως πλαίσιο καταλόγου. Αν κατά την υποβολή της φόρμας έχουν επιλεγεί περισσότερες από μία καταχωρήσεις, τότε θα σταλούν περισσότερες από μία τιμές ως ξεχωριστές καταχωρήσεις (με επανάληψη της ιδιότητας `NAME`). Για παράδειγμα, στο παράδειγμα που παρουσιάζεται στη Λίστα 19.9 (Εικόνα 19-18), στα δεδομένα που αποστέλλονται στο διακομιστή θα προστεθεί η παράσταση `language=java&language=perl`. Οι πολλαπλές καταχωρήσεις που μοιράζονται το ίδιο όνομα είναι η αιτία για την οποία οι συγγραφείς μικροϋπηρεσιών, εκτός από την πιο συνηθισμένη μέθοδο `getParameter`, θα πρέπει να είναι επίσης εξοικειωμένοι με τη μέθοδο `getParameterValues` της κλάσης `HttpServletRequest`. Δείτε σχετικά το Κεφάλαιο 4 (Χειρισμός αιτήσεων πελάτη: Δεδομένα φόρμας).

Λίστα 19-9

Παράδειγμα ενός μενού SELECT που επιτρέπει την επιλογή πολλών καταχωρίσεων

```

Languages you know: <BR>
<SELECT NAME="language" MULTIPLE>
  <OPTION VALUE="c">c
  <OPTION VALUE="c++">c++
  <OPTION VALUE="java" SELECTED>Java
  <OPTION VALUE="lisp">lisp
  <OPTION VALUE="perl" SELECTED>Perl
  <OPTION VALUE="smalltalk">Smalltalk
</SELECT>

```

Languages you know.

C
C++
Java
Lisp
Perl
Smalltalk

Εικόνα 19-18 Τα στοιχεία SELECT στα οποία καθορίζεται η ιδιότητα MULTIPLE ή η ιδιότητα SIZE εμφανίζονται ως πλαίσια καταλόγου.

Στοιχείο HTML: <OPTGROUP ...> ...</OPTGROUP>

Ιδιότητες: LABEL (υποχρεωτική)

Αυτό το στοιχείο, που υποστηρίζεται από το Netscape 7.0 και τον Internet Explorer 6.0, επιτρέπει την ομαδοποίηση επιλογών μενού. Είναι έγκυρο μόνο στο εσωτερικό ενός στοιχείου SELECT.

LABEL

Η ιδιότητα LABEL καθορίζει το κείμενο που θα εμφανιστεί στην ομάδα των επιλογών μενού. Το Netscape και ο Internet Explorer εμφανίζουν την ετικέτα σε έντονη και πλάγια γραφή.

Η Λίστα 19.10 δημιουργεί ένα μενού με επιλογές γλώσσας για το διακομιστή. Εδώ οι επιλογές μενού κατηγοριοποιούνται σε δύο ομάδες, μέσα του στοιχείου OPTGROUP. Η πρώτη ομάδα είναι η Common Servlet Languages, και η δεύτερη ομάδα είναι η Common CGI Languages. Η Εικόνα 19-19 δείχνει το μενού που θα εμφανιστεί με επιλεγμένο το στοιχείο Java από την ομάδα Common Servlet Languages. Για αυτή την επιλογή τα δεδομένα αίτησης που θα σταλούν στο διακομιστή είναι language=java1. Θα πρέπει να θυμάστε ότι δεν αποστέλλονται από το διακομιστή επιπλέον πληροφορίες στο Netscape ή τον Internet Explorer, οι οποίες να δείχνουν ποια καταχώριση OPTGROUP έχει επιλεγεί. Κατά συνέπεια, θα πρέπει να χρησιμοποιείτε μοναδικές τιμές για όλες τις επιλογές μενού, ανεξάρτητα από την ομάδα.

Η προσέγγιση του βιβλίου



Όταν χρησιμοποιείτε πολλά στοιχεία OPTGROUP, εξασφαλίστε ότι για όλα τα στοιχεία OPTION έχετε καθορίσει μοναδική τιμή για την ιδιότητα VALUE.

Λίστα 19-10

Παράδειγμα ενός μενού SELECT που χωρίζεται σε δύο ομάδες μέσω του στοιχείου OPTGROUP

```

Server-side Languages:
<SELECT NAME="language">
  <OPTGROUP LABEL="Common Servlet Languages">
    <OPTION VALUE="java1">Java
  </OPTGROUP>
  <OPTGROUP LABEL="Common CGI Languages">
    <OPTION VALUE="c">c
    <OPTION VALUE="c++">c++
    <OPTION VALUE="java2">Java
    <OPTION VALUE="perl">Perl
    <OPTION VALUE="vb">Visual Basic
  </OPTGROUP>
</SELECT>

```

Server-side Languages:

Java
Common Servlet Languages
Common CGI Languages
Java
C
C++
Java
Perl
Visual Basic

Εικόνα 19-19 Ένα στοιχείο SELECT που χρησιμοποιεί την ιδιότητα OPTGROUP για την ομαδοποίηση των επιλογών μενού, όπως φαίνεται στο Netscape 7.0.

19.7 Χειριστήρια ανεβάσματος αρχείων

Στοιχείο HTML: `<INPUT TYPE="FILE" ... >` (Προαιρετική ετικέτα τέλους)

Ιδιότητες: NAME (υποχρεωτική), VALUE (δεν λαμβάνεται υπόψη), SIZE, MAXLENGTH, ACCEPT, ONCHANGE, ONSELECT, ONFOCUS, ONBLUR (μη τυπική)

Αυτό το στοιχείο έχει αποτέλεσμα τη δημιουργία ενός πεδίου κειμένου ονόματος αρχείου δίπλα από ένα κουμπί Browse (Αναζήτηση). Οι χρήστες μπορούν να πληκτρολογήσουν στο πεδίο κειμένου μια διαδρομή ή να πατήσουν στο κουμπί για να εμφανίσουν ένα πλαίσιο διαλόγου επιλογής αρχείων, το οποίο σας επιτρέπει να επιλέξετε με αλληλεπιδραστικό τρόπο τη διαδρομή ενός αρχείου. Κατά την υποβολή της φόρμας μεταδίδονται τα περιεχόμενα του αρχείου — εφόσον βέβαια έχει καθοριστεί κατά την αρχική δήλωση του στοιχείου FORM η τιμή `multipart/form-data` για την ιδιότητα `ENCTYPE`. Για δεδομένα πολλών τμημάτων (`multipart data`) θα πρέπει να καθορίσετε ως μέθοδο αποστολής την `POST`. Αυτό το στοιχείο παρέχει ένα βολικό τρόπο δημιουργίας σελίδων υποστήριξης χρηστών, με τις οποίες ο χρήστης στέλνει μία περιγραφή του προβλήματος μαζί με τα σχετικά δεδομένα ή τα αρχεία διευθέτησης.

