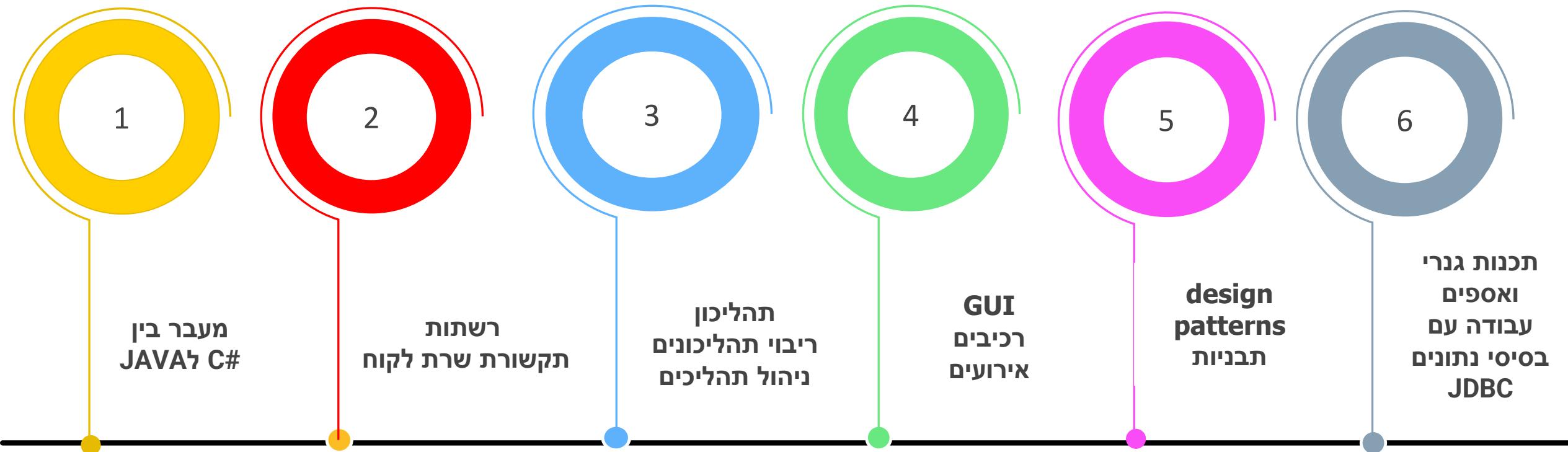


נושאים מתקדמים בתכנות מונחה עצמים ושפת JAVA

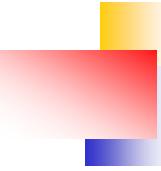
ד"ר רינה צbial-גירשין

יחידה 2

מפת הקורס



מבט על על הרצאה



- מושגים מתחום תקשורת
- מודל שירות לקוחות
- תקשורת TCP
- לקוחות
- שירות
- תקשורת UDP

