



## OPIS ZADANIA

Jest zaprojektowanie oraz wykonanie usługi sieciowej realizowanej przez komponent o nazwie Control. Komponent może znajdować się w jednym z dwóch stanów : zliczania lub wstrzymania. Początkowo komponent znajduje się w stanie wstrzymania a licznik komponentu jest wyzerowany. Przejście ze stanu wstrzymania do stanu zliczania jest możliwe wyłącznie po zakończonej sukcesem rejestracji i następuje w efekcie wywołania polecenia start. Przejście ze stanu zliczania do stanu wstrzymania następuje w efekcie wywołania polecenia stop. Wykonanie polecenia incr powoduje zmianę wartości licznika o 1, polecenia incr/n powoduje zmianę wartości licznika o wartość n, gdzie n jest liczbą naturalną. Wykonanie polecenia zmieniającego stan licznika dopuszczalne jest wyłącznie w stanie zliczania. Wykonanie metody incr/n lub decr/n w stanie wstrzymania nie zmienia wartości licznika, powoduje natomiast zarejestrowanie faktu niepoprawnego żądania poprzez zwiększenie licznika błędów (error).

## STRUKTURY DANYCH

W skład projektu wchodzi klasa `Control.java` oraz interfejs `IDbManager.java`

```
Control.java
void registerWork()
String start()
String stop()
String counter()
String errors()
String increment()
String increment(String number)
String decrement()
String decrement(String number)
```

Metoda `registerWork`, jej zadaniem jest zarejestrowanie zadania z użyciem metody `register` komponentu `DbManager`. Kolejne publiczne metody odpowiedzialne są za obsługę żądań:

Metoda	Adres	Opis	Żądanie
<code>start()</code>	<code>/cnt/start</code>	start procesu zliczania	GET
<code>stop()</code>	<code>/cnt/stop</code>	zakończenie procesu zliczania	GET
<code>counter()</code>	<code>/cnt/counter</code>	stan licznika	GET
<code>errors()</code>	<code>/cnt/errors</code>	stan licznika błędów	GET
<code>increment()</code>	<code>/cnt/incr</code>	zwiększenie o 1 licznika	GET
<code>increment(number)</code>	<code>/cnt/incr/n</code>	zwiększenie o liczbę n licznika	GET
<code>decrement()</code>	<code>/cnt/decr</code>	zmniejszenie o 1 licznika	GET
<code>decrement(number)</code>	<code>/cnt/decr/n</code>	zmniejszenie o liczbę n licznika	GET

`IDbManager.java` – Interfejs zawierający deklaracje metody `register`  
`boolean register(int hwork, String album)`



## OPIS MECHANIZMU WYSZUKIWANIA

Do zlokalizowania klasy DbManager oraz nawiązania z nią połączenia wykorzystany został interfejs JNDI (ang. Java Naming and Directory Interface). Pozwala on klientom na odkrywanie oraz wyszukiwania danych i obiektów na podstawie nazw.

Aby nawiązać połączenie należy utworzyć obiekt InitialContext:

```
InitialContext ctx = new InitialContext();
```

Następnie należy skorzystać z metody lookup, której implementacja umożliwia odnalezienie zasobu po nazwie:

```
IDbManager obj = (IDbManager) ctx.lookup("java:global/ejb-project/  
DbManager!pl.jrj.db.IDbManager");
```

Ścieżka przestrzeni nazw ma postać:

```
java:global/ejb-project/DbManager!pl.jrj.db.IDbManager
```

## ALGORYTM

Do realizacji zadania wykorzystana została biblioteka Jersey 2. Żądania protokołu HTTP obsługiwane są poprzez adnotowane metody klasy Control. Każde z żądań opisane zostało powyżej w sekcji struktury danych. Są to żądania GET protokołu HTTP.

Algorytm programu opiera się na zliczaniu żądań w zmiennej processCounter. Aby proces zliczania mógł się rozpocząć zmienna isStarting musi zostać zainicjowana wartością true, w przeciwnym wypadku zliczane będą błędy w zmiennej errorsCounter. Kolejno możliwe są operacje inkrementacji licznika o wartość 1 lub wartość n, gdzie n jest liczbą naturalną. Możliwe są również operacje dekrementacji o wartość 1 lub wartość n, gdzie n jest liczbą naturalną. W każdym momencie działania aplikacji możliwe jest wyświetlenie zawartości licznika wywołań oraz licznika błędów.

Wszystkie zmienne przechowywane są w sesji przeglądarki. Dzięki zastosowaniu takiego rozwiązania możliwe jest niezależne uruchamianie aplikacji za pośrednictwem różnych przeglądarek.

## URUCHOMIENIE

Proces kompilacji jest możliwy z użyciem komendy:

```
javac -cp <app-server-modules> -Xlint Control.java IDbManager.java  
*.java
```