Η προσέγγιση του βιβλίου



Στις φόρμες με χειριστήρια ανεβάσματος αρχείων θα πρέπει πάντοτε να καθορίζετε τις ιδιότητες `ENCTYPE="multipart/form-data"` και `METHOD="POST"`.

Δυστυχώς, η API των μικροϋπηρεσιών δεν παρέχει εργαλεία υψηλού επιπέδου για την ανάλυση των αρχείων που ανέβασε ο χρήστης: θα πρέπει εσείς οι ίδιοι να καλέσετε τη μέθοδο `request.getInputStream` και να αναλύσετε συντακτικά την αίτηση. Ευτυχώς, γι' αυτή την εργασία υπάρχουν διαθέσιμες πολυάριθμες βιβλιοθήκες τρίτων κατασκευαστών. Μια από τις πιο δημοφιλείς προέρχεται από τη βιβλιοθήκη Jakarta Commons για λειτουργίες, δείτε τη διεύθυνση <http://jakarta.apache.org/commons/fileupload/>.

NAME

Η ιδιότητα `NAME` επιτρέπει στο πρόγραμμα που εκτελείται στο διακομιστή να αναγνωρίσει το πεδίο κειμένου.

VALUE

Για λόγους ασφαλείας, αυτή η ιδιότητα δεν λαμβάνεται υπόψη· μόνο ο τελικός χρήστης μπορεί να καθορίσει ένα όνομα αρχείου. Διαφορετικά, κάποιος κακόβουλος δημιουργός σελίδων HTML θα μπορούσε να κλέψει αρχεία του πελάτη καθορίζοντας ένα όνομα αρχείου και μετά χρησιμοποιώντας κώδικα JavaScript για αυτόματη υποβολή της φόρμας κατά τη φόρτωση της σελίδας.

19.7 Χειριστήρια ανεβάσματος αρχείων

SIZE και MAXLENGTH

Οι ιδιότητες `SIZE` και `MAXLENGTH` χρησιμοποιούνται με τον ίδιο τρόπο όπως και στα πεδία κειμένου, και καθορίζουν τον αριθμό των ορατών χαρακτήρων και το μέγιστο αριθμό των επιτρεπόμενων χαρακτήρων, αντίστοιχα.

ACCEPT

Η ιδιότητα `ACCEPT` είχε σκοπό να είναι μια λίστα με τύπους MIME, που διαχωρίζονται με κόμματα, με στόχο τον περιορισμό των διαθέσιμων ονομάτων αρχείων. Ωστόσο, πολύ λίγοι φυλλομετρητές υποστηρίζουν αυτή την ιδιότητα.

ONCHANGE, ONSELECT, ONFOCUS, και ONBLUR

Αυτές οι ιδιότητες χρησιμοποιούνται από φυλλομετρητές που υποστηρίζουν την JavaScript για τον καθορισμό της ενέργειας που θα πραγματοποιηθεί όταν ο χρήστης από το πεδίο κειμένου μετά την πραγματοποίηση κάποιων αλλαγών, όταν ο χρήστης επιλέξει κείμενο στο πεδίο κειμένου, όταν το πεδίο κειμένου αποκτήσει την εστίαση εισόδου, και όταν το πεδίο κειμένου χάσει την εστίαση εισόδου, αντίστοιχα.

Για παράδειγμα, ο κώδικας της λίστας 19.11 δημιουργεί ένα χειριστήριο ανεβάσματος αρχείου. Η Εικόνα 19-20 δείχνει το αρχικό αποτέλεσμα και η Εικόνα 19-21 δείχνει ένα τυπικό αναδυόμενο παράθυρο που εμφανίζεται όταν ο χρήστης πατά στο κουμπί Browse.

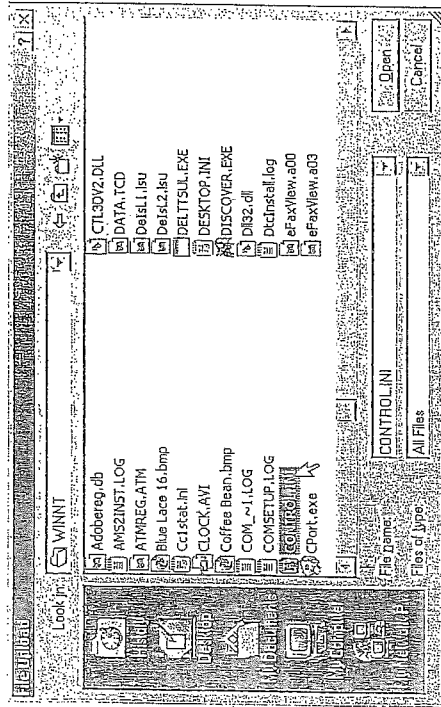
Λίστα 19.11 Προσέγγιση χειριστηρίου ανεβάσματος αρχείου

```
<FORM ACTION="http://localhost:8080/SomeProgram"
  ENCTYPE="multipart/form-data" METHOD="POST">
  Enter data file below:<BR>
  <INPUT TYPE="FILE" NAME="fileName">
</FORM>
```

Enter data file below:



Εικόνα 19-20 Αρχική εμφάνιση ενός χειριστηρίου ανεβάσματος αρχείου.



Εικόνα 19-21 Ένα πλαίσιο διαλόγου επιλογής αρχείου, που εμφανίστηκε με το πάτημα του κουμπιού Browse σε ένα χειριστήριο ανεβάσιματος αρχείου στα Windows 2000.

19.8 Χάρτες εικόνων στο διακομιστή

Στην τυπική HTML, τα στοιχεία MAP σας επιτρέπουν να συσχετίσετε URL με διάφορες περιοχές μιας εικόνας: κατόπιν, όταν ο χρήστης πατά με το ποντίκι σε κάποια από τις καθορισμένες περιοχές, ο φυλλομετρητής φορτώνει το κατάλληλο URL. Αυτή η μορφή αντιστοίχισης (mapping) είναι γνωστή σαν *χάρτης εικόνων στον πελάτη* (client-side image map), αφού ο προσδιορισμός του URL που θα κληθεί γίνεται στον πελάτη και δεν συμμετέχει κάποιο πρόγραμμα το οποίο εκτελείται στο διακομιστή. Η HTML υποστηρίζει επίσης και *χάρτες εικόνων στο διακομιστή* (server-side image maps), που μπορούν να χρησιμοποιηθούν σε φόρμες HTML. Σε αυτούς τους χάρτες σχεδιάζεται στην οθόνη μια εικόνα· όταν ο χρήστης πατά με το ποντίκι στην εικόνα, στέλλονται οι συντεταγμένες του ποντικιού σε ένα πρόγραμμα στο διακομιστή.

