|  |  |
| --- | --- |
|  | Министерство образования и науки Российской Федерации |
| Федеральное государственное автономное образовательное учреждение высшего образования |
| «Уральский федеральный университет имени первого Президента России Б. Н. Ельцина» |
| Институт радиоэлектроники и информационных технологий – РтФ |
|  |
|  | |

**ОТЧЕТ**

О ВЫПОЛНЕНИИ

ЛАБОРАТОНОЙ РАБОТЫ № 1

Руководитель ст. пр. Н. А. Архипов

Студент гр. РИМ-230971 Е.В. Треглазов

Екатеринбург 2024

Цель:

Создание простейшего веб-приложения с помощью фреймворка Spring Boot

2 Реализация дополнительного функционала

@RestController  
public class HelloController {  
  
 private ArrayList<String> arrayList = new ArrayList<>();  
 private HashMap<Integer, String> hashMap = new HashMap<>();  
 private Integer key = 0;  
  
 @GetMapping("/hello")  
 public String hello(@RequestParam(value = "name", defaultValue = "World") String name) {  
 return String.*format*("Hello %s!", name);  
 }  
  
 @GetMapping("/update-array")  
 public void updateArrayList(@RequestParam(value = "values") String values) {  
 arrayList.add(values);  
 }  
  
 @GetMapping("/show-array")  
 public ArrayList<String> showArrayList() {  
 return arrayList;  
 }  
  
 @GetMapping("/update-map")  
 public void updateHashMap(@RequestParam(value = "values") String values) {  
 hashMap.put(key++, values);  
 }  
  
 @GetMapping("/show-map")  
 public HashMap<Integer, String> showHashMap() {  
 return hashMap;  
 }  
  
 @GetMapping("/show-all-lenght")  
 public String showAllLength() {  
 int arrayListSize = arrayList.size();  
 int hashMapSize = hashMap.size();  
 return String.*format*("ArrayList size: %d\nHashMap size: %d", arrayListSize, hashMapSize);  
 }  
}

1 Создайте GET-метод updateArrayList(String s), который по url “/update-array” принимает аргумент и записывает его в ArrayList<String >, реализуйте логику - если это первый вызов метода, то создается пустой ArrayList, если он не пустой, то туда записывается значение.

@GetMapping("/update-array")  
public void updateArrayList(@RequestParam(value = "values") String values) {  
 arrayList.add(values);  
}

2 Создайте GET метод showArrayList(), который по url “/show-array” возвращается все элементы хранящиеся в ArrayList <String >, созданном в п. 1

@GetMapping("/show-array")  
public ArrayList<String> showArrayList() {  
 return arrayList;  
}

3 Создайте GET-метод updateHashMap(String s), который по url “/update-map” принимает аргумент и записывает его в HashMap< Integer, String>, еализуйте логику - если это первый вызов метода, то создается записывается значение.

@GetMapping("/update-map")  
public void updateHashMap(@RequestParam(value = "values") String values) {  
 hashMap.put(key++, values);  
}

4 Создайте GET метод showHashMap (), который по url “/show-map” возвращается все элементы хранящиеся в HashMap<Integer, String>, созданном в п. 3

@GetMapping("/show-map")  
public HashMap<Integer, String> showHashMap() {  
 return hashMap;  
}

5 Создайте GET метод showAllLenght (), который по url “/show-all-lenght” возвращает текст, в котором указано количество элементов в ArrayList и HashMap

@GetMapping("/show-all-lenght")  
public String showAllLength() {  
 int arrayListSize = arrayList.size();  
 int hashMapSize = hashMap.size();  
 return String.*format*("ArrayList size: %d\nHashMap size: %d", arrayListSize, hashMapSize);  
}

**3. Ответы на вопросы**

2 вариант

1. Чем отличаются системы сборки maven и gradle?

* Язык описания проекта: Maven использует XML для описания проекта, в то время как Gradle использует Groovy или Kotlin.
* Гибкость: Gradle обладает большей гибкостью и позволяет более сложные конфигурации, в то время как Maven имеет более строгую структуру и меньше возможностей для настройки.
* Производительность: Gradle обычно считается более быстрым и эффективным в сравнении с Maven.
* Плагины: В Maven плагины являются основным способом расширения функциональности, в то время как в Gradle они используются в меньшей степени благодаря встроенной поддержке скриптов.
* Конфигурация: Gradle позволяет более легко управлять зависимостями и версиями, в то время как в Maven это может быть более сложным и ограниченным.

3. Чем отличаются языки программирования Java и Groovy?

* Groovy имеет более компактный и выразительный синтаксис, поддерживает динамическую типизацию и другие современные функции.
* Groovy поддерживает как статическую, так и динамическую типизацию, в то время как Java использует только статическую типизацию.
* Groovy легко интегрируется с Java кодом и полностью совместим с библиотеками и фреймворками Java.
* Java обычно считается более производительным, чем Groovy, из-за дополнительных слоев абстракции и динамической типизации в Groovy.

5. Что нового добавилось в Java 11 в отличии от Java 8?

* Локальные переменные типа var: В Java 11 была добавлена возможность использовать ключевое слово var для объявления локальных переменных без явного указания типа.
* HTTP клиент: В Java 11 был добавлен стандартный модуль java.net.http, который предоставляет поддержку для выполнения HTTP запросов и обработки ответов.
* Улучшения в строковых методах: Java 11 включает новые методы для обработки строк, такие как strip(), stripLeading() и stripTrailing(), которые удаляют пробелы из начала и конца строки.
* Улучшения в модульной системе: Java 11 предоставляет улучшения в модульной системе, включая новые методы для работы с модулями и управления модульными путями.
* Расширенная поддержка TLS: Java 11 включает обновленную поддержку протокола TLS, включая поддержку более новых версий протокола и алгоритмов шифрования.

**Ссылка на репозиторий git:** https://github.com/kuk86kuk/java\_ufru\_2024