
DataArt

DataArt

Java-интенсив

DataArt

Про домашнее задание

- Не забывайте закрывать closable – ресурсы через try caught
- Если вам нужен интерфейс (нужны методы) Map, а не HashMap, то объявляйте переменную типа именно Map (это верно для любых классов, не только Map)
- Старайтесь параметризовывать методы, где это уместно

DataArt

Chapter 5.1

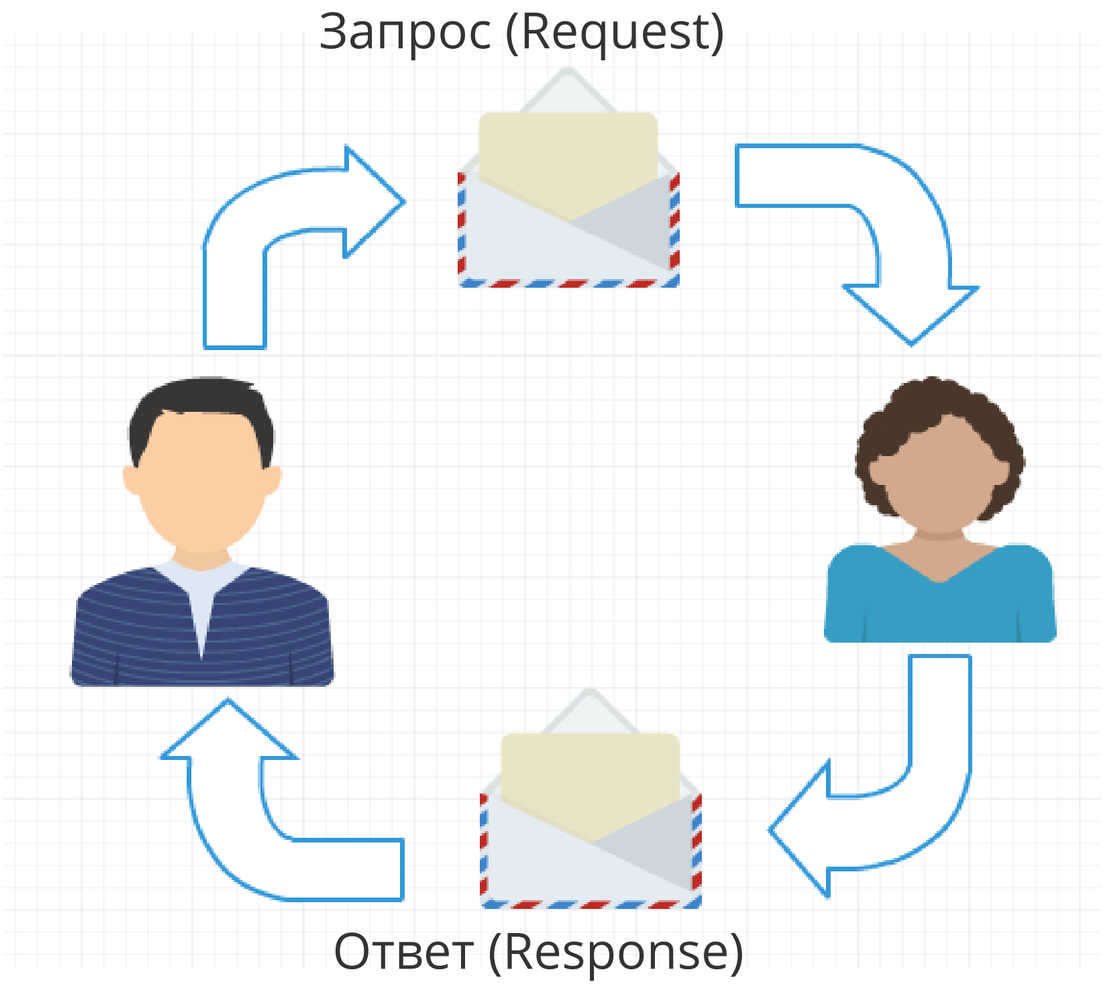
Протокол HTTP

Что такое протокол?

