

Παρακάτω περιγράφονται τα βήματα για την μεταγλώττιση και εκτέλεση προγραμμάτων Java με υποδοχές:

1. **Εκκίνηση κελύφους Linux:** Πρώτα πρέπει να έχουμε ανοιχτό ένα παράθυρο κελύφους Linux ώστε να εισάγουμε εντολές.
2. **Έλεγχος εντολών Java:** Έπειτα ελέγχουμε αν λειτουργούν ο μεταγλωττιστής και ο διερμηνευτής της Java μετά την εγκατάσταση της. Γι αυτό εισάγουμε τις παρακάτω εντολές στο κέλυφος Linux:

```
$ which javac
$ which java
```

Θα πρέπει να εμφανίζονται στην οθόνη οι πλήρεις διαδρομές.

3. **Δημιουργία ενός καταλόγου:** Δημιουργούμε ένα κατάλογο ώστε να αποθηκεύσουμε τα παραδείγματα των προγραμμάτων μας. Έτσι, δημιουργούμε ένα κατάλογο με όνομα `javasockets_lab` και μπαίνουμε σε αυτό τον κατάλογο. Οι εντολές είναι οι εξής:

```
$ mkdir javasockets_lab
$ cd javasockets_lab
```

4. **Ανάπτυξη προγράμματος πελάτη:** Εισάγουμε την παρακάτω κλάση `SimpleClient.java` που περιέχει το κώδικα πελάτη σε ένα απλό επεξεργαστή κειμένου (όπως το `kedit`):

```
import java.net.*;
import java.io.*;
public class SimpleClient {
    private static final int PORT = 1234;
    public static void main(String args[]) throws IOException {
        // Bhma 1o: Dhmiourgia ypodoxhs reymatos kai syndesi me to
        // diakomisth sthn thyra PORT
        Socket dataSocket = new Socket(InetAddress.getLocalHost(),PORT);
        System.out.println("Connection established");
        // Bhma 2o: Dhmiourgia reymatos eiserxomenon dedomenon ths
        // ypodoxhs dataSocket
        InputStream is = dataSocket.getInputStream();
        BufferedReader in = new BufferedReader(new InputStreamReader(is));
        // Bhma 2o: Dhmiourgia reymatos exerxomenon dedomenon ths ypodoxhs
        // dataSocket
        OutputStream os = dataSocket.getOutputStream();
        PrintWriter out = new PrintWriter(os, true);
        // Bhma 3o: Eggrafh mhnymatos Hello from Client! sto diakomisth
        out.println("Hello from Client!");
        System.out.println("Message sent: Hello from Client!");
        // Bhma 3o: Anagnosi mhnymatos apo to diakomisth
        String line = in.readLine();
        System.out.println("Received message from server: " + line);
        // Bhma 4o: Kleisimo ypodoxhs
        dataSocket.close();
        System.out.println("Data socket closed");
    }
}
```

Αφού εισάγουμε το παραπάνω πρόγραμμα, το αποθηκεύουμε σε ένα αρχείο με όνομα `SimpleClient.java` κάτω από το κατάλογο `javasockets_lab`. Εναλλακτικά μπορούμε να μεταφορτώσουμε το πρόγραμμα Java [SimpleClient.java](#) στο κατάλογο `javasockets_lab`.

5. **Ανάπτυξη προγράμματος διακομιστή:** Εισάγουμε την παρακάτω κλάση SimpleServer.java που περιέχει το κώδικα διακομιστή σε ένα επεξεργαστή κειμένου:

```
import java.net.*;
import java.io.*;
public class SimpleServer {
    private static final int PORT = 1234;
    public static void main(String args[]) throws IOException {
        // Bhma 1o: Dhmioyrgia ypodoxhs reymatos ServerSocket sthn thyra
        // PORT
        ServerSocket connectionSocket = new ServerSocket(PORT);
        System.out.println("Server started");
        // Bhma 2o: Anamoni kai apodoxh aithshs syndeshs apo pelath
        Socket dataSocket = connectionSocket.accept();
        System.out.println("Received request from " +
            connectionSocket.getInetAddress());
        // Bhma 3o: Dhmioyrgia reymatos eiserxomenon dedomenon ths ypodoxhs
        // dataSocket
        InputStream is = dataSocket.getInputStream();
        BufferedReader in = new BufferedReader(new InputStreamReader(is));
        // Bhma 3o: Dhmioyrgia reymatos exerxomenon dedomenon ths ypodoxhs
        // dataSocket
        OutputStream os = dataSocket.getOutputStream();
        PrintWriter out = new PrintWriter(os, true);
        // Bhma 4o: Anagnosi mhnymatos apo to pelath
        String line = in.readLine();
        System.out.println("Received message from client: " + line);
        // Bhma 4o: Eggrafi mhnymatos Good Bye! sto pelath
        out.println("Good Bye!");
        System.out.println("Message sent: Good Bye!");
        // Bhma 5o: Kleisimo ths syndeshs pelaths
        dataSocket.close();
        System.out.println("Data socket closed");
        // Kleisimo ths ypodoxhs diakomisth
        connectionSocket.close();
        System.out.println("Connection socket closed");
    }
}
```

Αφού εισάγουμε το παραπάνω πρόγραμμα, το αποθηκεύουμε σε ένα αρχείο με όνομα SimpleServer.java κάτω από το κατάλογο javasockets_lab. Εναλλακτικά μπορούμε να μεταφορτώσουμε το πρόγραμμα [Java SimpleServer.java](#) στο κατάλογο javasockets_lab.

6. **Μεταγλώττιση Java:** Για να μεταγλωττίσουμε τις κλάσεις SimpleClient.java και SimpleServer.java σε εκτελέσιμα αρχεία πληκρολογούμε τις παρακάτω εντολές στο κέλυφος ως εξής:

```
$ javac SimpleClient.java
$ javac SimpleServer.java
```

Έπειτα εισάγουμε την εντολή ls για να δούμε τα δύο αρχεία class που δημιούργησε ο μεταγλωττιστής στον τρέχοντα μας κατάλογο.

7. **Εκτέλεση:** Για να εκτελέσουμε τα δύο προγράμματα στο τοπικό μας υπολογιστή κάνουμε τις παρακάτω ενέργειες. Στο ένα παράθυρο κελύφους εισάγουμε την παρακάτω εντολή ως εξής:

```
$ java SimpleServer (δεν χρειάζεται να γράψουμε την επέκταση .class)
```

Το παραπάνω πρόγραμμα θα περιμένει την αίτηση του προγράμματος

πελάτη. Στην συνέχεια, ανοίγουμε ένα δεύτερο παράθυρο κελύφους και μπαίνουμε στο κατάλογο όπου βρίσκεται τα εκτελέσιμα προγράμματα (π.χ. `javasockets_lab`). Έπειτα στο δεύτερο παράθυρο εισάγουμε την παρακάτω εντολή ως εξής:

```
$ java SimpleClient (δεν χρειάζεται να γράψουμε την επέκταση .class)
```

Τέλος, πρέπει να έχουμε τις παρακάτω εξόδους στην οθόνη μας. Συγκεκριμένα, στο παράθυρο που τρέχει το πρόγραμμα διακομιστή πρέπει να έχει την παρακάτω έξοδο:

```
Server started  
Received request from  
Received message from client: Hello from Client!  
Message sent: Good Bye!  
Data socket closed  
Connection socket closed
```

Τέλος, στο παράθυρο που τρέχει το πρόγραμμα πελάτη πρέπει να έχει την παρακάτω έξοδο:

```
Connection established  
Message sent: Hello from Client!  
Received message from server: Good Bye!  
Data socket closed
```