

OOP_22.pdf — Mozilla Firefox

file:///home/ids/Вузовская%20дичь/oop/OOP_22.pdf

Очень простой сервлет

Помимо ко, нам нужно импортировать два пакета для сервлетов. Помните, что они не входят в состав стандартной библиотеки Java — придется загрузить их отдельно.

Помимо ко, нам нужно импортировать два пакета для сервлетов. Помните, что они не входят в состав стандартной библиотеки Java — придется загрузить их отдельно.

Помимо ко, нам нужно импортировать два пакета для сервлетов. Помните, что они не входят в состав стандартной библиотеки Java — придется загрузить их отдельно.

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class MyServletA extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String message = "Если вы видите это сообщение, значит сервер работает!";
        out.println("<HTML><BODY>");
        out.println("<H1>" + message);
        out.println("</BODY></HTML>");
        out.close();
    }
}
```

Перепишем докет для обработки запросов GET-сообщений по HTTP.

HTML-страница со ссылкой

```
<HTML>
<BODY>
<a href="servlets/MyServletA">Ссылка на сервлет</a>
</BODY>
</HTML>
```

Ради забавы сделаем так, чтобы наш генератор фраз работал в виде сервлета

Несмотря на обещание, большие слова не говорят о сервлетах, мы не смогли устоять перед искушением сериализации (да, у нас богатый словарный запас) генератора фраз из

Попробуйте мой новый генератор фраз, и вы узнаете, насколько красиво, как ваш начальник или парни из отдела продаж.

classes — Dolphin

Домашняя папка > tomcat > apache-tomcat-10.1.7 > webapps > first_servlet > WEB-INF > classes

classes x 22 x

MyServletA.java

OOP_22.pdf — Mozilla Firefox

file:///home/ids/Вузовская%20дичь/oop/OOP_22.pdf

Очень простой сервлет

Помимо ко, нам нужно импортировать два пакета для сервлетов. Помните, что они не входят в состав стандартной библиотеки Java — придется загрузить их отдельно.

Помимо ко, нам нужно импортировать два пакета для сервлетов. Помните, что они не входят в состав стандартной библиотеки Java — придется загрузить их отдельно.

Помимо ко, нам нужно импортировать два пакета для сервлетов. Помните, что они не входят в состав стандартной библиотеки Java — придется загрузить их отдельно.

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class MyServletA extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String message = "Если вы видите это сообщение, значит сервер работает!";
        out.println("<HTML><BODY>");
        out.println("<H1>" + message);
        out.println("</BODY></HTML>");
        out.close();
    }
}
```

Перепишем докет для обработки запросов GET-сообщений по HTTP.

HTML-страница со ссылкой

```
<HTML>
<BODY>
<a href="servlets/MyServletA">Ссылка на сервлет</a>
</BODY>
</HTML>
```

Ради забавы сделаем так, чтобы наш генератор фраз работал в виде сервлета

Несмотря на обещание, большие слова не говорят о сервлетах, мы не смогли устоять перед искушением сериализации (да, у нас богатый словарный запас) генератора фраз из

Попробуйте мой новый генератор фраз, и вы узнаете, насколько красиво, как ваш начальник или парни из отдела продаж.

classes — Dolphin

Домашняя папка > tomcat > apache-tomcat-10.1.7 > webapps > first_servlet > WEB-INF > classes

classes x 22 x

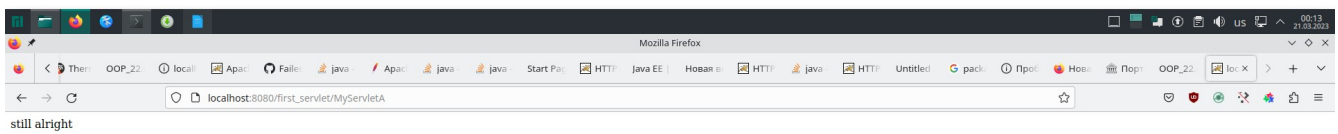
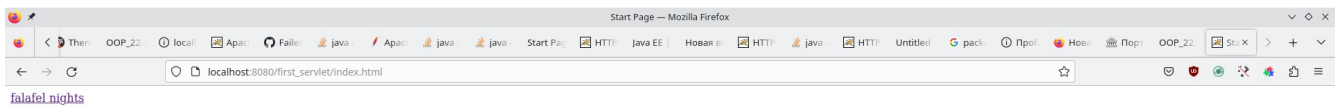
MyServletA.java