Οι χάρτες εικόνων στον πελάτη είναι ευκολότερες και πιο αποτελεσματικές από τους χάρτες στο διακομιστή, και θα πρέπει να χρησιμοποιούνται όταν το μόνο που θέλετε είναι να συσχετίσετε ένα σταθερό σύνολο διευθύνσεων URL με κάποιες προκαθορισμένες περιοχές εικόνων. Παρόλα αυτά, οι χάρτες εικόνων στο διακομιστή είναι κατάλληλες όταν το URL προκύπτει από υπολογισμό (π.χ., για μεταωρολογικούς χάρτες), όταν οι περιοχές αλλάζουν συχνά, ή όταν μαζί με την αίτηση πρέπει να συμπεριληφθούν και άλλα δεδομένα της φόρμας. Αυτή η ενότητα εξετάζει δύο προσεγγίσεις σχετικά με τους χάρτες εικόνων στο διακομιστή.

19.8 Χάρτες εικόνων στο διακομιστή

IMAGE — Τυπικοί χάρτες εικόνων στο διακομιστή

Ο συνηθισμένος τρόπος δημιουργίας χαρτών εικόνων στο διακομιστή είναι μέσω του στοιχείου `<INPUT TYPE="IMAGE" ... >` σε μια φόρμα.

Στοιχείο HTML: `<INPUT TYPE="IMAGE" ... >` (Χωρίς ειδικά τέλους)

Ιδιότητες: NAME (υποχρεωτική), SRC, ALIGN

Αυτό το στοιχείο εμφανίζει μια εικόνα η οποία, όταν ο χρήστης πατά σε αυτή, στέλνει τη φόρμα στη μικροϋπηρεσία ή σε κάποιο άλλο πρόγραμμα στο διακομιστή που καθορίζεται από το στοιχείο ACTION της φόρμας. Το ίδιο το όνομα δεν στέλνεται· αντίθετα, μεταδίδεται η παράσταση `name.x=ypros` και `name.y=ypros`, όπου `xpros` και `ypros` είναι οι συντεταγμένες του ποντικιού σε σχέση με την επάνω αριστερή γωνία της εικόνας.

NAME

Η ιδιότητα NAME χρησιμεύει στην αναγνώριση του στοιχείου μετά την υποβολή της φόρμας.

SRC

Η ιδιότητα SRC καθορίζει το URL της αντίστοιχης εικόνας.

ALIGN

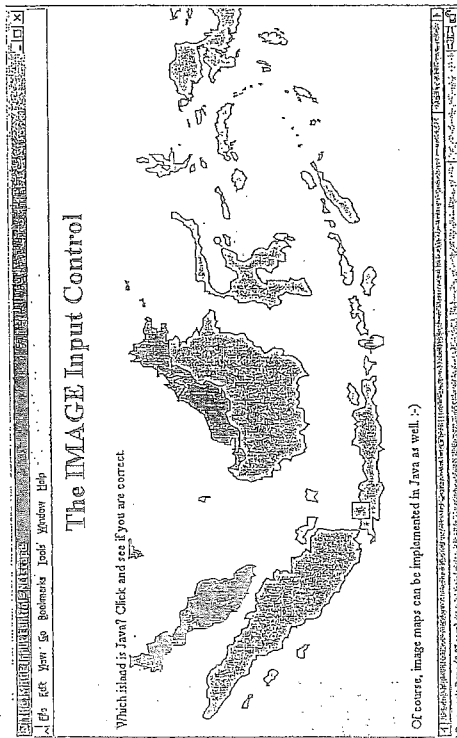
Η ιδιότητα ALIGN έχει τις ίδιες επιλογές (TOP, MIDDLE, BOTTOM, LEFT, RIGHT), και την ίδια προεπιλεγμένη τιμή BOTTOM, όπως και η ιδιότητα ALIGN του στοιχείου IMG, και χρησιμοποιείται με τον ίδιο τρόπο.

Η Λίστα 19.12 παρουσιάζει ένα απλό παράδειγμα, στο οποίο το στοιχείο ACTION της φόρμας καθορίζει μια θέση στο διακομιστή EchoServer, τον οποίο θα δούμε στην Ενότητα 19.12. Οι Εικόνες 19-22 και 19-23 δείχνουν τα αποτελέσματα πριν και μετά το πάτημα στην εικόνα.

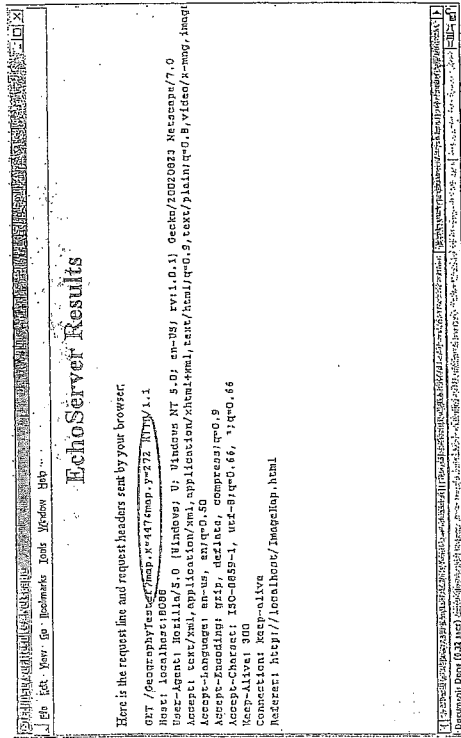
Λίστα 19.12 imageMap.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>The IMAGE Input Control</TITLE>
</HEAD>
<BODY>
<H1 ALIGN="CENTER">The IMAGE Input Control</H1>
Which island is Java? Click and see if you are correct.
<FORM ACTION="http://localhost:8088/GeographyTester">
  <INPUT TYPE="IMAGE" NAME="map" SRC="images/indonesia.gif">
</FORM>
Of course, image maps can be implemented <B>in</B>
Java as well. :-)
```