תקשורת

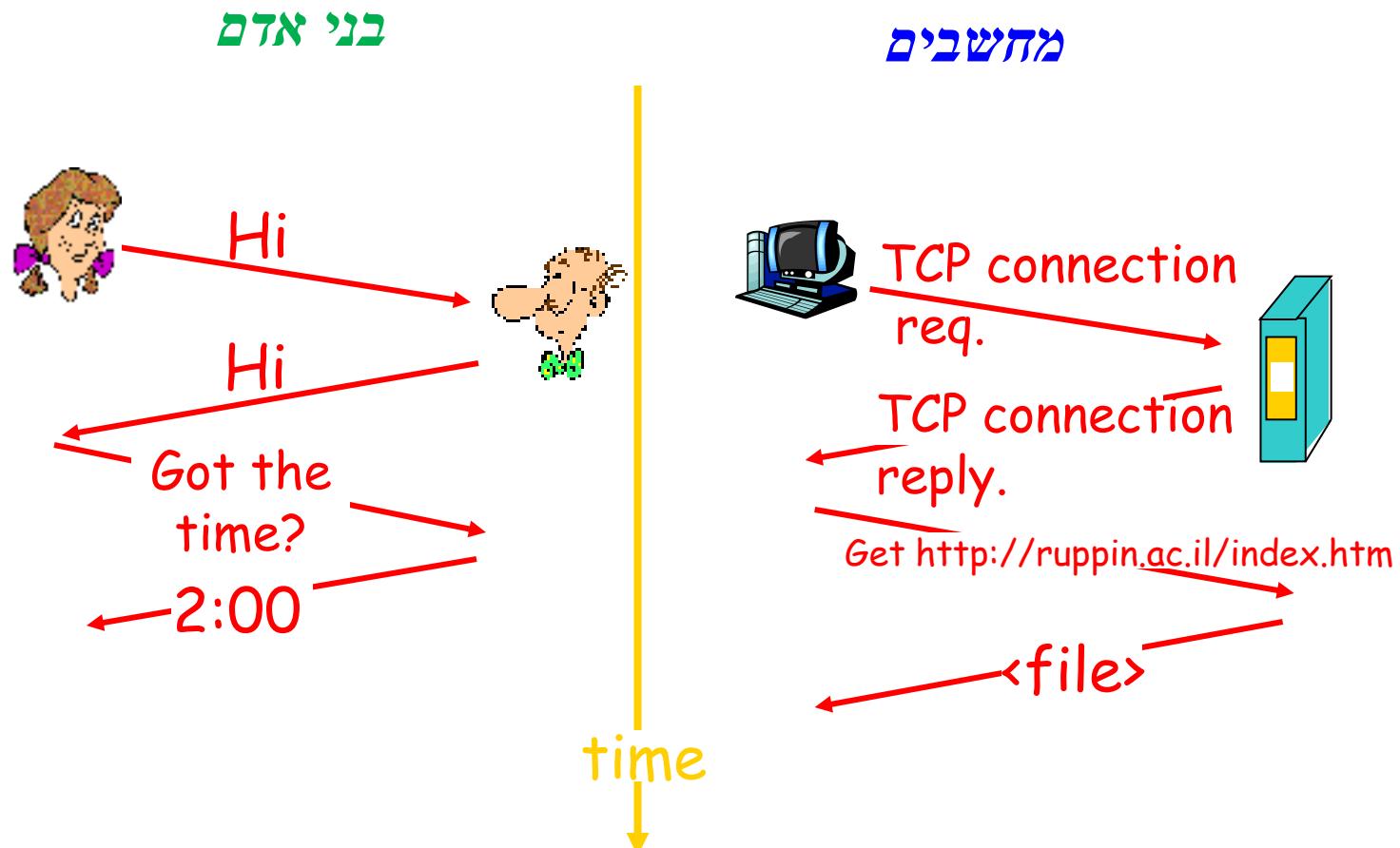


- **תקשורת** communication היא תהליך של העברת מידע בין שניים (או יותר) צדדים .
- כל אחד מהצדדים יכול להיות

 - שולח/ مصدر Sender או
 - מקבל/ קולט Receiver

- תהליך התקשורת כולל שליחת מסרים/הודעות, קבלה, הבנה ותגובה.

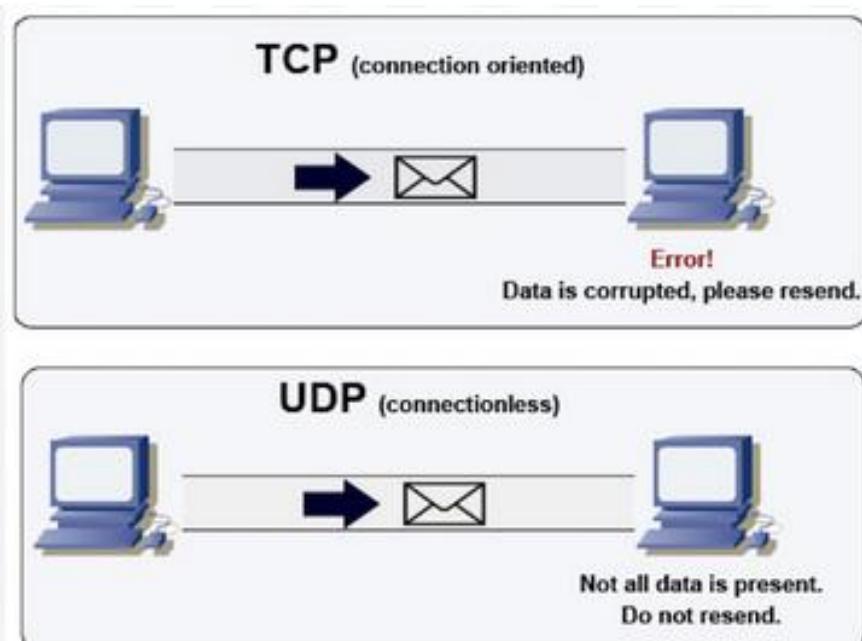
פרוטוקולי תקשורת



- **בערוצי תקשורת דיגיטליים** או מחשביים, התקשרות מתחכעת בין מחשבים או מכשירים אחרים, כאשר פרוטוקולים מסוימים מכתיבים את אופן העברת האימות והתגובה להודעות שנשלחות ביניהם.
- **פרוטוקול תקשורת – אוסף של כללים**

