



## Ćwiczenie 1:

Za pomocą komponentu Library, przy użyciu metody `registerNewUser` można rejestrować nowych użytkowników. Metoda korzysta z serwisu `StudentListService` aby zweryfikować istnienie studenta o podanym nazwisku. Za pomocą aspektów dodaj kod audytujący każde wywołania tej metody, logując w konsoli systemowej parametry wejściowe metody oraz czy wykonanie zakończyło się sukcesem.

## Instrukcja rozwiązania

1. Dołącz do projektu następujące biblioteki: `spring-aop`, `aopalliance`, `aspectjrt`, `aspectjweaver`, `cglib`
2. Utwórz nową poradę typu `Around Advice`, która wyświetli wartość parametru oraz wyświetli status zakończenia działania metody.
3. Zmodyfikuj plik konfiguracyjny `context.xml` tak aby aspekty zostały automatycznie dołączone do projektu.

## Rozwiązanie

```
@Aspect
@Component
public class AuditingAdvice {

    @Around("execution(* pl.devcastzone.spring.Library.registerNewUser(..))")
    public Object audit(ProceedingJoinPoint joinPoint) throws Throwable {

        StringBuffer auditLog = new StringBuffer();
        auditLog.append("Library.registerNewUser");
        auditLog.append(" (name=")
                .append(joinPoint.getArgs()[0]).append(")");

        Object proceed;
        try {
            proceed = joinPoint.proceed();
        } catch (Throwable e) {
            auditLog.append(" => user not registered");
            System.out.println(auditLog.toString());
            throw e;
        }

        auditLog.append(" => user registered");
        System.out.println(auditLog.toString());
        return proceed;
    }
}
```

```
<aop:aspectj-autoproxy/>
<context:annotation-config />
```