```
</BODY></HTML>
```



Εικόνα 19-22 Ένα χειριστήριο εισόδου IMAGE με NAME="map".



Εικόνα 19-23 Το πόντιμα στην εικόνα (στη θέση 447, 272) υποβάλλει τη φόρμα και προσβάει την παράσταση map . x=447 map . y=272 στα δεδομένα της φόρμας.

ISMAP — Εναλλακτικοί χάρτες εικόνων στο διακομιστή

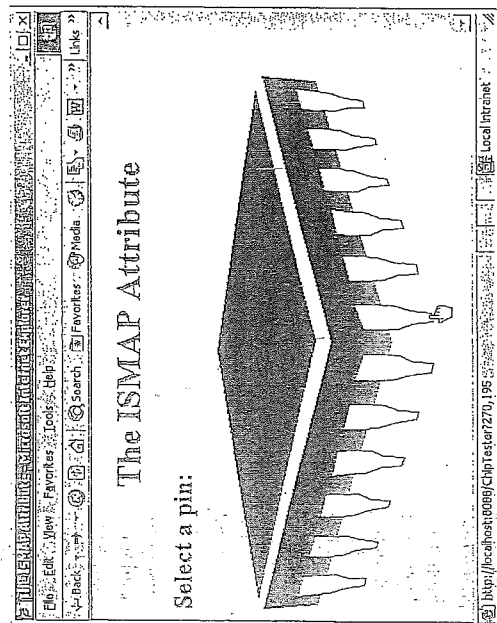
Η ιδιότητα ISMAP είναι μια προαιρετική ιδιότητα του στοιχείου IMG, και μπορεί να χρησιμοποιηθεί με τρόπο παρόμοιο με το στοιχείο φόρμας `<INPUT TYPE="IMAGE" ... >`. Στην πραγματικότητα, η ISMAP δεν είναι καθόλου στοιχείο FORM, αλλά παρόλα αυτά μπορεί να χρησιμοποιηθεί για απλές συνδέσεις με μικροϋπηρεσίες ή με άλλα προγράμματα στο διακομιστή. Αν μια εικόνα με την ιδιότητα ISMAP βρίσκεται στο εσωτερικό ενός συνδυασμένου υπερ-κειμένου, τότε το πόντιμα στην εικόνα έχει αποτέλεσμα την αποστολή των συντεταγμένων του ποντικού στη συγκεκριμένη διεύθυνση URL. Οι συντεταγμένες διαχωρίζονται με κόμματα και καθορίζονται σε ρίχνει σε σχέση με την επάνω αριστερή γωνία της εικόνας.

Για παράδειγμα, η Λίστα 19.13 ενσωματώνει μία εικόνα που χρησιμοποιεί την ιδιότητα ISMAP σε ένα σύνδεσμο υπερ-κειμένου προς τη διεύθυνση `http://localhost:8088/ChipTester`, στην οποία απαντά ο μίνι διακομιστής HTTP που παρουσιάζεται στην Ενότητα 19.12. Η Εικόνα 19-24 δείχνει την αρχική σελίδα, που είναι παρόμοια με εκείνη που θα εμφανιζόταν αν δεν είχε χρησιμοποιηθεί καθόλου η ιδιότητα ISMAP. Ωστόσο, αν πατήσουμε με το ποντίκι στη θέση που βρίσκεται 270 πίξελ δεξιά και 189 πίξελ κάτω από την επάνω αριστερή γωνία της εικόνας, ο φυλλομετρητής θα ζητήσει το URL `http://localhost:8088/ChipTester?270,189` (όπως φαίνεται στην Εικόνα 19-25).

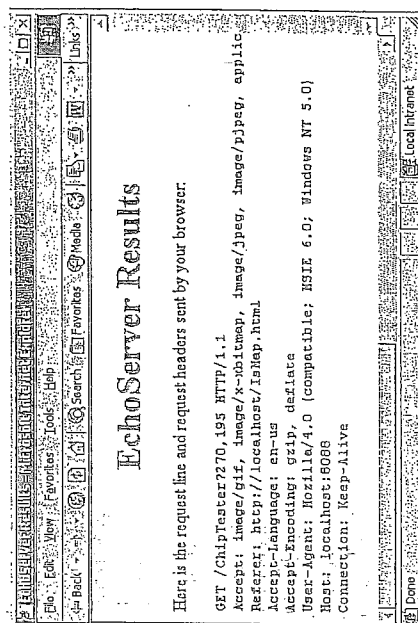
Όταν ο χάρτης εικόνων στο διακομιστή χρησιμοποιείται απλώς και μόνο για την επιλογή μεταξύ ενός στατικού συνόλου διευθύνσεων URL προορισμού, τότε τα στοιχεία MAP στον πόντιμα αποτελούν πολύ καλύτερη επιλογή επειδή δεν χρειάζεται να πραγματοποιηθεί επικοινωνία με το διακομιστή για να αποφασιστεί ποιο URL θα πρέπει να χρησιμοποιηθεί. Αν ο χάρτης εικόνων πρόκειται να αναμειχθεί με άλλα στοιχεία εισόδου, τότε είναι προτιμότερος ο τύπος εισόδου IMAGE. Ωστόσο, για έναν αυτόνομο χάρτη εικόνων όπου τα URL τα οποία έχουν αντιστοιχιστεί στις περισχές αλλάζουν συχνά ή απαιτούν υπολογισμό, η χρήση μιας εικόνας με την ιδιότητα ISMAP αποτελεί λογική επιλογή.

Λίστα 19.13 ISMAP.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>The ISMAP Attribute</TITLE>
</HEAD>
<BODY>
<H1 ALIGN="CENTER">The ISMAP Attribute</H1>
<H2>Select a pin:</H2>
<A href="http://localhost:8088/ChipTester">
<IMG SRC="images/chip.gif" WIDTH=495 HEIGHT=200 ALT="Chip"
  BORDER=0 ISMAP
</BODY></HTML>
```

Εικόνα 19-24 Ο ορισμός της ιδιότητας ISMAP για ένα στοιχείο IMG που βρίσκεται μέσα σε ένα σύνδεσμο υπερκειμένου αλλάζει το αποτέλεσμα που έχει το πάτημα στην εικόνα.



Εικόνα 19-25 Όταν ο χρήστης πατά σε μια εικόνα ISMAP, οι συντεταγμένες της επιλογής μεταδίδονται μαζί με το URL.

