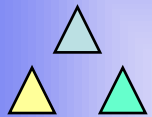


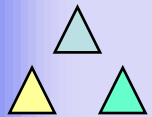
\mathcal{A}
 \mathcal{A}



UDP дейтаграма и сокети.

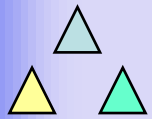
Аделина Алексиева

А
А



класове за UDP

- DatagramPacket
- DatagramSocket



Клас DatagramPacket

- `public final class DatagramPacket extends Object`
- Конструктори: различни конструктори за получаване и изпращане



Конструктори за получаване на дейтаграма

- `public DatagramPacket(byte[] buffer,
int length)`
- `public DatagramPacket(byte[] buffer,
int offset, int length) // Java 1.2`

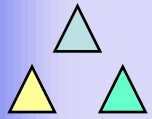
`length <= buffer.length - offset`

- `IllegalArgumentException`



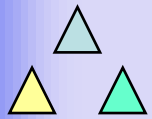
Конструктори за изпращане

- `public DatagramPacket(byte[] data, int length, InetAddress destination, int port)`
- `public DatagramPacket(byte[] data, int offset, int length, InetAddress destination, int port) // Java 1.2`
- `public DatagramPacket(byte[] data, int length, SocketAddress destination, int port) // Java 1.4`
- `public DatagramPacket(byte[] data, int offset, int length, SocketAddress destination, int port) // Java 1.4`



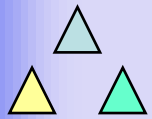
Get методи

- `public InetAddress getAddress()`
- `public int getPort()`
- `public SocketAddress getSocketAddress()`
// Java 1.4
- `public byte[] getData()`
- `public int getLength()`



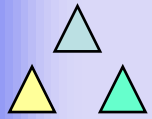
Set методи

- **public void setData(byte[] data)**
- **public void setData(byte[] data, int offset, int length)**
- **public void setAddress(InetAddress remote)**
- **public void setPort(int port)**
- **public void setLength(int length)**



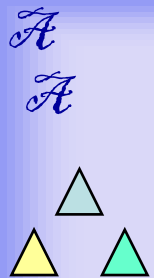
Клас DatagramSocket

- **public class DatagramSocket extends Object**
- Сокетите са свързани с локалния порт на който те слушат за получаване на данни и който е поместен в хедъра на дейтаграмата;
- При клиента само извиквате конструктор, който да позволи на системата да свържи към порт, който не се използва;
- При сървъра, клиента трябва да знае на кой порт сървъра слуша за получаване на дейтаграми. В противен случай, когато сървъра прави DatagramSocket, се указва и порт на който слуша.



Конструктори

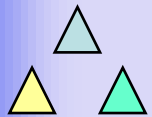
- `public DatagramSocket()` throws `SocketException`
- `public DatagramSocket(int port)` throws `SocketException`
- `public DatagramSocket(int port, InetAddress interface)` throws `SocketException`
- `public DatagramSocket(SocketAddress interface)` throws `SocketException` // Java 1.4
- `protected DatagramSocket(DatagramSocketImpl impl)` throws `SocketException` // Java 1.4



Изпращане и получаване на дейтаграми

- `public void send(DatagramPacket dp) throws IOException`
- `public void receive(DatagramPacket dp) throws IOException`
- `public void close()`
- `public int getLocalPort()`
- `public InetAddress getLocalAddress()`
- `public void connect(InetAddress host, int port)`
- `public void disconnect()`
- `public int getPort()`

А
А



ПРИМЕРЫ

- UDPPortScanner
- UDPSever
- UDPServer
- UDPDaytimeServer