

# Networking: Solutions

# 27

*If the presence of electricity can be made visible in any part of a circuit, I see no reason why intelligence may not be transmitted instantaneously by electricity.*

—Samuel F. B. Morse

*Protocol is everything.*

—Francois Giuliani

*What networks of railroads, highways and canals were in another age, the networks of telecommunications, information and computerization ... are today.*

—Bruno Kreisky

*The port is near, the bells I hear, the people all exulting.*

—Walt Whitman

## Objectives

In this chapter you'll learn:

- Java networking with URLs, sockets and datagrams.
- To implement Java networking applications by using sockets and datagrams.
- To implement Java clients and servers that communicate with one another.
- To implement network-based collaborative applications.
- To construct a simple multithreaded server.

## Self-Review Exercises

**27.1** Fill in the blanks in each of the following statements:

- a) Exception \_\_\_\_\_ occurs when an input/output error occurs when closing a socket.  
ANS: IOException.
- b) Exception \_\_\_\_\_ occurs when a hostname indicated by a client cannot be resolved to an address.  
ANS: UnknownHostException.
- c) If a DatagramSocket constructor fails to set up a DatagramSocket properly, an exception of type \_\_\_\_\_ occurs.  
ANS: SocketException.
- d) Many of Java's networking classes are contained in package \_\_\_\_\_.  
ANS: java.net.
- e) Class \_\_\_\_\_ binds the application to a port for datagram transmission.  
ANS: DatagramSocket.
- f) An object of class \_\_\_\_\_ contains an IP address.  
ANS: InetAddress.
- g) The two types of sockets we discussed in this chapter are \_\_\_\_\_ and \_\_\_\_\_.  
ANS: stream sockets, datagram sockets.
- h) The acronym URL stands for \_\_\_\_\_.  
ANS: Uniform Resource Locator.
- i) The acronym URI stands for \_\_\_\_\_.  
ANS: Uniform Resource Identifier.
- j) The key protocol that forms the basis of the World Wide Web is \_\_\_\_\_.  
ANS: HTTP.
- k) AppletContext method \_\_\_\_\_ receives a URL object as an argument and displays in a browser the World Wide Web resource associated with that URL.  
ANS: showDocument.
- l) Method getLocalHost returns a(n) \_\_\_\_\_ object containing the local IP address of the computer on which the program is executing.  
ANS: InetAddress.
- m) The URL constructor determines whether its String argument is a valid URL. If so, the URL object is initialized with that location. If not, a(n) \_\_\_\_\_ exception occurs.  
ANS: MalformedURLException.

**27.2** State whether each of the following is *true* or *false*. If *false*, explain why.

- a) UDP is a connection-oriented protocol.  
ANS: False; UDP is a connectionless protocol and TCP is a connection-oriented protocol.
- b) With stream sockets a process establishes a connection to another process.  
ANS: True.
- c) A server waits at a port for connections from a client.  
ANS: True.
- d) Datagram packet transmission over a network is reliable—packets are guaranteed to arrive in sequence.  
ANS: False; packets can be lost, arrive out of order or be duplicated.

## Exercises

*NOTE: Solutions to the programming exercises are located in the ch27solutions folder. Each exercise has its own folder named ex27\_## where ## is a two-digit number representing the exercise number. For example, exercise 27.13's solution is located in the folder ex26\_13.*

- 27.3** Distinguish between connection-oriented and connectionless network services.  
**ANS:** Connection-oriented services maintain a connection while data is being transferred. Connectionless services do not maintain a connection. Connection-oriented services are generally slower but more reliable.
- 27.4** How does a client determine the hostname of the client computer?  
**ANS:** `InetAddress.getLocalHost().getHostName()`.
- 27.5** Under what circumstances would a `SocketException` be thrown?  
**ANS:** If a socket operation fails due to a problem in the underlying protocol. For example, if a `DatagramSocket` cannot bind to the specified port.
- 27.6** How can a client get a line of text from a server?  
**ANS:** After connecting, the client can get data from the server using the stream object returned by the `Socket`'s `getInputStream` method. The code required to get a single line of text can be simplified by wrapping the `InputStream` in a `Scanner` or `ObjectInputStream`. Alternatively, the server can send the text by wrapping it in a `DatagramPacket` and using `send` method of class `DatagramSocket`.
- 27.7** Describe how a client connects to a server.  
**ANS:** A server listens for a connection on a specific address and port using the `ServerSocket` class. A client can then connect to the server by giving the address and port to the `Socket` constructor. Once the connect operation completes, the server and client can send data back and forth between each other.
- 27.8** Describe how a server sends data to a client.  
**ANS:** For stream-based sockets, the client connects to the server by creating a socket using the `Socket` class constructor. The name of the server and the port to connect to are passed to the `Socket` constructor. Information can be exchanged between the client and server using the socket's `InputStream` and `OutputStream`. For datagram-based sockets, the server can send a `DatagramPacket` to the client using the `send` method of class `DatagramSocket`.
- 27.9** Describe how to prepare a server to receive a stream-based connection request from a single client.  
**ANS:** First a `ServerSocket` object must be created and associated with a port on the server computer. If the `ServerSocket` is created properly, a call to the `accept` method can be issued on the `ServerSocket` object. This call will wait for a client to connect. When a client connects, a `Socket` object is returned from the `accept` call. The `Socket` object is used to get the `InputStream` and `OutputStream` objects for communication with the client.
- 27.10** How does a server listen for streams-based socket connections at a port?  
**ANS:** The `ServerSocket` `accept` method is used.
- 27.11** What determines how many connect requests from clients can wait in a queue to connect to a server?  
**ANS:** When the `ServerSocket` object is created on the server, the second argument to the `ServerSocket` constructor specifies the "queue length" (the number of clients that can wait to be processed by the server).
- 27.12** As described in the text, what reasons might cause a server to refuse a connection request from a client?  
**ANS:** A server usually has a capacity of the number of clients that can wait for a connection and be processed by the server. If the queue of clients is full, client connections are refused.