MyServletA.class

classes: zsh — Konsole

```
~/tomcat/apache-tomcat-10.1.7/webapps/first_servlet/WEB-INF/classes$ javac -cp /home/ids/tomcat/apache-tomcat-10.1.7/lib/servlet-api.jar MyServletA.java
~/tomcat/apache-tomcat-10.1.7/webapps/first_servlet/WEB-INF/classes$ tomcat_startup

Using CATALINA_BASE: /home/ids/tomcat/apache-tomcat-10.1.7
Using CATALINA_HOME: /home/ids/tomcat/apache-tomcat-10.1.7
Using CATALINA_TMPDIR: /home/ids/tomcat/apache-tomcat-10.1.7/temp
Using JRE_HOME: /usr
Using CLASSPATH: /home/ids/tomcat/apache-tomcat-10.1.7/bin/bootstrap.jar:/home/ids/tomcat/apache-tomcat-10.1.7/bin/tomcat-juli.jar
Tomcat started.
```



.zchrc:

```
alias tomcat_startup="sudo /home/ids/tomcat/apache-tomcat-10.1.7/bin/startup.sh"
alias tomcat_shutdown="sudo /home/ids/tomcat/apache-tomcat-10.1.7/bin/shutdown.sh"
```

Index.html:

```
<!DOCTYPE html>
<html>
```

```
<head>
  <title>Start Page</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
  <a href="MyServletA"> falafel nights </a>
</body>
</html>
```

Web.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-
app_3_1.xsd">
  <servlet>
    <servlet-name>MyServletA</servlet-name>
    <servlet-class>MyServletA</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>MyServletA</servlet-name>
    <url-pattern>/MyServletA</url-pattern>
  </servlet-mapping>
  <session-config>
    <session-timeout>
      30
    </session-timeout>
  </session-config>
</web-app>
```

MyServletA.java:

```
import java.io.*;
import jakarta.servlet.*;
import jakarta.servlet.http.*;

public class MyServletA extends HttpServlet {

  public void doGet(HttpServletRequest request, HttpServletResponse response)
  throws IOException, ServletException
  {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    String message = "still alright";
    out.println("<HTML><BODY>");
    out.println("<HI>" + message + "</HI>");
    out.println("</BODY></HTML>");
  }
}
```

```
}  
}
```

MyRemote.java

```
package lab_22;  
import java.rmi.*;  
  
public interface MyRemote extends Remote {  
    public String sayHello() throws RemoteException;  
}
```

MyRemoteImpl.java

```
package lab_22;  
import java.rmi.*;  
import java.rmi.server.*;  
  
public class MyRemoteImpl extends UnicastRemoteObject implements MyRemote {  
    public String sayHello() {  
        return "server say: hello";  
    }  
    public MyRemoteImpl() throws RemoteException {}  
    public static void main(String[] args) {  
        try {  
            MyRemote service = new MyRemoteImpl();  
            Naming.rebind("Hello", service);  
        } catch (Exception ex) {  
            ex.printStackTrace();  
        }  
    }  
}
```

MyRemoteClient.java

```
package lab_22;  
import java.rmi.*;  
  