DataArt

- Протокол – это набор правил.
- Понятие протокола на самом деле встречается и в реальной жизни
- В интернете информация передается по различным специальным протоколам

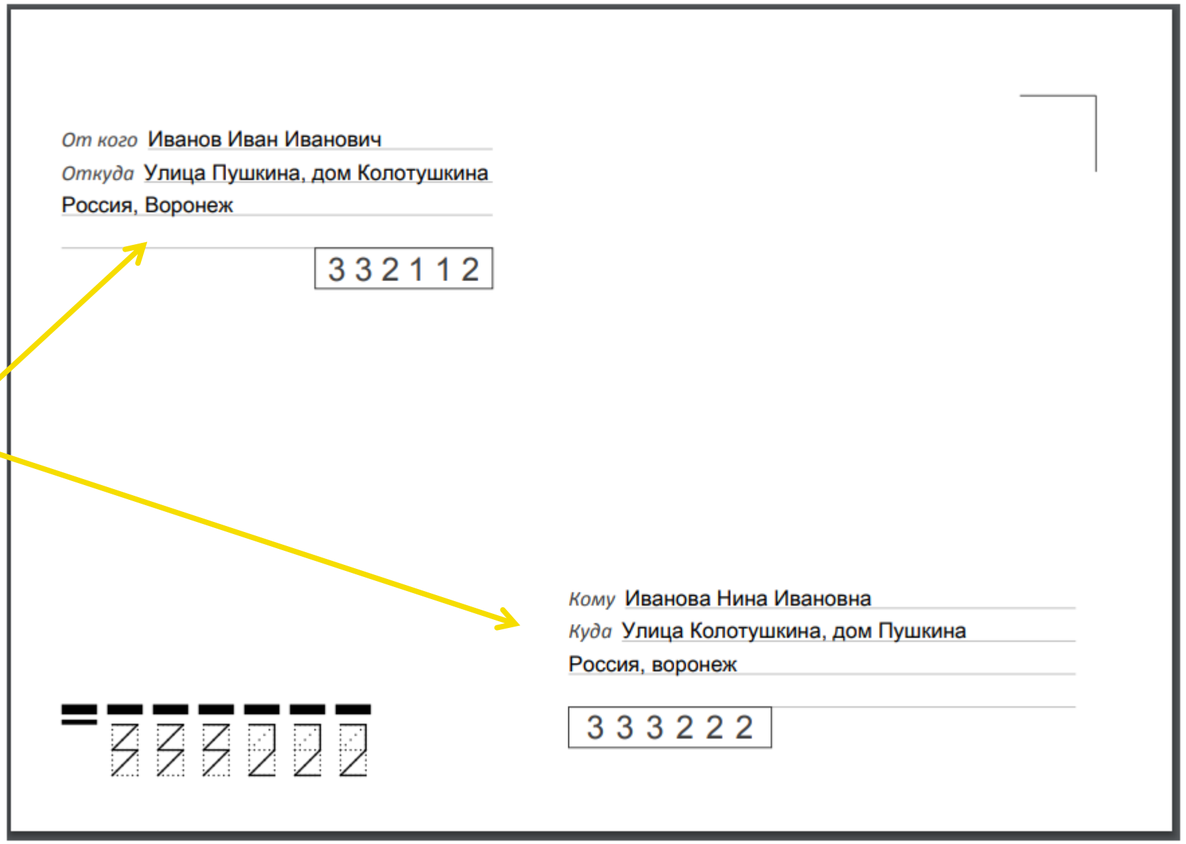
Что такое протокол (на примере обычной почты)



Что такое протокол (на примере обычной почты)

DataArt

Техническая информация



Что такое протокол (на примере обычной почты)

DataArt

- Наличие технической информации для почты (индекс, адреса, ФИО) – это как раз “правила” или “протокол” почты
- Если все указано верно – почта сможет доставить письмо, иначе – нет
- Однако, почта не знает, какая информация внутри конверта
- Поэтому протокол почты можно назвать “транспортным”

