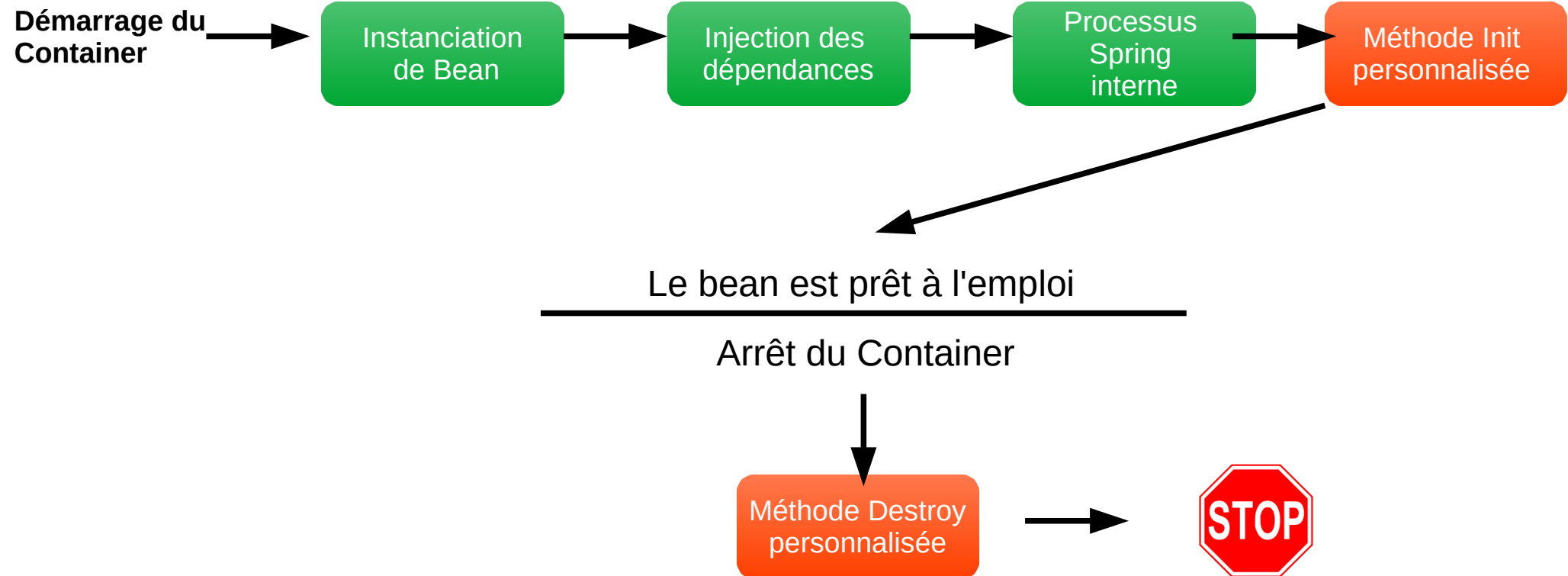




Spring core - annotations

Bean - LifeCycle

Rappel Lifecycle - diagram



Init & destroy méthodes

- On va pouvoir intervenir sur l'instanciation d'un bean en annotant une méthode de la classe avec l'annotation `@PostConstruct`, ainsi cette méthode sera appelée lors de la phase d'instanciation
- De la même manière on va pouvoir intervenir sur la destruction d'un bean en annotant une méthode avec `@PreDestroy()`

Démo - TD

- NB PostConstruct = le code va s'exécuter après le constructeur et après l'injection de dépendance
- NB : PreDestroy = le code va s'exécuter avant la suppression du bean
- Nb : return void en général
- Nb : aucun argument autorisé pour ces méthodes annotées avec @PostConstruct et @PreDestroy

Remarque java ≥ 9

- Ces annotations sont introuvables à partir de java 9
- Vous devez télécharger un jar supplémentaire
javax.annotation-api-1.3.2.jar
- (ajoutez la dans le répertoire lib et au classpath de votre application)

TD

```
@Component
public class Trompetiste implements Musicien {

    private PrepareService prepareService;

    @PostConstruct
    public void myInitMethod() {
        System.out.println(" Trompetiste : à l'intérieur de postConstruct() myInitMethod() est appelée ");
    }

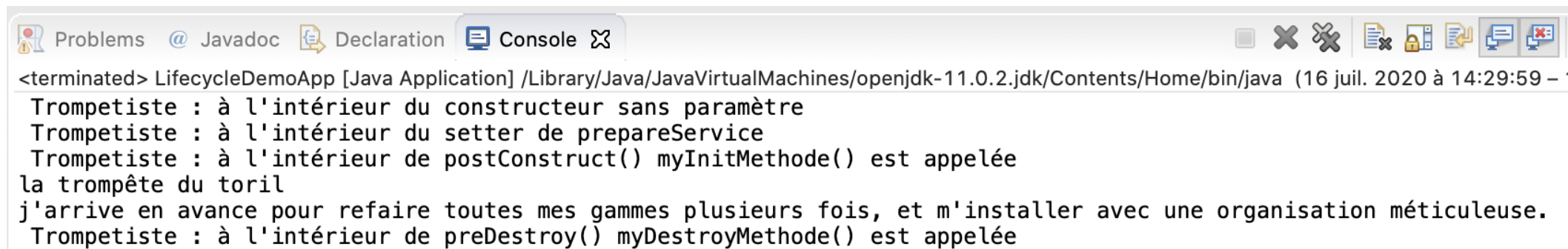
    @PreDestroy
    public void myDestroyMethod() {
        System.out.println(" Trompetiste : à l'intérieur de preDestroy() myDestroyMethod() est appelée ");
    }

    public PrepareService getPrepareService() {
        System.out.println(" Trompetiste : à l'intérieur du getter de prepareService");
        return prepareService;
    }

    @Autowired
    public void setPrepareService(PrepareService prepareService) {
        System.out.println(" Trompetiste : à l'intérieur du setter de prepareService");
        this.prepareService = prepareService;
    }

    public Trompetiste() {
        System.out.println(" Trompetiste : à l'intérieur du constructeur sans paramètre");
    }
}
```

Run as>Java application



The screenshot shows an IDE console window with the following content:

```
<terminated> LifecycleDemoApp [Java Application] /Library/Java/JavaVirtualMachines/openjdk-11.0.2.jdk/Contents/Home/bin/java (16 juil. 2020 à 14:29:59 - 1
Trompetiste : à l'intérieur du constructeur sans paramètre
Trompetiste : à l'intérieur du setter de prepareService
Trompetiste : à l'intérieur de postConstruct() myInitMethode() est appelée
la trompète du toril
j'arrive en avance pour refaire toutes mes gammes plusieurs fois, et m'installer avec une organisation méticuleuse.
Trompetiste : à l'intérieur de preDestroy() myDestroyMethode() est appelée
```

NB : `scope("prototype")`

- Spring ne gèrera pas le `preDestroy` pour le `scope prototype`.
-
- C'est à vous de réaliser les opérations dans votre code concernant les objet "scopés" en `prototype` pour libérer les ressources affectées.