19.9 Κρυφά πεδία

Τα κρυφά πεδία δεν επιτρέφουν την εμφάνιση της σελίδας που παρουσιάζεται στο χρήστη. Αντίθετα, αποθηκεύουν σταθερά ονόματα και τιμές που αποστέλλονται στο διακομιστή ανεξάρτητα από την είσοδο του χρήστη. Τα κρυφά πεδία χρησιμοποιούνται συνήθως για τρεις σκοπούς:

- Παρακολούθηση του χρήστη. Καθώς ο χρήστης μετακινείται στο εσωτερικό μιας τοποθεσίας ιστού, η καταχώριση του αναγνωριστικού του χρήστη σε κρυφά πεδία μπορεί να χρησιμοποιηθεί για να μας επιτρέψει να παρακολουθήσουμε ποιες σελίδες έχει επισκεφθεί ο χρήστης, ή μπορεί να δείχνει τις επιλογές που έχει κάνει ο χρήστης. Στην πράξη, οι συγγραφείς μικροπληρεσίων βασίζονται συνήθως στην API παρακολούθησης συνεδριών και δεν προσπαθούν να υλοποιήσουν την παρακολούθηση συνεδριών σε τόσο χαμηλό επίπεδο. Για λεπτομέρειες σχετικά με την παρακολούθηση συνεδριών δείτε το Κεφάλαιο 9.
- Παροχή προκαθορισμένης εισόδου σε ένα πρόγραμμα στο διακομιστή. Όταν μια ποικιλία στατικών σελίδων HTML λειτουργούν ως εμπεσοφυλακές για το ίδιο πρόγραμμα που λειτουργεί στο διακομιστή, τα προκαθορισμένα κρυφά πεδία μπορούν να επιτρέψουν την παροχή πληροφοριών σχετικά με την πηγή σελίδα που κάνει την αίτηση. Για παράδειγμα, ένα ηλεκτρονικό κατάστημα μπορεί να πληρώνει προμήθεια σε αυτούς που στέλνουν πελάτες στην τοποθεσία του. Σε αυτό το σενάριο η σελίδα αναφοράς θα μπορούσε να επιτρέπει στους επισκέπτες να ψάξουν τον κατάλογο του καταστήματος μέσω μιας φόρμας, ενσωματώνοντας ταυτόχρονα ένα κρυφό πεδίο που θα δίνει το αναγνωριστικό της σελίδας αναφοράς.
- Αποθήκευση σχετικών πληροφοριών σε σελίδες που παράγονται δυναμικά. Για παράδειγμα, σε έναν πίνακα με είδη για ένα καλθι αγοράς, μπορείτε να τοποθετήσετε ένα κρυφό πεδίο σε κάθε γραμμή έτσι ώστε να προσδιορίζετε τον κωδικό του κάθε είδους. Με αυτόν τον τρόπο ο χρήστης θα μπορεί να τροποποιήσει τον αριθμό των ειδών που παραγγέλνει, και όταν το κρυφό πεδίο θα υποβληθεί στο πρόγραμμα του διακομιστή, θα προσδιορίζει το είδος που τροποποιήθηκε. Ο χρήστης δεν χρειάζεται να βλέπει τον κωδικό του είδους στη σελίδα HTML.

Πρέπει να σημειώσουμε ότι ο όρος "κρυφό" δεν σημαίνει ότι δεν μπορεί να ανακαλυφθεί το πεδίο αυτό ο χρήστης, καθώς το πεδίο είναι ορατό στον πηγαίο κώδικα της σελίδας HTML. Επειδή δεν υπάρχει αξιόπιστος τρόπος "απόκρυψης" του κώδικα HTML που παράγει τη σελίδα, οι συγγραφείς θα πρέπει να προσέχουν να μη χρησιμοποιούν "κρυφά" πεδία για την ενσωμάτωση κωδικών πρόσβασης ή άλλων ευαίσθητων πληροφοριών.

Στοιχείο HTML: `<INPUT TYPE="HIDDEN" NAME="..." VALUE="..." >`
(Χωρίς επικέτα +έλους)

Ιδιότητες: NAME (υποχρεωτική), VALUE

Αυτό το στοιχείο αποθηκεύει ένα όνομα και μια τιμή, αλλά δεν δημιουργείται στο φυλλομετρητή κάποιο αντίστοιχο ορατό στοιχείο. Το ζευγάρι ονόματος/τιμής προστίθεται στα δεδομένα της

φόρμας κατά την υποβολή της. Για παράδειγμα, στον κώδικα που ακολουθεί θα στέλνεται πάντοτε μαζί με τα δεδομένα της φόρμας και η παράσταση `itemID=brown001`.

```
<INPUT TYPE="HIDDEN" NAME="itemID" VALUE="brown001">
```

19.10 Ομάδες χειριστηρίων

Η HTML 4.0 ορίζει το στοιχείο `FIELDSET`, το οποίο μαζί με ένα στοιχείο `LEGEND` μπορεί να χρησιμοποιηθεί για την ομαδοποίηση χειριστηρίων στο εσωτερικό μιας φόρμας. Προσέξτε ότι το στοιχείο `FIELDSET` λειτουργεί μόνο στο Netscape 6 ή μεταγενέστερο και στον Internet Explorer 6.0 ή μεταγενέστερο.

Στοιχείο HTML: `<FIELDSET> ... </FIELDSET>`

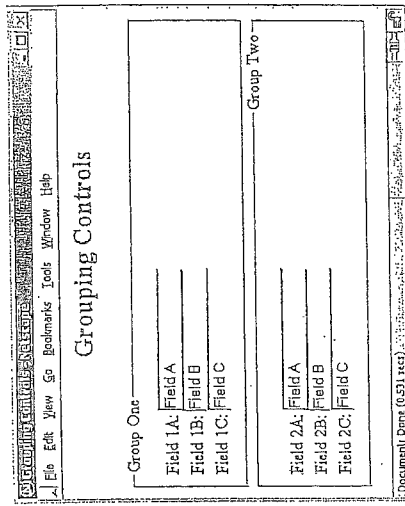
Ιδιότητες: Καμία

Το στοιχείο αυτό χρησιμοποιείται ως αποδέκτης όπου εσολογούνται χειριστήρια, σε συνδυασμό (προαιρετικά) με ένα στοιχείο `LEGEND`. Δεν έχει ιδιότητες, εκτός από τις καθολικές ιδιότητες για τα φύλλα στυλ, τη γλώσσα, κ.ο.κ. Η Λίστα 19.4 παρουσιάζει ένα παράδειγμα, ενώ το αποτέλεσμα φαίνεται στην Εικόνα 19-26.