Что такое протокол

DataArt

- Для передачи данных в интернете чаще всего так или иначе используются протоколы транспортного уровня
- TCP/IP
- UDP
- Каждый имеет набор правил и содержит под собой другие, более простые протоколы (стек протоколов)

Что такое протокол

DataArt

- UDP – негарантированная доставка (выше скорость)
- TCP/IP – гарантированная доставка (ниже скорость)

Гарантированная доставка

DataArt

Что такое протокол

DataArt

- Заметим, что почта не знает о содержании письма
- Если вы общаетесь с каким-то специфичным учреждением, на формат переписки могут быть наложены дополнительные правила
- То есть мы получаем “стек” протоколов (протокол учреждения “использует” протокол почты)

HTTP

DataArt

- HyperText Transfer Protocol — «протокол передачи гипертекста»
- Работает над (использует) TCP/IP
- Работает по принципу запрос/ответ
- Веб-приложения чаще всего общаются через HTTP
- В нашем примере про почту HTTP - содержание письма

Как устроен HTTP

DataArt

- Исследуем, как устроен HTTP
- Рассмотрим пример запроса, сгенерированного браузером
- Откроем TCP/IP соединение с помощью JAVA, и посмотрим, что нам пришлет браузер (выведем HTTP запрос в консоль)
- Браузер у нас это “клиент”, а Java приложение – “сервер”

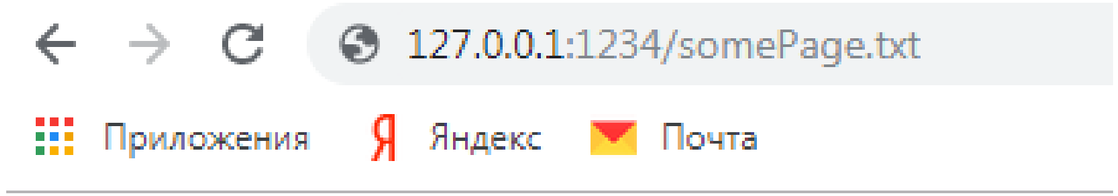
Как устроен HTTP

DataArt

```
public class Main {

    public static void main(String[] args) throws IOException {
        ServerSocket serverSocket = new ServerSocket(1234);
        Socket socket = serverSocket.accept();
        InputStream inputStream = socket.getInputStream();
        Scanner scanner = new Scanner(inputStream);
        while (scanner.hasNext()) {
            System.out.println(scanner.next());
        }
        inputStream.close();
    }


}
```

Как устроен HTTP

DataArt



Первая строка:метод, запрашиваемый “ресурс” и версия протокола

```
GET /somePage.txt HTTP/1.1
Host: 127.0.0.1:1234
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.131 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Accept-Encoding: gzip, deflate, br
Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7
```

Заголовки (HEADERS), фактически – ассоциативный массив (ключ-значение), техническая информация

Как устроен HTTP

DataArt

- Метод HTTP запроса – фактически глагол, которым вы говорите серверу, что вы от него хотите
- Два самых используемых метода: GET, POST
- GET в идеале – запрос на получение ресурса
- POST – отправка чего-либо на сервер (например, картинки)
- У POST есть “тело” запроса

Как устроен HTTP

DataArt

```
POST http://www.webservicex.com/globalweather.asmx HTTP/1.1
Host: www.webservicex.com
Content-Length: 341
Content-Type: application/soap+xml

<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Header></soap:Header>
  <soap:Body>
    <ns1:GetWeather xmlns:ns1='http://www.webserviceX.NET'>
       <ns1:CityName>Montreal</ns1:CityName>
       <ns1:CountryName>Canada</ns1:CountryName>
    </ns1:GetWeather>
  </soap:Body>
</soap:Envelope>
```

Тело отделяется с помощью \r\n\r\n

Тело (BODY) POST - запроса

Как устроен HTTP

DataArt

- HTTP сервер обычно генерирует какой-то ответ (response).
- Ответ также содержит техническую информацию, а также данные
- Браузер анализирует ответ и, возможно, рисует что-то в окне браузера
- Давайте “захардкодим” ответ (то есть сервер на любой запрос будет возвращать константную строку)

