*public interface MyService* {

 String getType();

 *void* checkStatus();

 *void* inc();

 Integer getCount();
}

*public abstract class* MyServiceCommon *implements MyService* {

 *private* Integer count = 0;

 @Override
 *public void* inc() {
 count++;
 }

 @Override
 *public* Integer getCount() {
 System.*out*.println("MVS : " + count);
 *return* count;
 }

 @Override
 *public void* checkStatus() {
 *// Your code* }

}

@Component
*public class* MyServiceOne *extends* MyServiceCommon {

 @Override
 *public* String getType() {
 *return* "one";
 }

}

@Component
*public class* MyServiceTwo *extends* MyServiceCommon {
 @Override
 *public* String getType() {
 *return* "two";
 }

}

@Service
*public class* MyServiceFactory {

 @Autowired
 *private List*<*MyService*> services;

 *private static final Map*<String, *MyService*> *myServiceCache* = *new* HashMap<>();

 @PostConstruct
 *public void* initMyServiceCache() {
 *for*(*MyService* service : services) {
 *myServiceCache*.put(service.getType(), service);
 }
 }

 *public MyService* getService(String type) {
 *MyService* service = *myServiceCache*.get(type);
 *if*(service == *null*) *throw new* RuntimeException("Unknown service type: " + type);
 *return* service;
 }
}

@Component
*public class* TestFactory2 *implements CommandLineRunner* {

 @Autowired
 MyServiceFactory myServiceFactory;

 @Override
 *public void* run(String... args) *throws* Exception {
 *MyService* myServiceOne = myServiceFactory.getService("one");
 myServiceOne.checkStatus();;
 *MyService* myServiceTwo = myServiceFactory.getService("two");
 myServiceTwo.checkStatus();
 *MyService* myServiceThree = myServiceFactory.getService("one");
 myServiceOne.inc();
 myServiceOne.getCount();
 myServiceThree.getCount();

 }
}

MVS : 1

MVS : 1