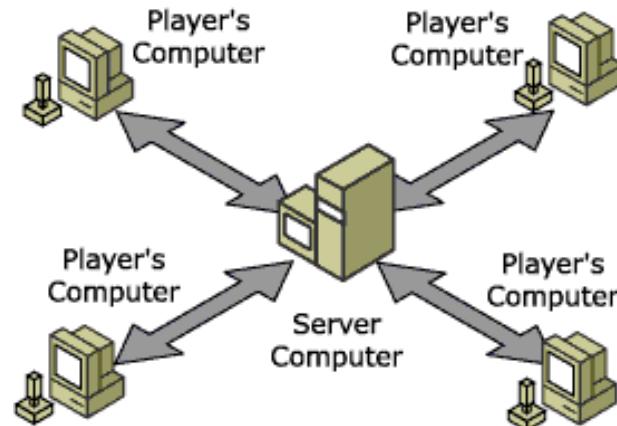
תקשרות אמינה או לא

- ישנם סוגי תקשורת איפה כל דבר שנשלח חייב להתקבל ללא שגיאות



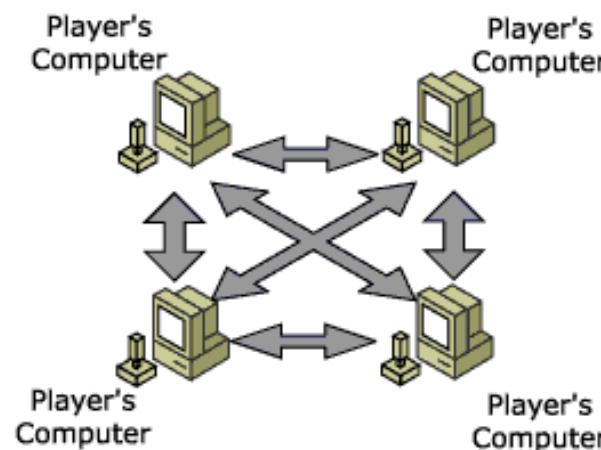
- חלק מן הישומים תקשורת יכולה להיות **לא אמינה**
- חלק מן הנתונים יכולים לאלט לאיבוד
- פרוטוקול UDP**
- בJAVA נשתמש במחלקות מוכנות שונות לתקשרות אמינה (*DatagramSocket*) ולא אמינה (*Socket*)

מודלי תקשורת



■ **שרות לקוחות client server**

- שרת מרכזי של קוחות פוניים אליו בבקשות
- שרת קבצים, שרת משחקים, שרת web ומו



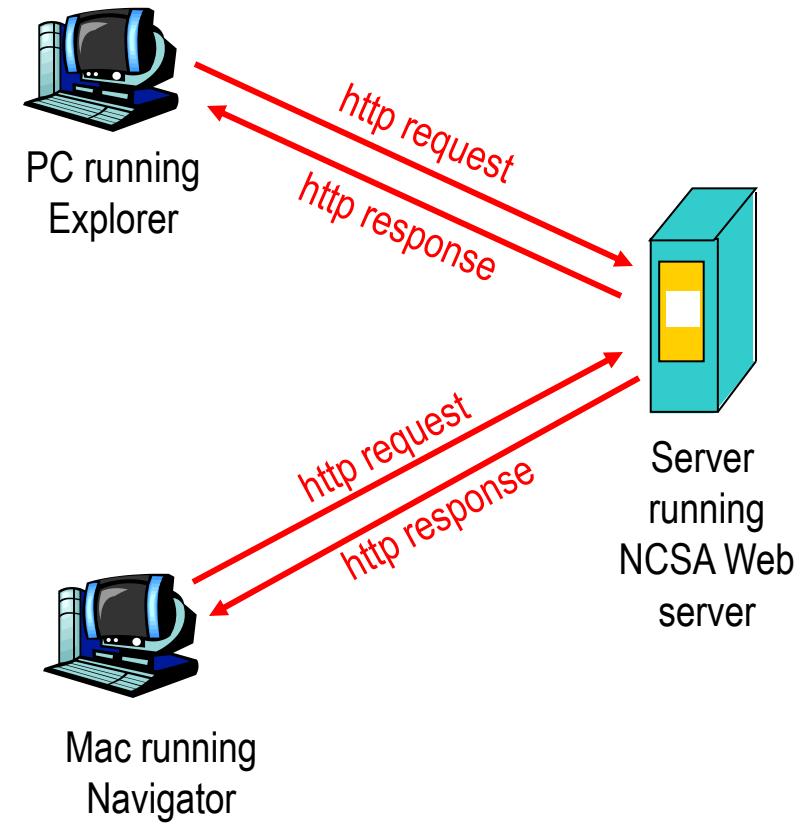
■ **חבר לחבר P2P**

- לקוחות מדברים אחד עם השני
- שניים יכולים להיות בו-זמנית גם שרת וגם לקוחות

The Web: the HTTP protocol

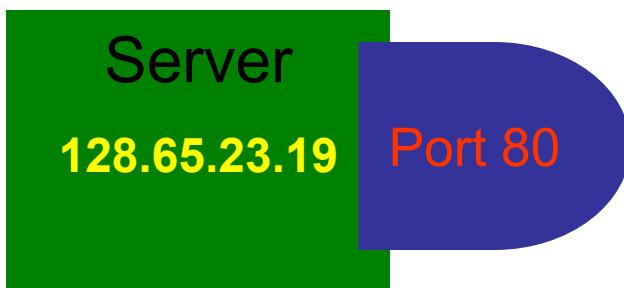
HTTP: hypertext transfer protocol

- Web's application layer protocol
- client/server model
 - *client*: browser that requests, receives, "displays" Web objects
 - *server*: Web server sends objects in response to requests



