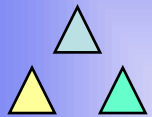
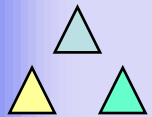


А
А



Класове за работа с URL и URI.

Аделина Алексиева

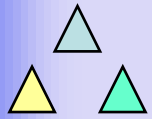


Uniform Resource Identifier (URI)

- Низ от символи със специфичен синтаксис, който идентифицира даден ресурс:

scheme:scheme-specific-part

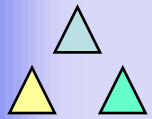
\mathcal{A}
 \mathcal{A}



Видове scheme

- *file*
- *ftp*
- *http*
- *gopher*
- *mailto*
- *news*
- *telnet*
- *urn*

А
А



scheme-specific-part

//authority/path?query

Пример:

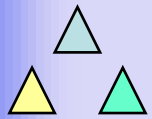
<http://www.powells.com/cgi-bin/biblio?inkey=62-1565928709-0>

Scheme – <http>

Източник - www.powells.com

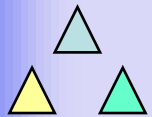
Път - [cgi-bin/biblio](http://www.powells.com/cgi-bin/biblio)

Заявка - [inkey=62-1565928709-0](http://www.powells.com/cgi-bin/biblio?inkey=62-1565928709-0)



Други примери

- ftp://mp3:mp3@ci43198-a.ashvil1.nc.home.com:33/VanHalen-Jump.mp3
- www.churchofsatan.com
- <http://www.amazon.com/exec/obidos/ISBN%3D1565924851/cafeaulaitA/002-3777605-3043449>



Видове URI

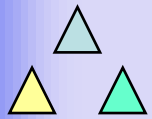
- Uniform Resource Locators (URLs)
- Uniform Resource Names (URNs):

urn:namespace:resource_name

Пример:

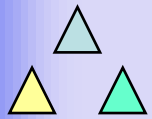
urn:ISBN:1565924851

\mathcal{A}
 \mathcal{A}



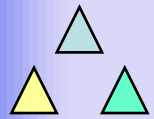
protocol://username@hostname:port/path/filename
?query#fragment

- protocol \equiv scheme
(file, ftp, http, https, gopher, news, telnet, wais)
- hostname – име на сървър



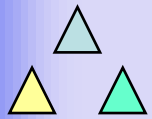
java.net.InetAddress

- На най-високо ниво представяне на IP адрес (IPv4 и IPv6);
- Използва се от класове като Socket, ServerSocket, DatagramSocket, DatagramPacket и др.;
- Включва работата с hostname и IP адреса



Методи

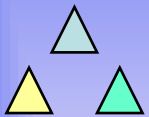
- public static InetAddress **getByName**(String hostName) throws UnknownHostException
- public static InetAddress[] **getAllByName**(String hostName) throws UnknownHostException
- public static InetAddress **getLocalHost**() throws UnknownHostException



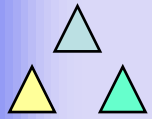
Getter методи

- `public String getHostName()`
- `public byte[] getAddress()`
- `public String getHostAddress()`

\mathcal{A}
 \mathcal{A}



ДЕМОСТРАЦИЯ



Пример

HostLookup.java

nslookup is an old Unix utility that converts hostnames to IP addresses and IP addresses to hostnames. It has two modes: interactive and command-line. If you enter a hostname on the command line, nslookup prints the IP address of that host. If you enter an IP address on the command line, nslookup prints the hostname. If no hostname or IP address is entered on the command line, nslookup enters interactive mode, in which it reads hostnames and IP addresses from standard input and echoes back the corresponding IP addresses and hostnames until you type "exit".



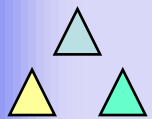
java.net.URL

- Представлява абстракция на Uniform Resource Locator като <http://www.hamsterdance.com/> или <ftp://ftp.redhat.com/pub/>
public final class URL extends Object
implements Serializable
- Представя URL, като обект с отделни полета, като протокол, източник, порт и т.н.



Конструктори

- public URL(String url) throws MalformedURLException
([ProtocolTester.java](#))
- public URL(String protocol, String hostname, String file)
throws MalformedURLException
([ProtocolTesterApplet.java](#))
- public URL(String protocol, String host, int port, String
file) throws MalformedURLException

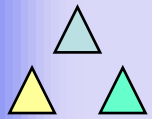


Разделяне на URL на части

- `public String getProtocol()`
- `public String getHost()`
- `public int getPort()`
- `public int getDefaultPort()`
- `public String getFile()`
- `public String getPath()`
- `public String getRef()`
- `public String getQuery()`
- `public String getUserInfo()`

\mathcal{A}

\mathcal{A}

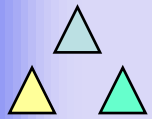


Примерна задача:

[URLSplitter.java](#)

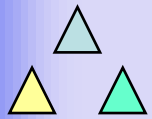
А

А



Методи за получаване на данни от URL

- `public InputStream openStream() throws IOException`
([SourceViewer.java](#))
- `public URLConnection openConnection() throws IOException`
- `public URLConnection openConnection(Proxy proxy) throws IOException // 1.5`
- `public Object getContent() throws IOException`
- `public Object getContent(Class[] classes) throws IOException // 1.3`



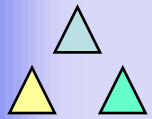
Символи позволени в URL

- A-Z
- a-z
- 0-9
- _ . ! ~ * '
- / & ? @ # ; \$ + = и % се използват с точно определена цел и ако присъстват те се кодират: превръщат се в байтове и всеки байт се пише като %последван от две 16-тични цифри: напр. + е %2B.



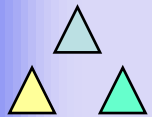
Използване на GET метод

```
// this string should be URL-encoded
URL url = new URL(
    http://myhost/cgi-
    bin/login.cgi?Name=Pat&Password=foobar
);
BufferedReader bin = new BufferedReader (
    new InputStreamReader(
        url.openStream( ) ) );
String line;
while ((line = bin.readLine( )) != null)
    System.out.println( line );
```



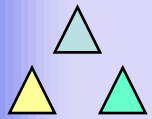
Класове за кодиране и декодиране

- `public class URLDecoder extends Object`
- `public class URLEncoder extends Object`



Задача

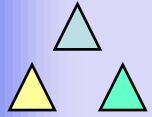
- Пример за използване на POST метод
Post.java



java.net.URI

- Докато URL обекта е object is a представяне на протокола в приложния слой на модела, то URI обект е само за парсване на стринг и обработката му.

\mathcal{A}
 \mathcal{A}



scheme:scheme-specific- part:fragment

- `public String getScheme()`
- `public String getSchemeSpecificPart()`
- `public String getRawSchemeSpecificPart()`
- `public String getFragment()`
- `public String getRawFragment()`