Λίστα 19.14 `Fieldset.htm`

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>Grouping Controls</TITLE>
</HEAD>
<BODY BGCOLOR="#FDF5E6">
<H2 ALIGN="CENTER">Grouping Controls</H2>
<FORM ACTION="http://localhost:8088/SomeProgram">
<FIELDSET>
<LEGEND>Group One</LEGEND>
Field 1A: <INPUT TYPE="TEXT" NAME="field1A" VALUE="Field A"><BR>
Field 1B: <INPUT TYPE="TEXT" NAME="field1B" VALUE="Field B"><BR>
Field 1C: <INPUT TYPE="TEXT" NAME="field1C" VALUE="Field C"><BR>
</FIELDSET>
<FIELDSET>
<LEGEND ALIGN="RIGHT">Group Two</LEGEND>
Field 2A: <INPUT TYPE="TEXT" NAME="field2A" VALUE="Field A"><BR>
Field 2B: <INPUT TYPE="TEXT" NAME="field2B" VALUE="Field B"><BR>
Field 2C: <INPUT TYPE="TEXT" NAME="field2C" VALUE="Field C"><BR>
</FIELDSET>
</FORM>
</BODY></HTML>
```

19.11 Χειριστήριο σειράς ενεργοποίησης



Εικόνα 19-26 Το στοιχείο `FIELDSET` σας επιτρέπει να ομαδοποιήσετε "σχετιζόμενα" χειριστήρια, όπως βλέπετε σε αυτή τη σελίδα του Netscape 7.0.

Στοιχείο HTML: `<LEGEND> ... </LEGEND>`

Ιδιότητες: `ALIGN`

Αυτό το στοιχείο τοποθετεί μια ετικέτα στο πλαίσιο που σχεδιάζεται γύρω από την ομάδα των χειριστηρίων είναι έγκυρο μόνο μέσα στο στοιχείο `FIELDSET`.

ALIGN

Αυτή η ιδιότητα ελέγχει τη θέση της ετικέτας. Οι έγκυρες τιμές είναι `TOP`, `BOTTOM`, `LEFT`, και `RIGHT`, με προεπιλεγμένη τιμή την `TOP`. Στην Εικόνα 19-26 η πρώτη ομάδα έχει την προεπιλεγμένη στοίχιση ετικέτας, ενώ η δεύτερη ομάδα ορίζει την ιδιότητα `ALIGN="RIGHT"`. Στην HTML τα φύλλα στυλ αποτελούν συνήθως καλύτερη λύση για τη ρύθμιση της στοίχισης των στοιχείων, αφού επιτρέπουν την εφαρμογή των αλλαγών σε πολλές σελίδες.

19.11 Χειριστήριο σειράς ενεργοποίησης

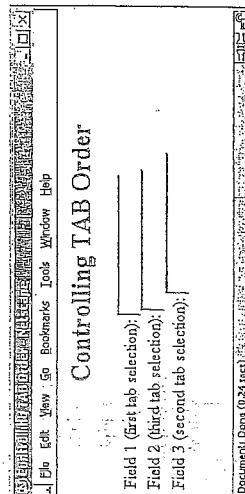
Η HTML 4.0 ορίζει μια ιδιότητα `TABINDEX` που μπορεί να χρησιμοποιηθεί με οποιοδήποτε ορατό στοιχείο της HTML. Η τιμή της ιδιότητας `TABINDEX` είναι ένας ακέραιος, και ρυθμίζει τη σειρά με την οποία τα στοιχεία θα αποκτήσουν την εστίαση εισόδου όταν ο χρήστης πατά το πλήκτρο `TAB`.

Στη Λίστα 19.15 παρουσιάζουμε τρία πεδία κειμένου, τα πεδία `field1`, `field2`, και `field3`. Εδώ η ιδιότητα `TABINDEX` έχει οριστεί έτσι ώστε η σειρά ενεργοποίησης να είναι από το `field1` στο `field3` και τέλος στο `field2`. Η σελίδα HTML φαίνεται στην Εικόνα 19-27.

Συνήθως η έμμεση σειρά ενεργοποίησης που χρησιμοποιείται από το φυλλομετρητή είναι από επάνω προς τα κάτω και από αριστερά προς τα δεξιά. Αν για την εφαρμογή σας είναι σημαντικό να ισχύει μια σειρά ενεργοποίησης που διαφέρει από την τυπική, θα πρέπει να την καθορίσετε ρητά.

Λίστα 19.15 Tabindex.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>Controlling TAB Order</TITLE>
</HEAD>
<BODY BGCOLOR="#FDF5E6">
<H2 ALIGN="CENTER">Controlling TAB Order</H2>
<FORM ACTION="http://localhost:8088/SomeProgram">
Field 1 (first tab selection):
<INPUT TYPE="TEXT" NAME="field1" TABINDEX=1><BR>
Field 2 (third tab selection):
<INPUT TYPE="TEXT" NAME="field2" TABINDEX=3><BR>
Field 3 (second tab selection):
<INPUT TYPE="TEXT" NAME="field3" TABINDEX=2><BR>
</FORM>
</BODY></HTML>
```



Εικόνα 19-27 Το διαδοχικό πάτημα του πλήκτρου TAB μεταφέρει την εστίαση εισόδου μεταξύ του πρώτου, του τρίτου, και του δεύτερου πεδίου κειμένου, με αυτή τη σειρά, όπως καθορίζεται από την ιδιότητα TABINDEX.

19.12 Ένας διακομιστής Ιστού για αποσφαλμάτωση

Σε αυτή την ενότητα θα παρουσιάσουμε ένα μίνι "διακομιστή Ιστού" που θα σας φανεί χρήσιμος όταν προσπαθείτε να καταλάβετε τη συμπεριφορά των φερμών HTML. Χρησιμοποιήσαμε το διακομιστή αυτόν σε πολλά από τα παραδείγματα που παρουσιάστηκαν σε αυτό το κεφάλαιο. Ο διακομιστής διαβάζει ακλώς όλα τα δεδομένα HTTP που του έχουν σταλεί από το φυλλομετρητή και επιστρέφει μια ιστοσελίδα με αυτές τις γραμμές ενσωματωμένες σε ένα στοιχείο PRE.

Αυτός ο διακομιστής είναι επίσης χρήσιμος για την αποσφαλμάτωση των μικροϋπηρεσιών. Όταν κάτι πηγαίνει στραβά, η πρώτη δουλειά είναι να προσδιορίσουμε αν το πρόβλημα βρίσκεται στον τρόπο συλλογής των δεδομένων ή στον τρόπο επεξεργασίας τους. Το πρόγραμμα Web-Client της Ενότητας 3.6 σας επιτρέπει να δείτε τα ανεπεξέργαστα δεδομένα που προκύπτουν από ένα πρόγραμμα το οποίο εκτελείται στο διακομιστή· το πρόγραμμα EchoServer σας επιτρέπει να δείτε τα ανεπεξέργαστα δεδομένα που μεταδίδονται από τη φόρμα του πελάτη.

Αν εκκινήσετε το πρόγραμμα EchoServer, για παράδειγμα στη θύρα 8088 του τοπικού σας μηχανήματος, και μετά αλλάξετε τις φόρμες ώστε να καθορίζουν τη διεύθυνση προορισμού <http://localhost:8088/>, τότε θα μπορείτε να βλέπετε αν τα δεδομένα που συλλέγονται έχουν τη μορφή που περιμένετε. Εκτός από την προβολή των δεδομένων της φόρμας, ο EchoServer σας δείχνει επίσης τις κεφαλίδες αιτήσεων HTTP που στέλνει ο φυλλομετρητής.