כתובת IP ופורט

- תקשורת בין מחשבים דורשת לדעת איפה מחשב נמצא ואיזה תחילה מעוניין אותו
- איפה מחשב נמצא - **כתובת IP**
- מקבלים מספק תקשורת/ספק אינטרנט
- כתובת IP גרסה 4
- 4 גושים בין 0-255
- 128.65.23.19
- איזה תחילה – **פורט**
- מע"פ OS מקצה פורטים לתקשורת
- במודל שרת ללקוח יש פорт קבוע לשרת ופורט שירותים ללקוח

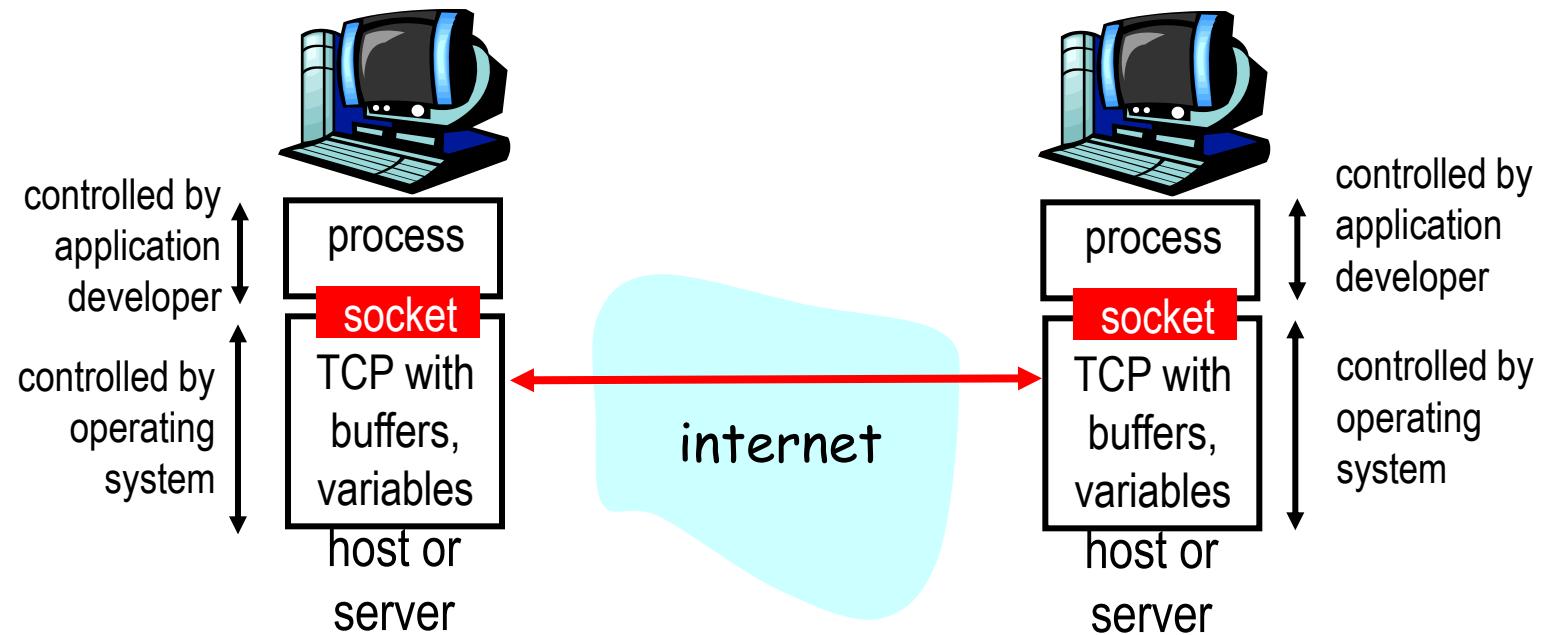


Socket-programming

Socket: a door between application process and end-end-transport protocol (UDP or TCP)

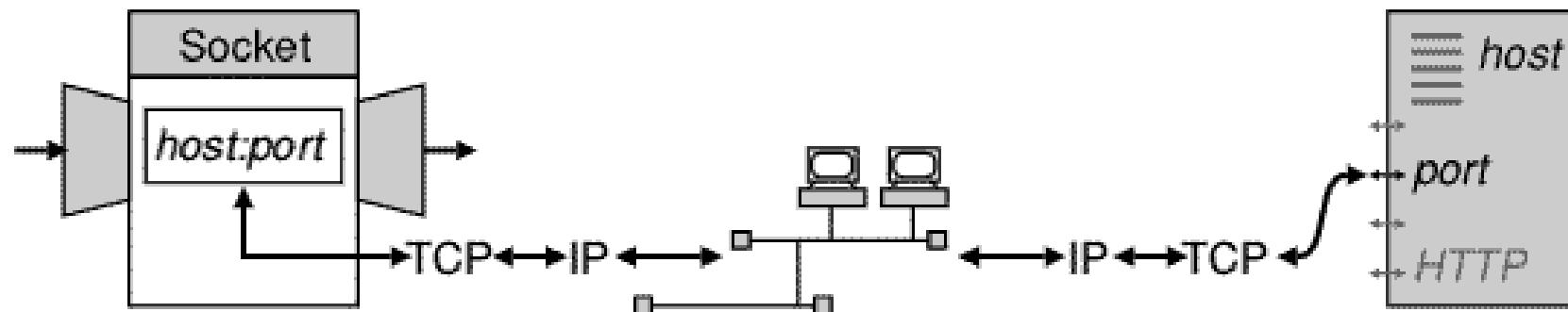
Port number: allows receiving host to determine to which local process the message should be delivered