public class MyRemoteClient {  
    public static void main(String[] args) {  
        new MyRemoteClient().go();  
    }  
    public void go() {  
        try {  
            MyRemote service = (MyRemote)  
Naming.lookup("rai://127.0.0.1/Remote Hello");  
            String s = service.sayHello();  
            System.out.println(s);  
        } catch (Exception ex) { ex.printStackTrace(); }  
    }  
}
```

JDK 15 Release Notes — Mozilla Firefox

ReadDiffBashPopJDK-XCannot rmicArchTherOOP_22localApacFailjavaApacjavajavaStart PaHTTJava EEНованHTTJava>+<v

https://www.oracle.com/java/technologies/javase/15-relnote-issues.html

ORACLEProducts Industries Resources Customers Partners Developers Events CompanyView AccountsContact Sales

This section describes the APIs, features, and options that were removed in Java SE 15 and JDK 15. The APIs described here are those that are provided with the Oracle JDK. It includes a complete implementation of the Java SE 15 Platform and additional Java APIs to support developing, debugging, and monitoring Java applications. Another source of information about important enhancements and new features in Java SE 15 and JDK 15 is the [Java SE 15 \(JSR 390\)](#) Platform Specification, which documents changes to the specification made between Java SE 14 and Java SE 15. This document includes the identification of removed APIs and features not described here. The descriptions below might also identify potential compatibility issues that you could encounter when migrating to JDK 15. See [CSRs Approved for JDK 15](#) for the list of CSRs closed in JDK 15.

core-libs/java.net

→ **Removal of Terminally Deprecated Solaris-specific `SO_FLOW_SL` Socket Option**

In this release, in conjunction with the removal of the Solaris port in JEP381 ([JDK-824787](#)), the JDK-specific socket option `jdk.net.ExtendedSocketOptions.SO_FLOW_SL`, which is only relevant to sockets on Solaris, and its supporting classes `SocketFlow` and `SocketFlow.Status`, have been removed.

See [JDK-8244582](#)

core-libs/java.rmi

→ **Removal of RMI Static Stub Compiler (`rmic`)**

The RMI static stub compiler `rmic` has been removed. The `rmic` tool is obsolete and has been deprecated for removal since JDK 13.

See [JDK-825319](#)

core-svc/javax.management

→ **Removal of Deprecated Constant `RMIConnectorServer.CREDENTIAL_TYPES`**

The terminally deprecated constant `javax.management.remote.rmi.RMIConnectorServer.CREDENTIAL_TYPE` has been removed. A filter pattern can be specified instead by using `RMIConnectorServer.CREDENTIALS_FILTER_PATTERN`.

See [JDK-8213222](#)

core-libs/jdk.nashorn

→ **Removal of Nashorn JavaScript Engine**

The Nashorn JavaScript engine, its APIs, and the `js` tool have been removed. The engine, the APIs, and the tool were deprecated for removal in Java 11 with the express intent to remove them in a future release.

See [JDK-8236933](#)

hotspot/gc

→ **Obsolete `-XXUseAdaptiveGCBoundary`**

The VM option `UseAdaptiveGCBoundary` is obsolete. Use of this option will produce an obsolete option warning but will otherwise be ignored.

This option was previously disabled by default, and enabling it only had an effect when also using `-XX:+UseParallelGC`. Enabling it was intended to provide a performance benefit for some applications. However, it has been disabled by default for a long time because of crashes and performance regressions.

jdk

Сортировать поРелевантность

Все

Установлены

Репозитории

AUR

Snap

Flatpak

jdk

19.0.2-1

Oracle Java Development Kit

AUR

↓

jdk-openjdk

19.0.2-1

OpenJDK Java 19 development kit

Официальные репозитории (extra)

301.0 MB

↓

jdk-android-studio

2.3

Integrate android-studio provided JDK with archlinux-java

AUR

↓

jdk-arm

19.0.2-1

Oracle Java Development Kit (v8) for ARMv7, ARMv8 and ARMv8 (64-bit)

AUR

↓

jdk-bcl

19.0.2-1

Oracle Java Development Kit (BCL)

AUR

↓

jdk-doc

19.0.2-1

Oracle Java documentation

AUR

↓

jdk-jbr-xdg

17.0.4.1-1

JetBrains Runtime Java 17 development kit - with improved Support for the XDG Base Directory Sp...

AUR

↓

jdk-jetbrains

11.6.15b-2043.56-1

OpenJDK development kit with some fixes and enhancements by JetBrains

AUR

↓

jdk-lts

17.0.6-1

Oracle Java Development Kit (LTS release)

AUR

↓

jdk-lts-doc

17.0.6-1

Oracle Java documentation (LTS release)

AUR

↓

jdk-microsoft-openjdk

17.0.5-1

Microsoft Build of OpenJDK

AUR

↓

jdk-mission-control-bin

19.0.2-1

OpenJDK Mission Control

AUR

↓

jdk-openj9

19.0.2-1

OpenJDK Java 18 development kit

AUR

↓

jdk-openj9-bin

19.0.2-1

Eclipse (former IBM) OpenJ9 with openjdk18

AUR

↓

jdk-openjdk-xdg

19.0.2-1

OpenJDK Java 19 development kit - with improved Support for the XDG Base Directory Specification

AUR

↓

jdk-openjdk-xdg-bin

19.0.2-1

OpenJDK Java 19 development kit - with improved Support for the XDG Base Directory Specification

AUR

↓

jdk-temurin

19.0.2-1

AUR

↓

jdk10

19.0.2-1

AUR

↓

jdk-openjdk 19.0.2-1

OpenJDK Java 19 development kit

УдалитьПереустановить

ПодробностиЗависимостиФайлы

<https://openjdk.java.net/>

Лицензии:custom

Репозиторий:extra

Упаковки:Frederik Schwan freswa@archlinux.org

Дата сборки:Ср 18 янв 2023 18:17:00

Дата установки:Сб 04 мар 2023 12:39:50

Причина установки:Установлен как зависимость для другого пакета

Отметить как специально установленный

Подтверждение:Signature