EchoServer

Η Λίστα 19.16 παρουσιάζει τον κώδικα για το διακομιστή, στο ανώτατο επίπεδο. Συνήθως θα εκτελείτε το πρόγραμμα EchoServer από τη γραμμή διαταγών, καθορίζοντας τη θύρα που θα πρέπει να παρακολουθεί ή αφήνοντας την προεπιλεγμένη θύρα 8088. Το πρόγραμμα EchoServer δέχεται στη συνέχεια τις αιτήσεις HTTP από τους πελάτες, συσκευάζει όλα τα δεδομένα HTTP που του στέλνονται σε μια ιστοσελίδα, και επιστρέφει τη σελίδα αυτή στον πελάτη. Στις περισσότερες περιπτώσεις ο διακομιστής διαβάζει μέχρι να συναντήσει μια κενή γραμμή, που υποδεικνύει το τέλος της GET, της HEAD, ή των περισσότερων τύπων αιτήσεων HTTP. Στην περίπτωση της POST, όμως, ο διακομιστής ελέγχει την κεφαλίδα αίτησης Content-Length και διαβάζει τον αντίστοιχο αριθμό byte μετά την κενή γραμμή.

Οι Λίστες 19.17 και 19.18 παρουσιάζουν μερικές βοηθητικές κλάσεις που απλοποιούν τη διτύωση. Το πρόγραμμα EchoServer έχει δομηθεί πάνω σε αυτές.

Λίστα 19.16 EchoServer.java

```
import java.net.*;
import java.io.*;
import java.util.*;
```

```
/** Ένας απλός διακομιστής HTTP που παράγει μια ιστοσελίδα η οποία δείχνει
 * όλα τα δεδομένα που παρέλαβε από τον πελάτη Ιστού (συνήθως ένα
 * φυλλομετρητή). Για να χρησιμοποιήσετε το διακομιστή, ξεκινήστε τον στο
 * σύστημα της επιλογής σας, καθορίζοντας έναν αριθμό θύρας αν θέλετε να
 * διαφέρει από τη θύρα 8088. Ονομάστε αυτό το σύστημα server.com. Μετά
 * ξεκινήστε ένα φυλλομετρητή Ιστού στο ίδιο ή σε διαφορετικό σύστημα,
 * και συνδεθείτε με τη διεύθυνση http://server.com:8088/κάτι.Η
 * ιστοσελίδα που θα προκύψει δείχνει τα δεδομένα που έστειλε ο
 * φυλλομετρητής σας. Για αποσφαλμάτωση μικροϋπηρεσιών ή κάποιου
 * άλλου προγράμματος στο διακομιστή, καθορίστε τη διεύθυνση
 * http://server:8088/κάτι στην ιδιότητα ACTION της φόρμας HTML.
 * Μπορείτε να στείλετε δεδομένα είτε με τη μέθοδο GET είτε με τη μέθοδο
```

Αλγόριθμος 19-16 EchoServer.java (συνέχεια)

```

* POST* και στις δύο περιπτώσεις, η σελίδα που προκύπτει θα δείχνει
* όλα όσα έστειλε ο φυλλομετρητής.
*/

public class EchoServer extends NetworkServer {
    protected int maxRequestLines = 50;
    protected String serverName = "EchoServer";

    /** Αριθμός θύρας ως όρισμα της γραμμής διαταγών.
    * Διαφορετικά χρησιμοποιείται η θύρα 8088.
    */
    public static void main(String[] args) {
        int port = 8088;
        if (args.length > 0) {
            try {
                port = Integer.parseInt(args[0]);
            } catch (NumberFormatException nfe) {}
        }
        new EchoServer(port, 0);
    }

    public EchoServer(int port, int maxConnections) {
        super(port, maxConnections);
        listen();
    }

    /** Υποσκελίζει τη μέθοδο handleConnection της κλάσης NetworkServer
    * ώστε να διαβάζει κάθε γραμμή δεδομένων που λαμβάνει, να την
    * αποθηκεύει σε έναν πίνακα αλφαριθμητικών, και να τη στέλνει πίσω
    * ενσωματωμένη σε ένα στοιχείο PRE σε μια σελίδα HTML.
    */
    public void handleConnection(Socket server)
        throws IOException {
        System.out.println
            (serverName + ": got connection from " +
             server.getInetAddress().getHostName());
        BufferedReader in = SocketUtil.getReader(server);
        PrintWriter out = SocketUtil.getWriter(server);
        String[] inputLines = new String[maxRequestLines];
        int i;
        for (i=0; i<maxRequestLines; i++) {
            inputLines[i] = in.readLine();
            if (inputLines[i] == null) // Ο πελάτης έκλεισε τη σύνδεση.
                break;
            if (inputLines[i].length() == 0) { // Κενή γραμμή.
                if (usingPost(inputLines)) {
                    readPostData(inputLines, i, in);
                    i = i + 2;
                }
            }
        }
    }

```

Αλγόριθμος 19-16 EchoServer.java (συνέχεια)

```

        }
        break;
    }
    printHeader(out);
    for (int j=0; j<i; j++) {
        out.println(inputLines[j]);
    }
    printTrailer(out);
    server.close();
}

// Αποστολή τυπικής απάντησης HTTP και της αρχής μιας τυπικής ιστοσελίδας.
// Χρήση του HTTP 1.0 για συμβατότητα με όλους τους πελάτες.

private void printHeader(PrintWriter out) {
    out.println
        ("HTTP/1.0 200 OK\r\n" +
         "Server: " + serverName + "\r\n" +
         "Content-Type: text/html\r\n" +
         "\r\n" +
         "<!DOCTYPE HTML PUBLIC " +
         "\"-//W3C//DTD HTML 4.0 Transitional//EN\">\r\n" +
         "<HTML>\r\n" +
         "<HEAD>\r\n" +
         "  <TITLE>" + serverName + " Results</TITLE>\r\n" +
         "</HEAD>\r\n" +
         "\r\n" +
         "<BODY BGCOLOR=\"#FDF5E6\">\r\n" +
         "  <H1 ALIGN=\"CENTER\">" + serverName +
         "  Results</H1>\r\n" +
         "  Here is the request line and request headers\r\n" +
         "  sent by your browser:\r\n" +
         "  <PRE>");
}

// Εκτύπωση του τέλους μιας τυπικής ιστοσελίδας.

private void printTrailer(PrintWriter out) {
    out.println
        ("</PRE>\r\n" +
         "</BODY>\r\n" +
         "</HTML>\r\n");
}

// Οι κανονικές αιτήσεις ιστοσελίδων χρησιμοποιούν την GET, οπότε αυτές
// ο διακομιστής μπορεί απλώς να διαβάζει μία γραμμή κάθε φορά. Όμως οι
// φόρμες HTML μπορεί να χρησιμοποιούν τη μέθοδο POST, οπότε θα πρέπει
// να προσδιορίσουμε τον αριθμό των byte της POST που έχουν σταλεί,