Socket = IP address + port number



מה זה שקע socket

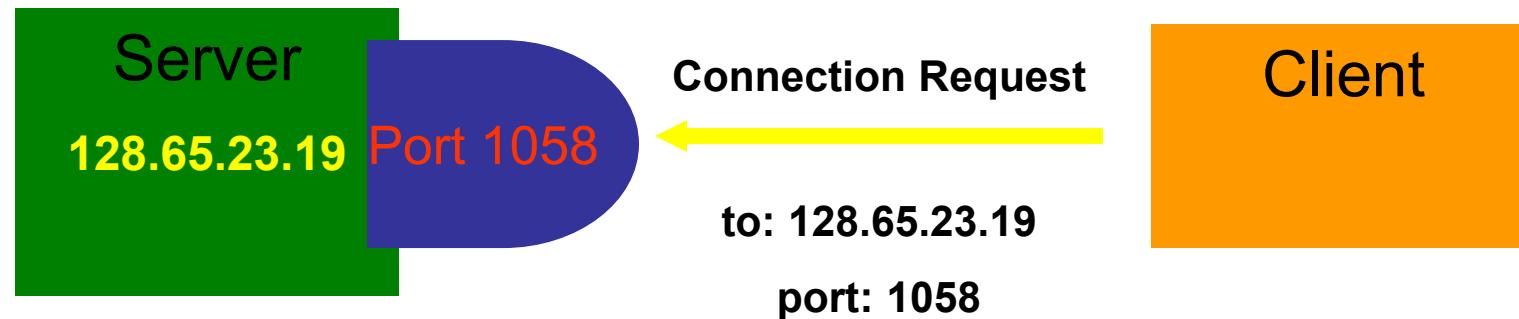
- **שקע Socket** הינו קצה אחד של קישור תקשורת דו-כיוונית בין שני תוכניות (ופורטים)
- נוצר בכל צד (בכל תוכנית) - גם לשרת וגם ללקוח וגם לPeer
- תקשורת מתבצעת על ידי משלוח הודעות משקע לשקע



מנגנון תקשורת

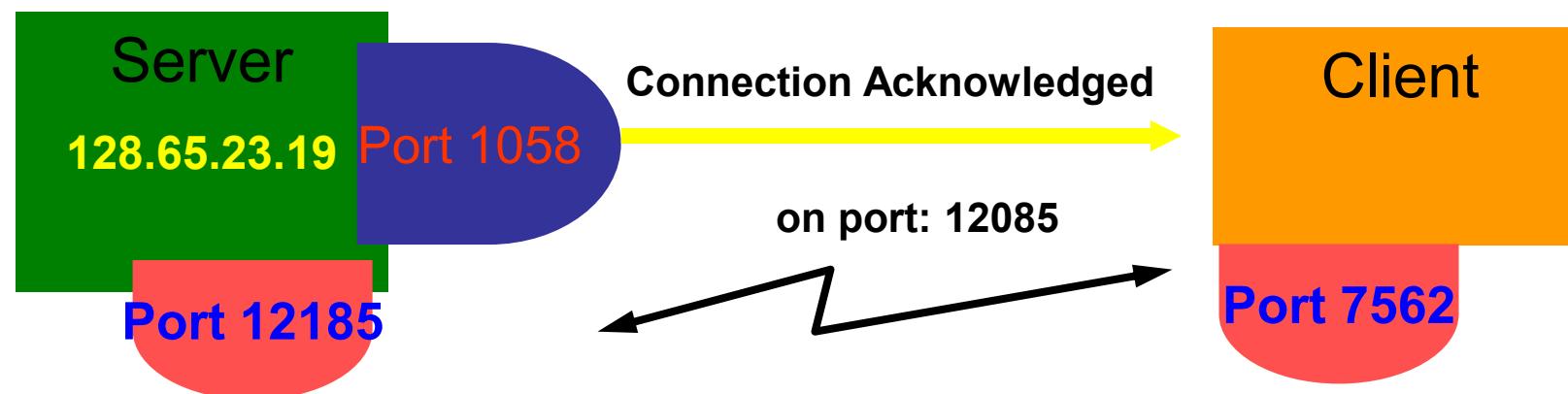


- לקוח Client פונה לשרת
- בקשה להתחבר
- בקשה לבצע שירות
- שרת Server מצוי בולאה אינסופית לפורט אליו מגיעות בקשות מלוקחות



