

# Programowanie portali biznesowych

## WYKŁAD 7

# Java Persistence API (JPA)

**Specyfikacja JSR (Java Specification Request) 338 dla JPA:**

<https://jcp.org/en/jsr/detail?id=338>

## Java Persistence API (JPA)

- **JPA** (*Java Persistence API*) – specyfikacja definiująca interfejs API umożliwiający zarządzanie mechanizmami utrwalania i odwzorowania obiektowo-relacyjnego.
- JPA definiuje standardowe odwzorowanie tabeli bazy danych na obiekt POJO (*Plain Old Java Object*).

# Java Persistence API (JPA)

- JPA definiuje również:
  - transakcje,
  - buforowanie,
  - sprawdzanie poprawności,
  - inne mechanizmy potrzebne do obsługi operacji dostępu aplikacji do bazy danych.

## **Encja**

- **Encja** – obiekt POJO z bezargumentowym konstruktorem publicznym.
- Klasa takiego obiektu oznaczana jest adnotacją

### **@Entity**

- Encja używana jest do odwzorowania jednej lub więcej tabel w relacyjnej bazie danych.

# Encja

- Inne wybrane adnotacje:
  - **@Id** – adnotacja dla pola *id*
  - **@NotNull** – adnotacja dla pola wymaganego
  - **@Table** – adnotacja wskazująca odwzorowywaną tabelę relacyjnej bazy danych

# Encja

- Przykład:

```
@Entity
```

```
@Table(name="Miasta")
```

```
public class Miasto implements Serializable{
```

```
    @Id
```

```
    private int id;
```

```
    private String nazwa;
```

```
    public int getId() { return id; }
```

```
    public void setId(int id) { this.id=id; }
```

```
    public String getNazwa() { return nazwa; }
```

```
    public void setNazwa(String nazwa) { this.nazwa= nazwa;}
```

```
}
```

## Jednostka utrwalania

- **Jednostka utrwalania** – definiowana jest w pliku *persistence.xml*.
- Zawiera informacje o:
  - menedżerze encji, jego konfiguracji,
  - zbiorze zarządzanych encji,
  - metadanych specyfikujących odwzorowania klas na tabele relacyjnej bazy danych



## Operacje CRUD

- Operacje CRUD (*create, read, update, delete*) mogą być definiowane za pomocą:
  - języka **JPQL** (*Java Persistence Query Language*),
  - metamodelu **Criteria API**,
  - natywnego języka **SQL**.

# Operacje CRUD

- *Create*

- Przykład (użycie *EntityManager*):

```
EntityManagerFactory emf=
```

```
    Persistence.createEntityManagerFactory("ExamplePU");
```

```
EntityManager em=emf.createEntityManager();
```

```
em.getTransaction().begin();
```

```
Miasto m=new Miasto();
```

```
m.setNazwa("Rzeszów");
```

```
em.persist(m);
```

```
em.getTransaction().commit();
```

# Operacje CRUD

- *Read*

- Przykład 1 (użycie JPQL):

```
EntityManagerFactory emf=
```

```
    Persistence.createEntityManagerFactory("ExamplePU");
```

```
EntityManager em=emf.createEntityManager();
```

```
em.createQuery("SELECT m FROM Miasto m",  
              Miasto.class).getResultList();
```

# Operacje CRUD

- *Read*

- Przykład 2 (użycie JPQL):

```
EntityManagerFactory emf=
```

```
    Persistence.createEntityManagerFactory("ExamplePU");
```

```
EntityManager em=emf.createEntityManager();
```

```
em.createQuery("SELECT m FROM Miasto m WHERE m.id= :id",  
    Miasto.class).setParameter("id", 2).getResultList();
```

# Operacje CRUD

- *Update*

- Przykład:

```
EntityManagerFactory emf=
```

```
    Persistence.createEntityManagerFactory("ExamplePU");
```

```
EntityManager em=emf.createEntityManager();
```

```
Query q=em.createQuery("UPDATE Miasto m SET m.nazwa=
```

```
    :nazwa WHERE m.id= :id", Miasto.class);
```

```
q.setParameter("nazwa", "Lublin");
```

```
q.setParameter("id", "1");
```

```
q.executeUpdate();
```

# Operacje CRUD

- *Delete*

- Przykład (użycie *EntityManager*):

```
EntityManagerFactory emf=
```

```
    Persistence.createEntityManagerFactory("ExamplePU");
```

```
EntityManager em=emf.createEntityManager();
```

```
Miasto m=em.find(Miasto.class, 2);
```

```
em.remove(m);
```

# Walidacja encji

- Wybrane adnotacje:
  - @Min
  - @Max
  - @Pattern
  - @Size

# Walidacja encji

- Przykład:

```
@Entity
@Table(name="Miasta")
public class Miasto implements Serializable{
    @Id
    private int id;
    @Size(max=30)
    private String nazwa;

    //...
}
```