Как устроен HTTP

DataArt

```
private static  final String HARDCODED_RESPONSE = "HTTP/1.1 200 OK\n" +
        "Date: Mon, 27 Jul 2009 12:28:53 GMT\n" +
        "Server: Apache/2.2.14 (Win32)\n" +
        "Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT\n" +
        "Content-Length: 15\n" +
        "Content-Type: text/html\n" +
        "Connection: Closed" +
        "\r\n\r\n" +
        "Hello java school!!!";
```

Как устроен HTTP

DataArt

```
public static void main(String[] args) throws IOException {
    ServerSocket serverSocket = new ServerSocket(1234);
    Socket socket = serverSocket.accept();
    InputStream inputStream = socket.getInputStream();
    Scanner scanner = new Scanner(inputStream);
    String line = null;
    do{
        line = scanner.nextLine();
        System.out.println(line);
    }
    while (!"".equals(line.trim()));

    //lets send a response
    socket.getOutputStream().write(HARDCODED_RESPONSE.getBytes());

    serverSocket.close();
}
```

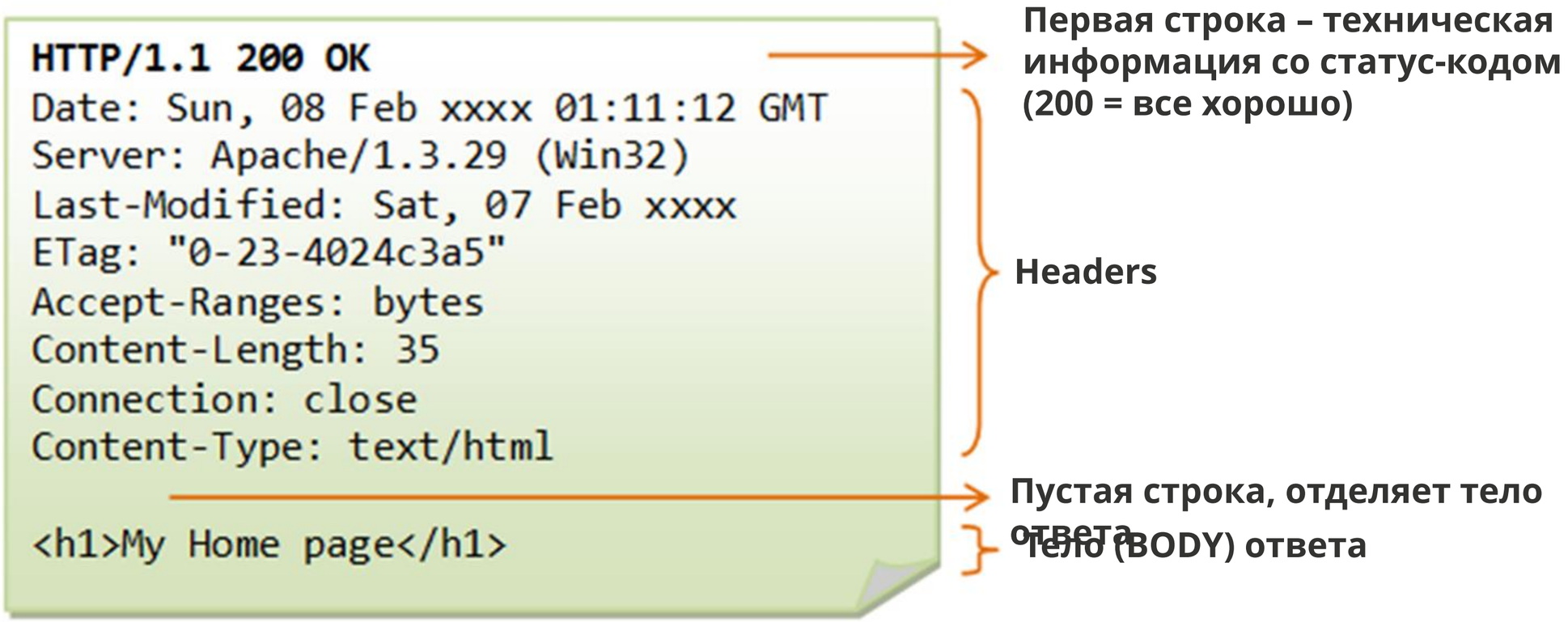
Почему у нас сообщение отобразилось не полностью?