מבנה תקשורת 2

- אם שרת Server מקבל את השיחה אז הוא (OS) מקצת פורט חדש לתקשורת עם לקוח ספציפי
- גם אצל לקוח נפתח פורט לתקשורת עם השרת



שרת ללקוח פשוט

```
import java.net.*
```

- נשתמש במחלקה `java.net.Socket` לייצירת השרת ו-`ServerSocket`
- יש לצרף חבילה `*.java.net.`
- שרת `java.net.ServerSocket server`
- מקבל בקשה מלוקוח להתחבר
- מייצר شكע לתקשורת
- לקוח `java.net.Socket client`
- פונה לשרת

client

מחלקה Socket

- מחלקה לייצירת שקע מסווג ללקוח **Socket**
- בנייתו מקבל כתובות IP או שם השירות ופורט שירות פתוח לתקשורת
- **public Socket(String host, int port) throws UnknownHostException, IOException**
- **מетодות חשובות**
- **public InputStream getInputStream() throws IOException**
 - מקבלת נתונים מן השירות
- **public OutputStream getOutputStream() throws IOException**
 - שולחת נתונים לשירות
- **public void close() throws IOException**
 - סוגרת את הקשר

מחלקה ServerSocket

- מחלקה לייצירת שקע מסוג `Server` שרת
- ממתינה לבקשת של לקוחות לפורט מסוים
- בנאי מקבל פורט

`public ServerSocket(int port) throws IOException`

- לאחר מכך מתודות חשובות

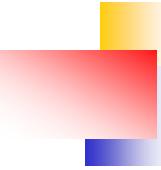
`public Socket accept() throws IOException`

- ממתינה לבקשת מהלכה, מייצרת חיבור ומחזירה שקע בצד שרת עבור לקוח ספציפי

`public void close() throws IOException`

- סוגרת את הקשר

תקשורת וIO



- זרימת נתונים יכולה להיות בתווים/אותיות או בBITS/ביתים
- Character streams: Readers and Writers
- Byte streams: InputStream and OutputStream
- לכן יש לצרף java.io

קלט בסיסי

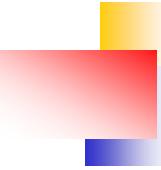
- הינה מחלקה אבסטרקטית **InputStream** ■
 - **read()** int or bytes in the current machine's byte format
 - **available()** returns the number of bytes available for reading
 - **skip()** skips a number of bytes
- נשתמש במחלקות ספציפיות ■
 - **BufferedInputStream** ■
 - **FilterInputStream** ■
 - **DataInputStream** ■
 - **FileInputStream** ■

Basic Output



- **מחלקה אבסטרקטית דומה ל - OutputStream- InputStream**
- `write()` int or bytes to destination
- `flush()` buffered data
- **נשותש במחלקות ספציפיות**
- `BufferedOutputStream`
- `DataOutputStream`
- `ObjectOutputStream`
- `FilterOutputStream`
- `FileOutputStream`
- `PipeOutputStream`

דוגמה



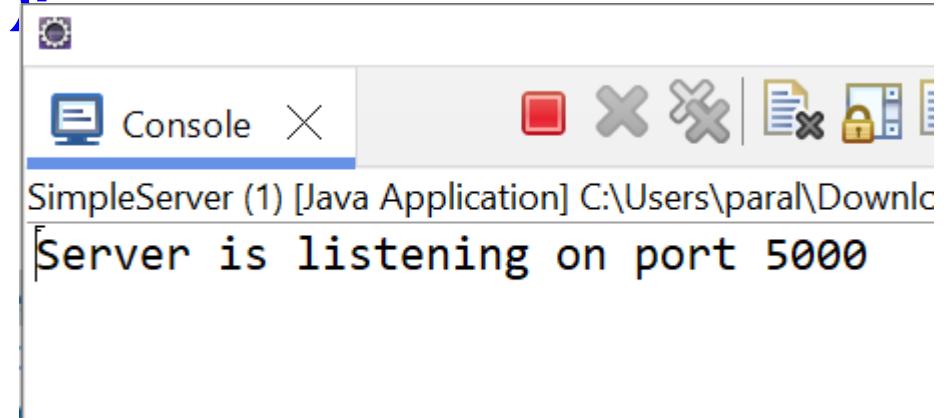
- נכתב שרת פשוט שמאזין לפורט 5000
 - נכתב לקוח שיתחבר לפורט 5000 בשרת
-
- **הרצאה:**
 - .1. קודם מרכיבים שרת
 - .2. רק אחר כך לקוח

