

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и информационных технологий
институт
Кафедра «Информатика»
кафедра

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №3

Конфигурация Spring при помощи аннотаций
тема

Преподаватель

подпись, дата

А.С. Черниговский

ициалы, фамилия

Студент КИ18-16б 031831229

номер группы, зачетной книжки

подпись, дата

В.А. Прекель

ициалы, фамилия

Красноярск 2020

1 Цель работы

Познакомиться с аннотациями в Spring

2 Общая постановка задачи

Взять практическое задание №2 и сконфигурировать контекст приложения без использования xml, а только при помощи аннотаций и java-файла конфигурации. Снабдить ранее спроектированные классы init и destroy методами, а также использовать фабричный метод для любого из классов.

3 Исходный код

Листинг

1

```
Lab03\src\main\java\com\github\prekel\JavaSpring\Lab03\components\Cpu.java
package com.github.prekel.JavaSpring.Lab03.components;

public interface Cpu {
    int getClock();
}
```

Листинг

2

```
Lab03\src\main\java\com\github\prekel\JavaSpring\Lab03\components\LockedCpu.java
package com.github.prekel.JavaSpring.Lab03.components;

public class LockedCpu implements Cpu {
    private final int clock;

    @Override
    public String toString() {
        return "LockedCpu{" +
            "clock=" + clock +
            '}';
    }

    public LockedCpu(int clock) {
        this.clock = clock;
    }
}
```

2

```

    @Override
    public int getClock() {
        return clock;
    }
}

```

Листинг

3

Lab03\src\main\java\com\github\prekel\JavaSpring\Lab03\components\PersonalComputer.java

```

package com.github.prekel.JavaSpring.Lab03.components;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.context.annotation.Bean;
import org.springframework.stereotype.Component;

@Component("pcAsComponent")
public class PersonalComputer {
    private final Cpu cpu;
    private String name = "Default PC name";

    @Autowired
    public PersonalComputer(@Qualifier("cpu1") Cpu cpu) {
        this.cpu = cpu;
    }

    public PersonalComputer(Cpu cpu, String name) {
        this.cpu = cpu;
        this.name = name;
    }

    @Bean("pcFromFabricMethod")
    public static PersonalComputer pcFromFabricMethod(@Qualifier("cpu4") Cpu
cpu) {
        return new PersonalComputer(cpu, "Pc from fabric method");
    }

    public void turnOn() {
        System.out.println("P' PeP»CHC#PuPS " + toString());
    }

    @Override
    public String toString() {
        return "PersonalComputer{" +
            "cpu=" + cpu +
            ", name='" + name + '\'' +
            '}';
    }

    public void turnOff() {
        System.out.println("P' C< PeP»CHC#PuPS " + toString());
    }

    public Cpu getCpu() {

```

```

        return cpu;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}

```

Листинг 4 – Lab03\components\UnlockedCpu.java

```

package com.github.prekel.JavaSpring.Lab03.components;

public class UnlockedCpu implements Cpu {
    private final int busClock;
    private int multiplier;

    @Override
    public String toString() {
        return "UnlockedCpu{" +
            "busClock=" + busClock +
            ", multiplier=" + multiplier +
            ", clock=" + getClock() +
            '}';
    }

    public UnlockedCpu(int busClock, int multiplier) {
        this.busClock = busClock;
        this.multiplier = multiplier;
    }

    public int getMultiplier() {
        return multiplier;
    }

    public void setMultiplier(int multiplier) {
        this.multiplier = multiplier;
    }

    @Override
    public int getClock() {
        return busClock * multiplier;
    }

    public int getBusClock() {
        return busClock;
    }
}

```

Листинг 5 – Lab03\src\main\java\com\github\prekel\JavaSpring\Lab03\Config.java

```

package com.github.prekel.JavaSpring.Lab03;

import com.github.prekel.JavaSpring.Lab03.components.Cpu;
import com.github.prekel.JavaSpring.Lab03.components.LockedCpu;
import com.github.prekel.JavaSpring.Lab03.components.PersonalComputer;
import com.github.prekel.JavaSpring.Lab03.components.UnlockedCpu;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.PropertySource;

@Configuration
@PropertySource("classpath:pc4.properties")
@ComponentScan(basePackages = "com.github.prekel.JavaSpring.Lab03.components")
public class Config {
    @Value("${pc4.name}")
    private String volume;

    @Bean(value = "pc1", initMethod = "turnOn", destroyMethod = "turnOff")
    public PersonalComputer pc1() {
        return new PersonalComputer(cpu1(), "Pc with cpu1");
    }

    @Bean("cpu1")
    public Cpu cpu1() {
        return new LockedCpu(3600);
    }

    @Bean("pc2")
    public PersonalComputer pc2() {
        return new PersonalComputer(cpu2(), "Pc with cpu2");
    }

    @Bean("cpu2")
    public Cpu cpu2() {
        return new LockedCpu(2600);
    }

    @Bean("pc3")
    public PersonalComputer pc3() {
        var ret = new PersonalComputer(cpu3());
        ret.setName("Pc with cpu3");
        return ret;
    }

    @Bean("cpu3")
    public Cpu cpu3() {
        return new UnlockedCpu(100, 40);
    }

    @Bean("pc4")
    public PersonalComputer pc4() {
        var ret = new PersonalComputer(cpu4());
        ret.setName(volume);
        return ret;
    }

    @Bean("cpu4")
    public Cpu cpu4() {
        return new UnlockedCpu(110, 20);
    }
}

```

```
}
```

Листинг 6 – Lab03\src\main\java\com\github\prekel\JavaSpring\Lab03\Program.java

```
package com.github.prekel.JavaSpring.Lab03;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import
org.springframework.context.annotation.AnnotationConfigApplicationContext;

import com.github.prekel.JavaSpring.Lab03.components.PersonalComputer;

@SpringBootApplication
public class Program {
    public static void main(String[] args) {
        SpringApplication.run(Program.class, args);

        AnnotationConfigApplicationContext context = new
AnnotationConfigApplicationContext(Config.class);
        var config = context.getBean("config", Config.class);

        var pc1 = config.pc1();
        var pc2 = config.pc2();
        var pc3 = config.pc3();
        var pc4 = context.getBean("pc4", PersonalComputer.class);
        var pc = context.getBean("pcAsComponent", PersonalComputer.class);
        var pcf = context.getBean("pcFromFabricMethod", PersonalComputer.class);
        System.out.println(pc1);
        System.out.println(pc2);
        System.out.println(pc3);
        System.out.println(pc4);
        System.out.println(pc);
        System.out.println(pcf);

        System.out.println();

        for (String beanName : context.getBeanDefinitionNames()) {
            System.out.println(beanName);
        }

        context.close();
    }
}
```

Листинг 7 – Lab03\src\main\resources\pc4.properties

```
pc4.name=Pc with cpu4
```