DataArt

```
private static  final String HARDCODED_RESPONSE = "HTTP/1.1 200 OK\n" +
        "Date: Mon, 27 Jul 2009 12:28:53 GMT\n" +
        "Server: Apache/2.2.14 (Win32)\n" +
        "Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT\n" +
        "Content-Length: 15\n" +
        "Content-Type: text/html\n" +
        "Connection: Closed" +
        "\r\n\r\n" +
        "Hello java school!!!";
```

Рассмотрим Response более детально

DataArt



HTTP Request method

DataArt

- Если не говорить про REST, то наиболее используемые типы запросов – GET и POST
- GET передает данные в адресной строке браузера (это может не подходить вам, если вы передаете на сервер, например, пароль)
- POST передает данные в “теле” запроса. Можно передавать большие количества данных.

DataArt

Chapter 5.2

HTML

HTTP Request method

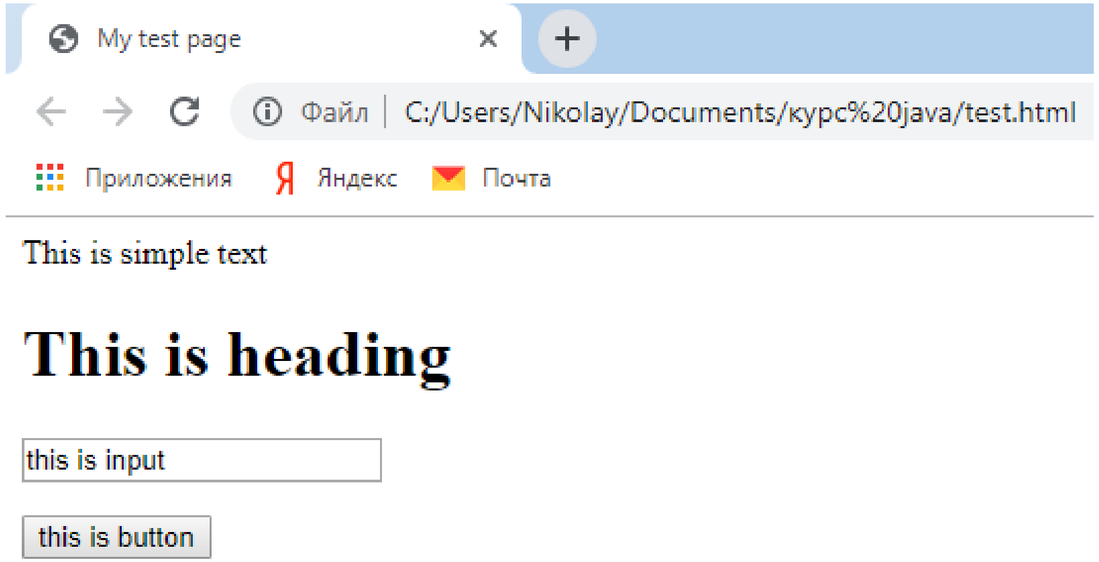
DataArt

- Помните, что HTTP это HyperText Transfer Protocol?
- Так вот, HTML - HyperText Markup Language — «язык гипертекстовой разметки»
- Вообще, HTML - частный случай XML, но эту часть теории изучите дома
- HTML, это фактически просто текст, но со специальными директивами – тегами, указывающими браузеру, как отображать контент

Идем сразу к практике

DataArt

```
1  <html>
2  <head>
3      <title>My test page</title>
4  </head>
5
6  <body>
7
8  This is simple text
9  <p>
10 <h1>This is heading</h1>
11 <p>
12 <input type = "text" value="this is input">
13 <p>
14 <input type = "button" value="this is button">
15 </body>
16
17 </html>
```



Как передать через html данные на сервер?