Server

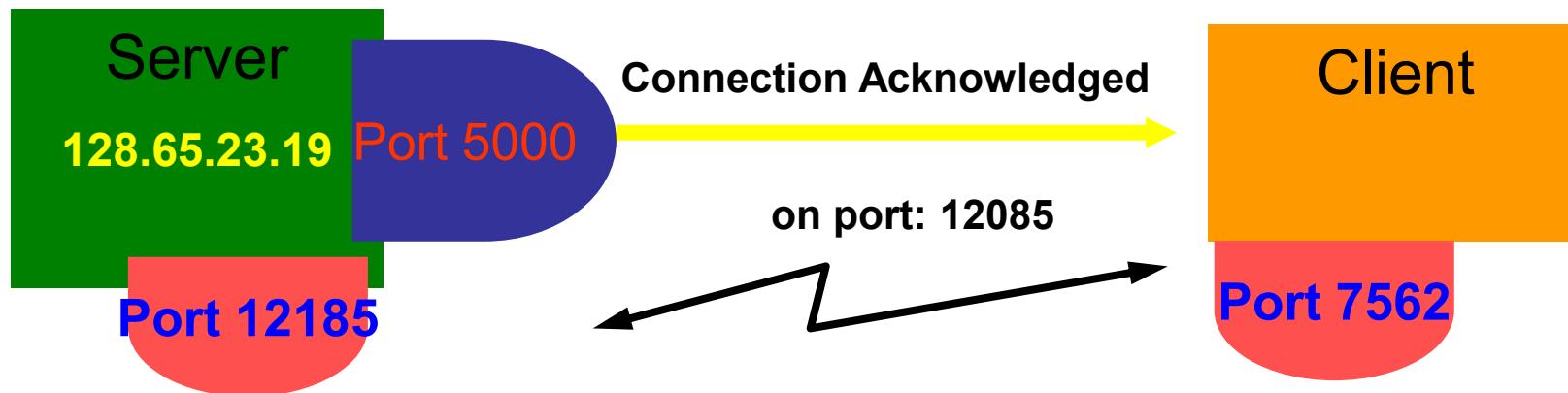
```
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;

public class SimpleServer
{
    public static void main(String[] args) throws IOException
    {
        try (ServerSocket serverSocket = new ServerSocket(5000))
        {
            System.out.println("Server is listening on port 5000");
            Socket socket = serverSocket.accept();
            System.out.println("New client connected");
        } catch (IOException e)
        {
            e.printStackTrace();
        }
    }
} //end of class
```

Port to listen



accept()



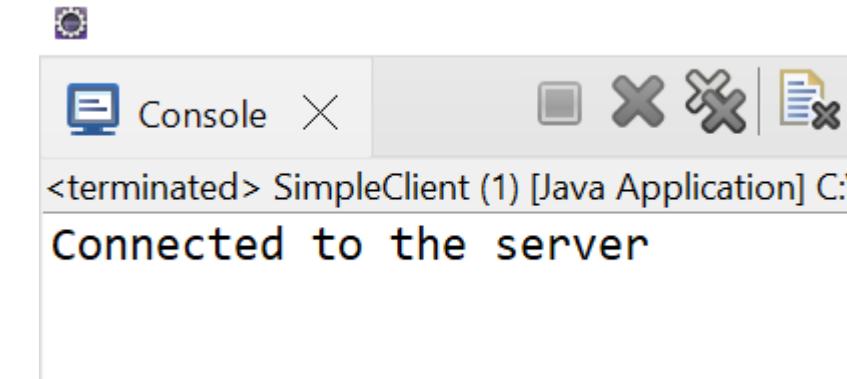
- שרת ממתיין לבקשת התחברות
- מקבל בקשה להתחבר `accept` ומיציר שקע לתקשורת
- Listens for a connection to be made to this socket and accepts it.
- בסוף תהליך תקשורת רצוי לסגור את הקשר

Client

```
import java.io.IOException;
import java.net.Socket;

public class SimpleClient
{
    public static void main(String[] args)
    {
        try (Socket socket = new Socket("localhost", 5000))
        {
            System.out.println("Connected to the server");
        } catch (IOException e)
        {
            e.printStackTrace();
        }
    }
}//end of class
```

Server IP address/name and port
127.0.0.1/localhost



הרצה

SimpleServer (1) [Java Application] C:\Users\paral\Downlc
Server is listening on port 5000

<terminated> SimpleClient (1) [Java Application] C:
Connected to the server

<terminated> SimpleServer (1) [Java Application] C:\Users\p
Server is listening on port 5000
New client connected

איך מרים גם שרת וגם לקוח?

```
t java.net.Socket;
c class SimpleClient
public static void main(String[] args)
    System.out.println("Connected to the
} catch (IOException e)
```

Console

1 SimpleServer [Java Application] C:\Users\paral\Download\SimpleServer\src\SimpleServer.java (id: 27772)
2 SimpleClient [Java Application] C:\Users\paral\Download\SimpleClient\src\SimpleClient.java (id: 28332)

