

Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут ім. Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки

## ЛАБОРАТОРНА РОБОТА № 1

з дисципліни «Розроблення застосунків з використанням Spring  
Framework»  
на тему «Перше знайомство з Spring Framework»

ВИКОНАВ:  
студент III курсу ФІОТ  
групи ІО-25  
Мамченко Д.О.  
Залікова №2518  
Скоробогатов І.В.  
Залікова №2527

Київ – 2024

## **Тема:** Перше знайомство з Spring Framework.

### **Код програми:**

#### **Lab1Application.java (1-й проєкт)**

```
package ua.kpi.lab1;

import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
@Order(2)
public class Lab1Application implements CommandLineRunner {

    public static void main(String[] args) {
        SpringApplication.run(Lab1Application.class, args);
    }

    @Override
    public void run(String... args) throws Exception {
        System.out.println("Hello from Spring Boot!");
    }
}
```

#### **First.java (1-й проєкт)**

```
package ua.kpi.lab1_1;

import org.springframework.boot.CommandLineRunner;
import org.springframework.core.annotation.Order;
import org.springframework.stereotype.Component;

@Component
@Order(1)
public class First implements CommandLineRunner {

    @Override
    public void run(String... args) throws Exception {
        System.out.println("First");
    }
}
```

## Second.java (1-й проєкт)

```
package ua.kpi.lab1_1;

import org.springframework.boot.CommandLineRunner;
import org.springframework.core.annotation.Order;
import org.springframework.stereotype.Component;

@Component
@Order(3)
public class Second implements CommandLineRunner {

    @Override
    public void run(String... args) throws Exception {
        System.out.println("Second");
    }
}
```

## MainController.java (2-й проєкт)

```
package ua.kpi.lab1;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class MainController {

    @GetMapping("/")
    public String helloWorld() {
        return "Hello, world!";
    }
}
```

## Опис:

Задля забезпечення порядку виведення рядків у консоль (6-е контрольне питання) було використано анотацію **@Order**. Вивід у консоль забезпечується методом run з інтерфейсу CommandLineRunner.

Було створено контролер в другому проєкті з єдиним ендпоінтом “/”. Також було добавлено сторінку credentials.html з даними учасників команди (7-е контрольне питання).

## Результат виконання роботи:

Begin of main

```

      / \  / --- \  / --- \  / --- \  / --- \  / --- \
      ( ) _ _ _ | ' _ | ' _ | ' _ | ' _ | ' _ | ' _ | ' _ |
      \ / _ _ _ | | | | | | | | | | | | | | | | | | | | | |
      ' _ _ _ | _ _ | _ _ | _ _ | _ _ | _ _ | _ _ | _ _ |
      =====|_|=====|_|=====|_|=====|_|=====|_|=====

```

```
:: Spring Boot :: (v3.3.3)
```

```

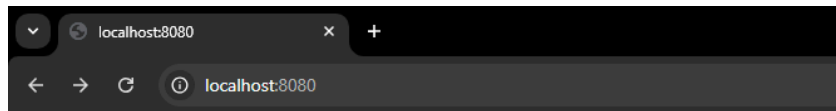
2024-09-16T20:58:52.422+02:00 INFO 21828 --- [lab1_1] restartedMain ua.kpi.lab1_1.Lab1Application : Starting Lab1Application using Java 17.0.6 with PID 21828 (C:\KPIfiles\sem5\Spring\lab1_1\target\classes
2024-09-16T20:58:52.429+02:00 INFO 21828 --- [lab1_1] restartedMain ua.kpi.lab1_1.Lab1Application : No active profile set, falling back to 1 default profile: "default"
2024-09-16T20:58:52.471+02:00 INFO 21828 --- [lab1_1] restartedMain .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.add-properties' to 'false' to disable
2024-09-16T20:58:52.471+02:00 INFO 21828 --- [lab1_1] restartedMain .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the 'logging.level.web' property to 'DEBUG'
2024-09-16T20:58:53.179+02:00 INFO 21828 --- [lab1_1] restartedMain o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
2024-09-16T20:58:53.190+02:00 INFO 21828 --- [lab1_1] restartedMain o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-09-16T20:58:53.190+02:00 INFO 21828 --- [lab1_1] restartedMain o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.28]
2024-09-16T20:58:53.230+02:00 INFO 21828 --- [lab1_1] restartedMain o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2024-09-16T20:58:53.230+02:00 INFO 21828 --- [lab1_1] restartedMain w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 759 ms
2024-09-16T20:58:53.528+02:00 INFO 21828 --- [lab1_1] restartedMain o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2024-09-16T20:58:53.555+02:00 INFO 21828 --- [lab1_1] restartedMain o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path '/'
2024-09-16T20:58:53.565+02:00 INFO 21828 --- [lab1_1] restartedMain ua.kpi.lab1_1.Lab1Application : Started Lab1Application in 1.472 seconds (process running for 1.982)

First
Hello from Spring Boot!

Second

End of main

```



Hello, world!



## Our Team

Мамченко Дмитро Олександрович

Telegram: @Dimitri0808

Номер телефону: +380 95 034 9843

Пошта: [mamchenko2210@gmail.com](mailto:mamchenko2210@gmail.com)

**Скоробогатов Ігор Віталійович**

Telegram: @thundergodswrath

Номер телефону: +380 50 421 0781

Пошта: igorskor0912@gmail.com

## Відповіді на контрольні питання:

1. **Spring Framework** – це великий фреймворк для створення Java-додатків, яка забезпечує розробку на основі принципів Inversion of Control (IoC), Dependency Injection (DI) та аспектно-орієнтованого програмування (AOP). Проте для повноцінного використання Spring потрібно налаштувати багато конфігурацій вручну (XML, Java-конфігурації).

**Spring Boot** – це надбудова над Spring Framework, яка спрощує розробку додатків, надаючи попередньо сконфігуровані компоненти та автоматичну конфігурацію. Spring Boot дозволяє створювати готові до використання додатки з мінімальною кількістю конфігурацій. Автоматична конфігурація, вбудований сервер (наприклад Tomcat), вбудована система управління залежностями (наприклад Maven) – головні фішки Spring Boot.

2. **pom.xml** – це файл конфігурації для Maven-проектів. Maven – це інструмент для управління проектами, який використовується для компіляції, тестування та розгортання Java-додатків.

3. **@SpringBootApplication** – це складна анотація, яка об'єднує три основні анотації Spring:

*@SpringBootConfiguration* – маркує клас як джерело конфігурації для Spring Boot.

*@EnableAutoConfiguration* – включає автоматичну конфігурацію, яка намагається здогадатися та налаштувати необхідні компоненти.

*@ComponentScan* – дозволяє Spring сканувати пакет та підпакети для пошуку компонентів, які необхідно підключити до контексту.

Якщо прибрати цю анотацію, Spring Boot втратить можливість автоматично конфігурувати додаток, а також не зможе знайти компоненти, контролери, ентиті і тд у вашому пакеті, що призведе до помилок у виконанні.

4. **CommandLineRunner** – це функціональний інтерфейс у Spring Boot, який має метод *run(String... args)*. Цей метод виконується після того, як Spring Boot завершить завантаження контексту додатка. Інтерфейс

використовується для виконання додаткової ініціалізації або дій під час запуску додатка, наприклад, для попередньої обробки даних або запуску початкових задач.

5. Основна різниця між **CommandLineRunner** та **ApplicationRunner** полягає в методі run. **CommandLineRunner** приймає масив `String... args` як аргументи. **ApplicationRunner** використовує інтерфейс **ApplicationArguments**, який дає змогу легше обробляти аргументи командного рядка.