

## Работа с сетью (пакет java.net)

JAVA поддерживает TCP/IP за счет расширения уже имеющихся интерфейсов потокового ввода-вывода и добавления средств, необходимых для построения объектов ввода-вывода в сети.

Класс **InetAddress** используется для инкапсуляции как числового IP-адреса, так и доменного имени для этого адреса.

JAVA поддерживает адреса IPv4 и IPv6. Подклассы **InetAddress**: **Inet4Address** и **Inet6Address**.

Для создания объекта **InetAddress** нужно использовать один из методов-фабрик.

Методы- фабрики (factory method) – это соглашение, в соответствии с которым статические методы класса возвращают экземпляр этого класса.

Методы-фабрики InetAddress:

**static InetAddress getLocalHost( )**

**throws UnknownHostException**

- возвращает объект InetAddress, представляющий локальный хост

**static InetAddress getByName(String hostName)**

**throws UnknownHostException**

- возвращает InetAddress хоста по заданному имени

**static InetAddress[ ] getAllByName(String hostName)**

**throws UnknownHostException**

- возвращает массив InetAddress, представляющий все адреса, в которые преобразуется конкретное имя.

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

```
class InetAddressTest{  
    public static void main(String args[])  
        throws UnknownHostException {  
  
        InetAddress Address = InetAddress.getLocalHost();  
        System.out.println(Address) ;  
        Address = InetAddress.getByName("google.com");  
        System.out.println(Address);  
        InetAddress SW [] = InetAddress.getAllByName("ya.ru") ;  
  
        for (int i=0; i<SW.length; i++)  
            System.out.println(SW[i]);  
    }  
}
```

C:\WINDOWS\system32\cmd.exe

Системе не удается найти указанный путь.

alena/192.168.0.26

google.com/173.194.32.168

ya.ru/213.180.193.3

ya.ru/93.158.134.3

ya.ru/213.180.204.3

Для продолжения нажмите любую клавишу . . .

Сокеты tcp /IP применяются для реализации надежных двунаправленных, постоянных соединений между точками - хостами в Internet на основе потоков. Сокет может использоваться для подключения системы ввода-вывода java к другим программам, которые могут находиться как на локальной машине, так и на любой машине в Internet.

В JAVA существуют два вида сокетов TCP.

Класс **ServerSocket** предназначен для серверов.

Класс **Socket** предназначен для клиентов ( разработан так, чтобы соединяться с серверными сокетами и инициировать обмен по протоколу).

Конструкторы

**Socket (String hostName, int port) throws UnknownHostException, IOException** - создает сокет, подключенный к заданным хосту и порту

**Socket(InetAddress ipAddress, int port) throws IOException** создает сокет, используя ранее существующий объект InetAddress и порт.

**InetAddress getAddress ()** - возвращает InetAddress, ассоциированный с объектом Socket. В случае если сокет не подключен, возвращает null.

**int getPort ()** - возвращает удаленный порт, к которому подключен вызывающий объект Socket. Если сокет не подключен, возвращает 0.

**int getLocalPort ()** - возвращает локальный порт, к которому привязан вызывающий объект Socket. Если сокет не привязан, возвращает -1.

```
import java.net.*;  
import java.io.*;  
  
class Whois {  
    public static void main(String args[]) throws Exception {  
        int c;  
        Socket s = new Socket("whois.ripn.net", 43);  
        InputStream in = s.getInputStream();  
        OutputStream out = s.getOutputStream();  
        String str = (args.length == 0 ? "ya.ru" : args[0]) + "\n";  
        byte buf[] = str.getBytes();  
        out.write(buf);  
  
        while ((c = in.read()) != -1){  
            System.out.print((char) c);  
        }  
  
        s.close();  
}}
```

C:\WINDOWS\system32\cmd.exe

Системе не удастся найти указанный путь.

% By submitting a query to RIPN's Whois Service

% you agree to abide by the following terms of use:

% <http://www.ripn.net/about/servpol.html#3.2> (in Russian)

% <http://www.ripn.net/about/en/servpol.html#3.2> (in English).

domain: YA.RU

ns1.yandex.ru.

ns2.yandex.ru.

state: REGISTERED, DELEGATED, VERIFIED

org: YANDEX, LLC.

registrar: RU-CENTER-REG-RIPN

admin-contact: <https://www.nic.ru/whois>

created: 1999.07.12

paid-till: 2014.08.01

free-date: 2014.09.01

source: TCI

Last updated on 2013.11.29 22:26:39 MSK

Для продолжения нажмите любую клавишу . . .



Класс **URL** ( Uniform Resource Locator )

Конструктор

**URL(String urlSpecifier) throws MalformedURLException**

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

```
class URLDemo {  
    public static void main(String args[]) throws MalformedURLException {  
        URL hp = new URL("http://ya.ru/ffffffffff ");  
        System.out.println( hp.getProtocol());  
        System.out.println( hp.getPort () );  
        System.out.println( hp.getHost());  
        System.out.println ( hp.getFile ());  
        System.out.println( hp.toExternalForm () );  
    } }
```

C:\WINDOWS\system32\cmd.exe

Системе не удастся найти указанный путь.

http

-1

ya.ru

/fffffffffff

http://ya.ru/fffffffffff

Для продолжения нажмите любую клавишу . . .

## Сервлеты

Сервлетами (servlet) называются небольшие программы, которые выполняются на стороне сервера Web-соединения.

Для создания сервлетов вам будет необходим доступ к среде разработки сервлетов. Это может быть

**JSDK – Java Servlet Development Kit** с **[http:// java.sun.com](http://java.sun.com)**

или **Tomcat**. Tomcat представляет собой открытый продукт, который поддерживает проект **jakarta** от фонда Apache Software Foundation.

Этот продукт содержит библиотеки классов, документацию и среду времени выполнения, что будет необходимо для создания и тестирования сервлетов. **<http://jakarta.apache.org>**

Классы и интерфейсы, необходимые для построения сервлетов, содержатся в двух пакетах: **javax.servlet** и **javax.servlet.http**

Они являются стандартными расширениями, предлагаемыми Tomcat. (не входят в состав Java SE6)

Жизненный цикл сервлета определяют три основных метода:

**init()**, **service()** и **destroy()**. Они реализуются каждым сервлетом и вызываются сервером в определенное время.

1. Пусть пользователь ввел в окне браузера URL-адрес. На основании этого URL-адреса браузер генерирует HTTP-запрос, посылаемый соответствующему серверу.
2. HTTP-запрос принимает Web-сервер. Сервер находит соответствие между запросом и конкретным сервлетом. Сервлет динамически принимается и загружается в адресное пространство сервера.
3. Сервер вызывает метод **init()** сервлета. Этот метод вызывается только тогда, когда сервлет впервые загружается в память компьютера. Сервлету можно передавать параметры инициализации, поэтому он может конфигурировать себя самостоятельно.
4. Сервер вызывает метод **service()** сервлета. Этот метод вызывается для обработки HTTP-запроса. Сервлет может считывать данные, содержащиеся в HTTP-запросе. Он может также сформулировать HTTP-отклик клиенту. Сервлет остается в адресном пространстве и является доступным для обработки любых других HTTP-запросов, полученных от клиентов. Метод **service ()** вызывается для каждого HTTP-запроса.
5. Сервер может принять решение выгрузить сервлет из памяти. Для освобождения ресурсов, выделенных для сервлета, сервер вызывает метод **destroy ()**.