```

Λίστα 19.16 EchoServer.java (συνέχεια)

```
// ώστε να γνωρίζουμε τα επιπλέον δεδομένα που πρέπει να διαβαστούν
// μετά από τις τυπικές κεφαλίδες HTTP.
private boolean usingPost(String[] inputs) {
    return(inputs[0].toUpperCase().startsWith("POST"));
}

private void readPostData(String[] inputs, int i,
    BufferedReader in)
    throws IOException {
    int contentLength = contentLength(inputs);
    char[] postData = new char[contentLength];
    int read(postData, 0, contentLength);
    inputs[++i] = new String(postData, 0, contentLength);
}

// Με δεδομένη μια γραμμή που αρχίζει με το κείμενο Content-Length,
// επιστρέφει την ακέραια τιμή που καθορίζεται.
private int contentLength(String[] inputs) {
    String input;
    for (int i=0; i<inputs.length; i++) {
        if (inputs[i].length() == 0)
            break;
        input = inputs[i].toUpperCase();
        if (input.startsWith("CONTENT-LENGTH"))
            return (getLength(input));
    }
    return (0);
}

private int getLength(String length) {
    StringTokenizer tok = new StringTokenizer(length);
    tok.nextToken();
    return (Integer.parseInt(tok.nextToken()));
}
}
```

Λίστα 19.17 NetworkServer.java

```
import java.net.*;
import java.io.*;

/** Ένα σημείο εκκίνησης για διακομιστές δικτύου. Θα πρέπει να υποσκελιστεί
 * η handleConnection, αλλά σε πολλές περιπτώσεις η listen μπορεί να
 * μετρίκει αποτελεσματικά. Η κλάση NetworkServer χρησιμοποιεί την SocketUtil
 * για απλοποίηση της δημιουργίας των PrintWriter και BufferedReader.
 */
```

Λίστα 19.17 NetworkServer.java (συνέχεια)

```
public class NetworkServer {
    private int port, maxConnections;

    /** Δόμηση διακομιστή στη καθορισμένη θύρα. Θα συνεχίσει να δέχεται
     * συνδέσεις, μεταβιβάζοντάς τις στη μέθοδο handleConnection μέχρι να
     * σταλεί ρητή εντολή εξόδου (π.χ., System.exit) ή να δημιουργηθεί ο
     * μέγιστος αριθμός συνδέσεων. Αν θέλετε να εκτελείται ο διακομιστής
     * συνεχώς, καθορίστε τιμή 0 για την maxConnections.
     */

    public NetworkServer(int port, int maxConnections) {
        setPort(port);
        setMaxConnections(maxConnections);
    }

    /** Παρακολούθηση μιας θύρας για συνδέσεις. Κάθε φορά που δημιουργείται
     * μια σύνδεση, μεταβίβαση της υποδοχής (Socket) που προκύπτει
     * στην handleConnection.
     */

    public void listen() {
        int i=0;
        try {
            ServerSocket listener = new ServerSocket(port);
            Socket server;
            while((i++ < maxConnections) || (maxConnections == 0)) {
                server = listener.accept();
                handleConnection(server);
            }
        } catch (IOException ioe) {
            System.out.println("IOException: " + ioe);
            ioe.printStackTrace();
        }
    }

    /** Αυτή είναι η μέθοδος που παρέχει τη συμπεριφορά στο διακομιστή,
     * αφού καθορίσει τι θα γίνει με την υποδοχή (socket) που προκύπτει.
     * <B>Υποσκελίστε αυτή τη μέθοδο στους διακομιστές που αναπτύσσετε
     * <P>
     * <P>
     * Αυτή η γενική εκδοχή αναφέρει απλώς τον υπολογιστή υπηρεσίας που
     * έκανε τη σύνδεση, δείχνει την πρώτη γραμμή που έστειλε ο πελάτης,
     * και στέλνει μια γραμμή ως απάντηση.
     */

    protected void handleConnection(Socket server)
        throws IOException {
        BufferedReader in = SocketUtil.getReader(server);
        PrintWriter out = SocketUtil.getWriter(server);
        System.out.println

```

Λίστα 19.17 NetworkServer.java (συνέχεια)

```

("Generic Network Server: got connection from " +
 server.getInetAddress().getHostName() + "\n" +
 "with first line '" + in.readLine() + "'");
out.println("Generic Network Server");
server.close();
}

/** Λήψη του μέγιστου αριθμού συνδέσεων που θα χειριστεί ο διακομιστής
 * πριν τερματιστεί. Η τιμή 0 καθορίζει ότι ο διακομιστής θα πρέπει να
 * εκτελείται μέχρι να τερματιστεί ρητά.
 */
public int getMaxConnections() {
    return (maxConnections);
}

/** Καθορισμός του μέγιστου αριθμού συνδέσεων. Η τιμή 0 καθορίζει ότι ο
 * διακομιστής θα πρέπει να εκτελείται συνεχώς (μέχρι να τερματιστεί
 * ρητά).
 */
public void setMaxConnections(int maxConnections) {
    this.maxConnections = maxConnections;
}

/** Λήψη της θύρας που παρακολουθεί ο διακομιστής. */
public int getPort() {
    return (port);
}

/** Καθορισμός της θύρας. <B>Μπορείτε να το κάνετε μόνο πριν κληθεί η
 * "connect".</B> Αυτό συνήθως γίνεται στη συνάρτηση δόμησης.
 */
protected void setPort(int port) {
    this.port = port;
}
}

```

Λίστα 19.18 SocketUtil.java

```

import java.net.*;
import java.io.*;

/** Ένας συντομευμένος τρόπος δημιουργίας των BufferedReader και
 * PrintWriter που σχετίζονται με μια υποδοχή.
 */

```

Λίστα 19.18 SocketUtil.java (συνέχεια)

```

public class SocketUtil {
    /** Δημιουργία ενός BufferedReader για τη λήψη εισερχόμενων δεδομένων.
    */

    public static BufferedReader getReader(Socket s)
        throws IOException {
        return (new BufferedReader(
            new InputStreamReader(s.getInputStream())));
    }

    /** Δημιουργία ενός αντικειμένου PrintWriter για την αποστολή
    * εξερχόμενων δεδομένων. Αυτό το αντικείμενο PrintWriter θα αδειάσει
    * αυτόματα το ρεύμα όταν κληθεί η println.
    */

    public static PrintWriter getWriter(Socket s)
        throws IOException {
        // Η τιμή true στο δεύτερο όρισμα σημαίνει autoflush
        // (αυτόματο άδειασμα).
        return (new PrintWriter(s.getOutputStream(), true));
    }
}

```

