

Gniazdka

Platformy Technologiczne

Michał Piotrowski
Michał Wójcik

Katedra Architektury Systemów Komputerowych
Wydział Elektroniki, Telekomunikacji i Informatyki
Politechnika Gdańska

4 marca 2011

TCP (Transmission Control Protocol):

- wiarygodny,
- połączeniowy,
- retransmisje,
- zachowana kolejność,
- potwierdzenia.

UDP (User Datagram Protocol):

- niewiarygodny,
- bezpołączeniowy,
- brak retransmisji,
- kolejność niezapewniona,
- brak potwierdzeń.

Klasy dotyczące połączeń:

- TCP (Transmission Control Protocol):
 - URL,
 - URLConnection,
 - Socket,
 - ServerSocket,
- UDP (User Datagram Protocol):
 - DatagramPacket,
 - DatagramSocket,
 - MulticastSocket.

URL (Uniform Resource Locator):

- nazwa hosta, nazwa pliku, numer portu, referencja,
- konstruktor:
 - `URL(String address)`,
 - `URL(URL baseUrl, String relativeURL)`,
 - `URL(String protocol, String host, int port, String file)`,
- jednorazowego zapisu,
- `MalformedURLException` — null lub nie znany protokół,
- `InputStream URL.openStream()` — czytanie bezpośrednio z URL,
- `URLConnection URL.openConnection()` — otwarcie połączenia:
 - `URLConnection.getInputStream()`,
 - `URLConnection.getOutputStream()`.

Gniazdko (Socket):

- punkt końcowy w dwukierunkowej komunikacji pomiędzy dwoma programami działającymi w sieci,
- przypisane do konkretnego portu,
- Socket — zapewnia jedno lub dwustronną komunikację,
- `ServerSocket` — gniazdko po stronie serwera, nasłuchiwanie połączeń,
- typowo tworzy i uruchamia się:
 - jedną aplikację serwera, który w oddzielnych wątkach obsługuje kolejnych klientów,
 - aplikacje klienckie, które nawiązują kontakt z serwerem.

- `Socket(String host, int port)`,
- `OutputStream Socket.getOutputStream()`,
- `InputStream Socket.getInputStream()`,
- `ServerSocket(int port)` — gniazdko nasłuchujące po stronie serwera,
- `Socket ServerSocket.accept()` — akceptowanie połączenia, metoda blokująca,
- `void ServerSocket.setSoTimeout(int timeout)` — timeout dla metody `accept()`.

Datagram:

- niezależna, samodzielna wiadomość przesłana przez sieć, której dostarczenie, czas dostarczenia i treść nie są gwarantowane,
- `DatagramSocket` — gniazdko datagramowe,
- `DatagramPacket` — pojedynczy pakiet,
- `MulticastSocket` — wysyłanie pakietu do kilku gniazdek na raz,
- `InetAddress` — reprezentacja adresu protokołu IP.

- `DatagramPacket(byte[] buff, int lenght, InetAddress address, int port)`,
- `DatagramSocket(int port)`,
- `void DatagramSocket.send(DatagramPacket packet)` — wysyłanie pakietu,
- `void DatagramSocket.receive(DatagramPacket packet)` — odbieranie pakietu,
- `void DatagramSocket.setSoTimeout(int timeout)` — timeout dla metody `receive()`.

Serwer:

- wysyłanie za pomocą `DatagramSocket`,
- `InetAddress.getByName("230.0.0.1")` — adres multicast.

Klient:

- `MulticastSocket(int port)`,
- `void MulticastSocket.joinGroup(InetAddress group)` — przyłączenie się do grupy,
- `void MulticastSocket.leaveGroup(InetAddress group)` — opuszczenie grupy,
- `void MulticastSocket.receive(DatagramPacket packet)` — odebranie pakietu.