

Programowanie portali biznesowych

WYKŁAD 6

Enterprise JavaBeans

Specyfikacja JSR (Java Specification Request) 345 dla EJB:

<https://jcp.org/en/jsr/detail?id=345>

Warstwy aplikacji Java EE

**Warstwa kliencka
(przeglądarka, aplikacja)**

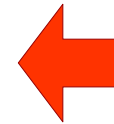
**Warstwa webowa
(serwlety, JSP, JSF)**

**Warstwa biznesowa
(EJB)**

Warstwa danych

**Stosowana
w rozbudowanych
aplikacjach.**

**W prostych
aplikacjach jej rolę
przejmuje warstwa
webowa.**



Enterprise JavaBeans

- Komponenty Enterprise JavaBeans (EJB) umożliwiają tworzenie aplikacji rozproszonych na bazie komponentów oferujących:
 - skalowalność,
 - przetwarzanie transakcyjne,
 - bezpieczeństwo.
- Komponenty EJB działają w kontenerze EJB, który zajmuje się ich tworzeniem, usuwaniem i przypisywaniem do żądań klientów.
- W komponentach EJB implementowana jest logika biznesowa aplikacji.

Enterprise JavaBeans

- Klientami komponentów EJB mogą być:
 - serwlety,
 - komponenty zarządzane,
 - komponenty CDI,
 - JSP,
 - JSF.

Enterprise JavaBeans

- Rodzaje komponentów EJB:
 - komponenty sesyjne (*Session Beans*) – obsługują żądania klientów,
 - komponenty sterowane komunikatami (*Message-driven Beans*),
 - komponenty encyjne (*Entity Beans*) – zastąpione przez Java Persistence API.

Komponenty sesyjne EJB

- Stanowy komponent sesyjny:
 - przechowuje stan konwersacji z określonym klientem,
 - oznaczany jest adnotacją **@Stateful**

Komponenty sesyjne EJB

- Stanowy komponent sesyjny – przykład:

```
@Stateful
```

```
public class Koszyk
```

```
{
```

```
    private Towar[] towary;
```

```
    public Koszyk() {}
```

```
    public Towar[] getTowary() { return towary; }
```

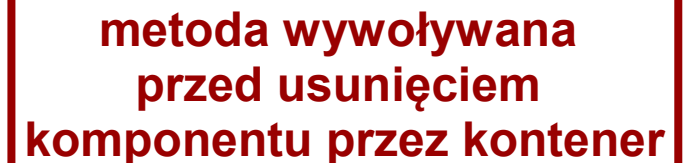
```
    public void setTowary(Towar[] towary)
```

```
        { this.towary=towary; }
```

```
@Remove
```

```
public void remove() { /... }
```

```
}
```



metoda wywoływana
przed usunięciem
komponentu przez kontener

Komponenty sesyjne EJB

- Stanowy komponent sesyjny - przykład:
 - dostęp dla klienta:

@EJB Koszyk koszyk;

Komponenty sesyjne EJB

- Bezstanowy komponent sesyjny:
 - nie zawiera żadnej informacji o stanie knowersacji z określonym klientem,
 - oznaczany jest adnotacją **@Stateless**

Komponenty sesyjne EJB

- Bezstanowy komponent sesyjny – przykład:

```
@Stateless
```

```
public class waluta
```

```
{
```

```
    private double kurs;
```

```
    public waluta() {}
```

```
    public double getKurs() { return kurs; }
```

```
    public void setKurs(double kurs)
```

```
        { this.kurs=kurs; }
```

```
}
```

Komponenty sesyjne EJB

- Bezstanowy komponent sesyjny - przykład:
 - dostęp dla klienta:

@EJB Bank bank;

Komponenty sesyjne EJB

- Singletonowy komponent sesyjny:
 - tworzona jest tylko jedna jego instancja w danej aplikacji,
 - oznaczany jest adnotacją **@Singleton**

Komponenty sesyjne EJB

- Bezstanowy komponent sesyjny – przykład:

```
@Singleton
```

```
public class Zarzadca
```

```
{
```

```
    //...
```

```
}
```

Komponenty EJB sterowane komunikatami

- Komponent sterowany komunikatami używany jest do asynchronicznego przetwarzania komunikatów.
- Komponent sterowany komunikatami jest komponentem bezstanowym.
- Komponent sterowany komunikatami oznaczany jest adnotacją **@MessageDriven**

Typy komunikatów

- **StreamMessage**
- **MapMessage**
- **TextMessage**
- **ObjectMessage**
- **BytesMessage**

Komponenty sesyjne EJB

- Komponent sterowany komunikatami – przykład:

```
@MessageDriven
```

```
public class Zapis
```

```
{
```

```
    //...
```

```
    @Override
```

```
    public void onMessage(Message komunikat)
```

```
    {
```

```
        try { //... }
```

```
        catch(JMSEException ex) { //... }
```

```
    }
```

```
}
```



metoda wywoływana
przez kontener