**Практическое задание 5: Java Servlets.**

**Цель:**

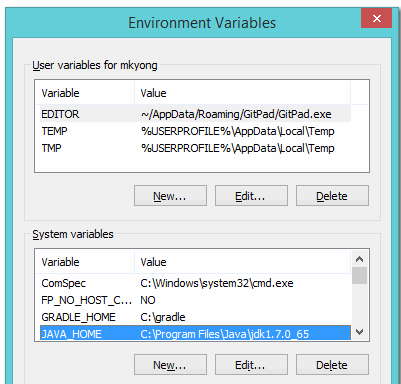
1. Настроить рабочую среду для работы java & web
2. Разобраться с работой java servlets
3. Создать простое web-приложение с использованием сервлетов

**Задание**

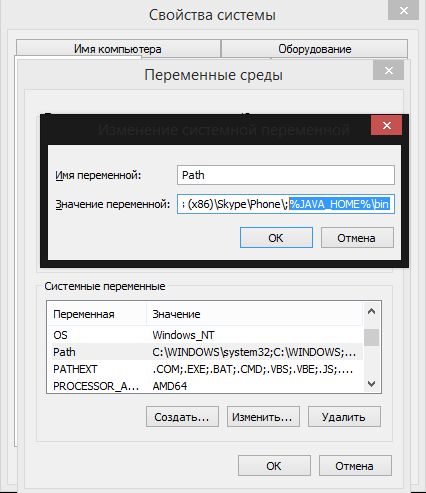
**1. Настроить рабочую среду**

* Установить JDK (у кого не установлен). Скачать можно [здесь](https://www.oracle.com/technetwork/java/javaee/downloads/jdk8-downloads-2133151.html).
* Добавить системную переменную (БЕЗ /bin) **JAVA\_HOME**

**Windows**



* Обновить переменную PATH (для этого добавить к существующему значению точку с запятой (если ее нет) и затем JAVA\_HOME\bin)



**Linux (Рекомендуется для начала сделать копию файла /etc/profile)**

-sudo vim /etc/profile - Добавить строку export JAVA\_HOME=/bin/java/jdk1.8.0\_60 - Обновить PATH   export PATH=$PATH:$JAVA\_HOME/bin - Загрузить изменения   sudo source /etc/profile

* Скачать Apache Tomcat с [официального сайта](http://ftp.byfly.by/pub/apache.org/tomcat/tomcat-8/v8.5.39/bin/apache-tomcat-8.5.39.zip), распаковать в С:\Program Files/tomcat
* Скачать [Servlet API Library](http://central.maven.org/maven2/javax/servlet/javax.servlet-api/3.1.0/javax.servlet-api-3.1.0.jar).

*Примечание: для разработки на Java рекомендуется использовать* [*IntelliJ IDEA*](https://www.jetbrains.com/idea/download/#section=windows) *(понадобится Ultimate Edition).*

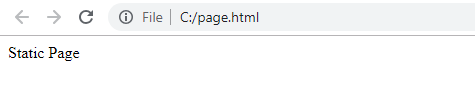
**2. Создать простое web-приложение, основанное на java servlets**

1. Код должен компилироваться и запускаться на JDK 8 / Tomcat 8
2. Приложение должно открываться по запросу [http://localhost:8080](http://localhost:8080/)
3. При открытии на странице должен выводиться текст “Java Web Application” (потребуется создать index.jsp / index.html файлы).
4. По запросу <http://localhost:8080/status> должен показываться текст “Application Is Running”
5. По запросу [http://localhost:8080/get?name={NAME}](http://localhost:8080/get?name=%7BNAME%7D)  должен показываться текст “Name is {NAME}”, {NAME} - любой текст длиной не более 100 символов.
6. При вызове запроса POST <http://localhost:8080/check> ответ от сервера должен содержать данные в формате json (напр. {“sucesss” : true}).      *Примечание: рекомендуется использовать Postman для отправки запроса методом POST.*
7. При вызове запроса <http://localhost:8080/page> на экран должно выводиться содержимое статической страницы page.html (файл следует разместить в директории WEB-INF)

Страница:



На экране:



**На 9-10 баллов:**

1. Реализовать пример навигации через **forward / redirect** a. Запрос <http://localhost:8080/test1> должен перенаправляться на <http://localhost:8080/status>(см. пункт 4) **,** должен показываться такой же текст, т.е. “Application Is Running”

\* Адресная строка браузера должна содержать  <http://localhost:8080/test1>

b. Запрос [http://localhost:8080/](http://localhost:8080/page)[test2](http://localhost:8080/test2)должен перенаправляться на <http://localhost:8080/page>**,** на странице должно отображаться содержимое файла page.html

\*Адресная строка браузера должна содержать  [http://localhost:8080](http://localhost:8080/test)/[page](http://localhost:8080/page)*Примечание: потребуется реализация 2-х новых сервлетов, которые перенаправляют запросы на уже созданные сервлеты.*

2. Реализовать Filter, который выводит URL каждого запроса, HTTP Method а также время обработки запроса сервером в миллисекундах, например, *“POST -* [*http://localhost:8080/status*](http://localhost:8080/status) *- 2ms”*