משימה



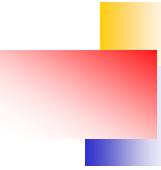
- יש לשכת קוד השירות כך שיירץ לולאה
- `While(true)`
- יש לקרוא למחלקה `SimpleServerLoop`

java.net



- חרגות חשובות
 - BindException
 - ConnectException
 - MalformedURLException
 - NoRouteToHostException
 - ProtocolException
 - SocketException
 - UnknownHostException
 - UnknownServiceException
- מחלקות חשובות
 - ServerSocket ■
 - Socket ■
 - InetAddress ■
 - DatagramPacket ■
 - DatagramSocket ■

דוגמה



- ניצור אפליקציה שרת ל��ח מסוג הד ECHO
- ל��ח יפנה לשרת לצורף התחברות
- כל מילה שלקוח שולח שרת ישלח בחזרה

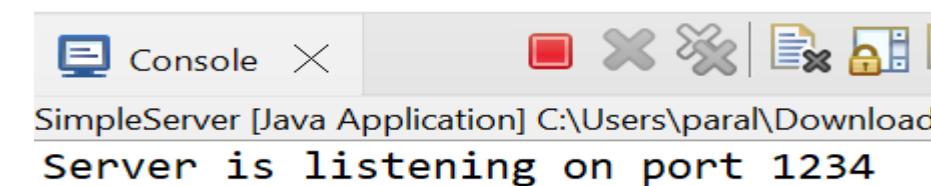
Echo Server

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

public class SimpleServer
{
    public static void main(String[] args) throws IOException
    {
        ServerSocket serverSocket = new ServerSocket(1234);
        System.out.println("Server is listening on port 1234");
        Socket clientSocket = serverSocket.accept();
        PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true);
        BufferedReader in =
            new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

        String inputLine;
        while ((inputLine = in.readLine()) != null)
        {
            System.out.println("Received: " + inputLine);
            out.println("Echo: " + inputLine);
        }

        in.close();    out.close();    clientSocket.close();    serverSocket.close();
    } //end of class
}
```



Echo Client

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

public class SimpleClient
{
    public static void main(String[] args) throws IOException
    {
        Socket socket = new Socket("127.0.0.1", 1234);
        PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
        BufferedReader in =
            new BufferedReader(new InputStreamReader(socket.getInputStream()));
        BufferedReader stdIn = new BufferedReader(new InputStreamReader(System.in));

        String userInput;
        while ((userInput = stdIn.readLine()) != null)
        {
            out.println(userInput);
            System.out.println("Server says you sent: " + in.readLine());
        }
        out.close();      in.close();      socket.close();
    } //end of class
}
```

eclipse-workspace - ClientServer/src/SimpleClientServer/SimpleServer.java - Eclipse IDE

File Edit Source Refactor Source Navigate Search Project Run Window Help



Console X

SimpleServer [Java Application] C:\Users\paral\Downloads\eclipse-java-2020-

Server is listening on port 1234
Received: hello server

Console X

SimpleServer [Java Application] C:\Users\paral\Downloads\eclipse-java-2020-

Server is listening on port 1234
Received: hello server
Received: thank you

Console X

SimpleClient [Java Application] C:\Users\paral\Downloads\eclipse-java-
hello server
Server says you sent: Echo: hello server
thank you

Console X

SimpleClient [Java Application] C:\Users\paral\Downloads\eclipse-java-2020-

hello server
Server says you sent: Echo: hello server
thank you
Server says you sent: Echo: thank you

הרצה

הרצה לאחר סגירת שרת

```
SimpleServer [Java Application] C:\Users\paral\Downloads\SimpleServer.jar  
Server is listening on port 1234  
Received: hello server  
Received: thank you
```

```
SimpleClient [Java Application] C:\Users\paral\Downloads\SimpleClient.jar  
hello server  
Server says you sent: Echo: hello server  
thank you  
Server says you sent: Echo: thank you  
what is your name  
Exception in thread "main" java.net.SocketException: Connection reset  
at java.base/sun.nio.ch.NioSocketImpl.implRead(NioSocketImpl.java:323)  
at java.base/sun.nio.ch.NioSocketImpl.read(NioSocketImpl.java:350)  
at java.base/sun.nio.ch.NioSocketImpl$1.read(NioSocketImpl.java:803)  
at java.base/java.net.Socket$SocketInputStream.read(Socket.java:981)  
at java.base/sun.nio.cs.StreamDecoder.readBytes(StreamDecoder.java:297)  
at java.base/sun.nio.cs.StreamDecoder.implRead(StreamDecoder.java:339)  
at java.base/sun.nio.cs.StreamDecoder.read(StreamDecoder.java:188)  
at java.base/java.io.InputStreamReader.read(InputStreamReader.java:181)  
at java.base/java.io.BufferedReader.fill(BufferedReader.java:161)
```

מגבלות לרגע זה



- כרגע שרתamazon ללקוח אחד.
 - לקוח שמתחבר "ראשון" חוסם תקשורת עם שאר הלקוחות.
 - בהמשך נרצה ששרת אחת "ישרת" הרבה לקוחות.
- multithreading ■