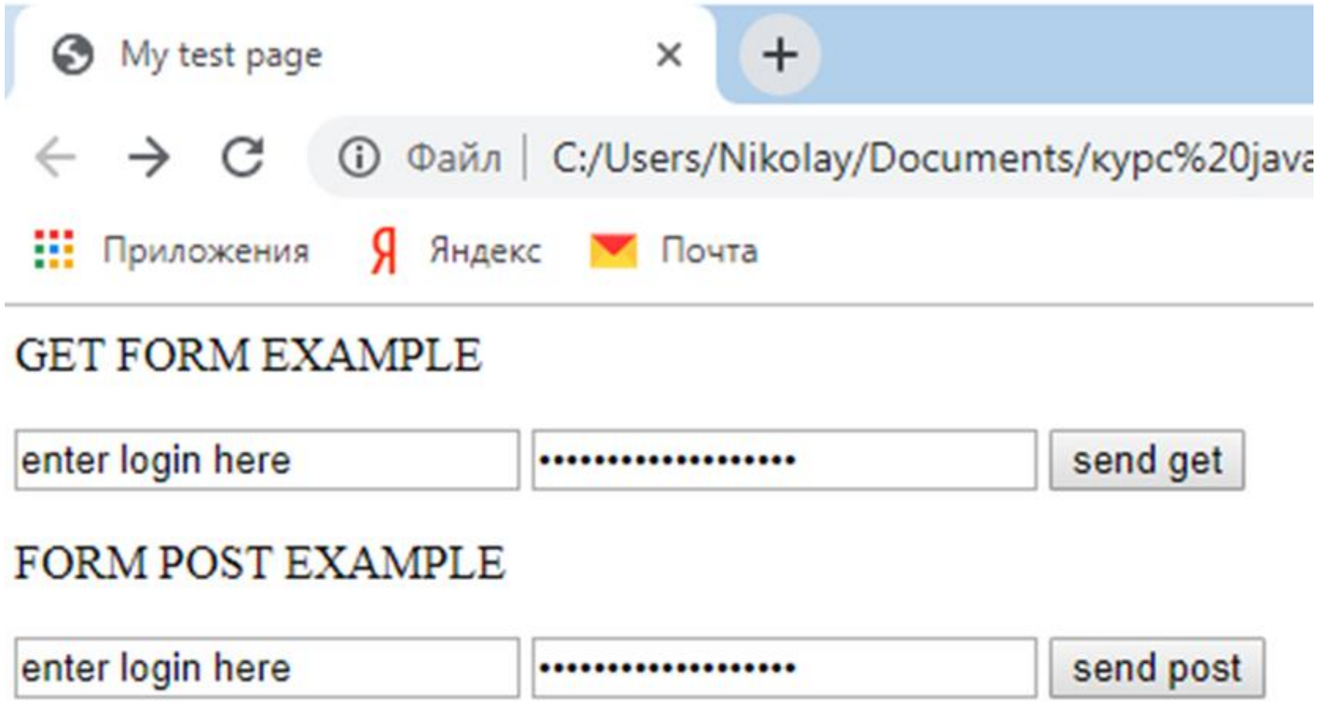
DataArt

- Нам нужен специальный tag - <form>
- Все инпуты, которые будут внутри формы, будут отправлены на сервер
- Но для этого нужно специальное действие – submit.
- Давайте посмотрим, как создать форму и заставить её отправить данные на сервер методами GET или POST

Как передать через html данные на сервер?

DataArt

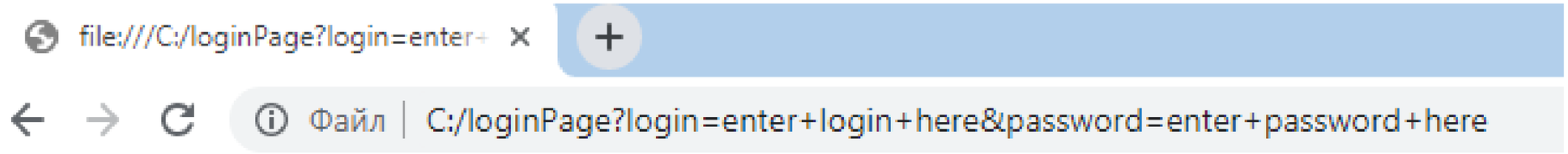
```
<html>
<head>
    <title>My test page</title>
</head>

<body>
<p>
GET FORM EXAMPLE
<form action="/loginPage" method="GET">
    <input type = "text" name="login" value="enter login here">
    <input type = "password" name="password" value="enter password here">
    <input type = "submit" value="send get">
</form>

<p>
FORM POST EXAMPLE
<form action="/loginPage" method="POST">
    <input type = "text" name="login" value="enter login here">
    <input type = "password" name="password" value="enter password here">
    <input type = "submit" value="send post">
</form>
</body>

</html>
```



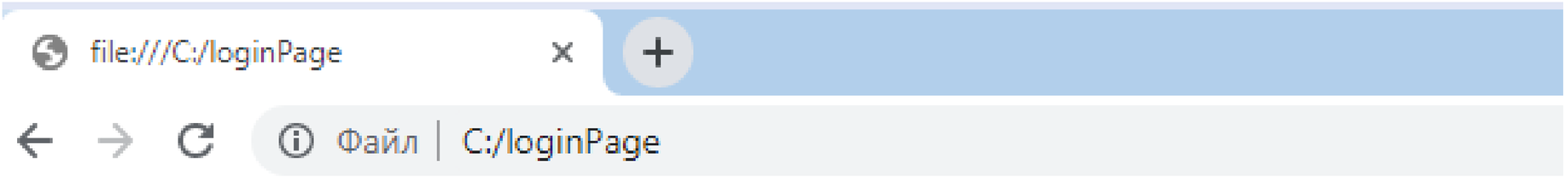
HTTP

DataArt

Жмем Send Get (данные в адресной строке)



Жмем Send Post (данные в BODY запроса)



DataArt

Вопросы и ответы

DataArt

Chapter 5.3

Сборщик Maven

Maven

DataArt

- Часто нам нужно собирать проекты особым образом
- Кроме того, нужно подключать к проекту внешние библиотеки
- Можно конечно скачивать jar вручную, но это не очень удобно
- Maven решает проблемы внешних зависимостей и сборки
- Основной файл Maven – pom.xml

Pom.xml

DataArt

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
         http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.spring.aspect</groupId>
    <artifactId>SpringAspect</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <url>http://maven.apache.org</url>
    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.0.1</version>
            <scope>test</scope>
        </dependency>
    </dependencies>

</project>
```

Maven основные команды

DataArt

- Clean – очищаем (удаляем) собранный проект
- Compile – компилируем проект
- Test – запускаем тесты
- Package - Создание .jar файла или war, ear в зависимости от типа проекта
- Install - Копирование .jar (war , ear) в локальный репозиторий

Домашнее задание

DataArt

- По итогам курса мы решили выделить несколько наиболее успешных студентов
- Критерии такие : за каждую работающую лабу (кроме 1ой, её целью было понять ваш уровень) будет начислен 1 балл.
- За 2 последние лабы может быть получено больше баллов, для каждой лабы может быть указано отдельно

Домашнее задание

DataArt

- Используем maven-проект
- Создать http-сервер, который может реагировать на запросы и отдавать в браузер текстовые/html файлы
- Максимум – 5 баллов

Критерии

DataArt

- Приложение работает и корректно отдает txt/html файлы на “рендеринг” браузеру = 2 балла
- В приложении присутствуют классы Request и Response (с нужными методами, напр. генерирующими Headers, или toString для request (если нужен) = 1 балл
- Приложение распознает не только файлы из корневой папки (вида /hello.txt) но и из подпапок (/myFiles/task.html) = 1 балл
- В приложении есть не более 2х недочетов вида: не закрывается closeable, совсем уж непонятные названия переменных, игнорируются

Не забыть

DataArt

- К следующей лекции поставьте себе Maven
- К следующей лекции поставьте себе Apache Tomcat (версия 8+) (просто скачайте)

DataArt

Вопросы и ответы


DataArt