НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Java EE / Jakarta Persistence (JPA)


Беркунский Е.Ю., кафедра ИУСТ, НУК
eugeny.berkunsky@gmail.com
http://www.berkut.mk.ua

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Содержание

- Что такое и Почему используем O/R Mapper (ORM)?
- Что нам дает JPA?
- O/R Отображения
- Что такое Entity?
- Программная модель JPA
- EntityManager и операции управления Entity
- Отсоединенные объекты
- Жизненный цикл Entity
- Persistence контекст и EntityManager

# Почему Object/Relational Mapping?

- Одна из главных частей любого энтерпрайз приложения – уровень persistence
  - Доступ и управление перманентными данными, обычно с применением реляционной БД
- ORM берет на себя “превращение” таблицы в объект
  - Данные живут в реляционной БД, т.е. в таблицах (в строчках и столбцах)
  - Мы же хотим работать с объектами, а не с колонками и столбцами

# Что нам дает JPA?

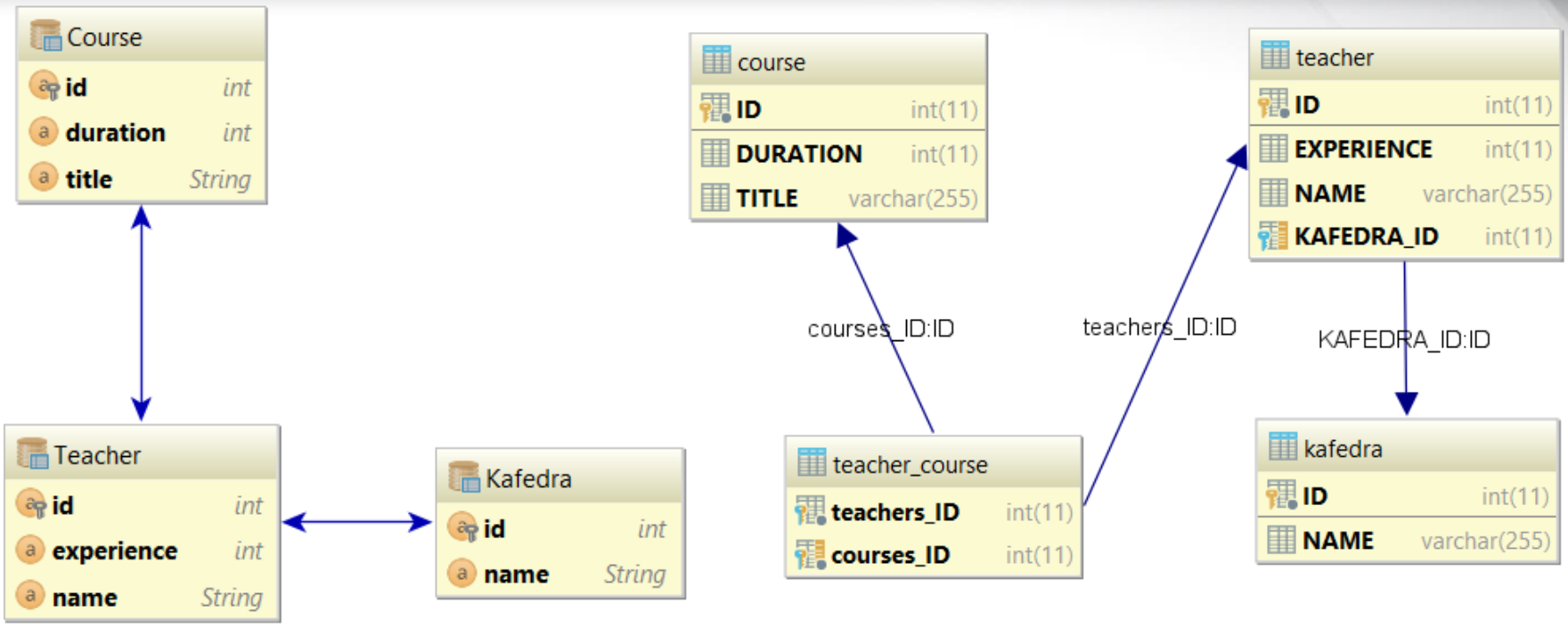
- Упрощение модели persistence
  - –Использование значений по умолчанию вместо сложных настроек
  - – Отказ от конфигурационных файлов
- Предоставление легковесной модели persistence
  - – Увеличение быстродействия
- Предоставление возможности тестировать вне контейнера
  - – TDD
- Единый API для Java SE и Java EE

# O/R Отображения

- Обширный набор аннотаций для описания отображений (mapping)
  - Связи
  - Объединения
  - Таблицы и колонки БД
  - Генераторы последовательностей для БД
  - Многое другое
- Возможно использовать отдельный конфигурационный файл для описания отображений (mapping)

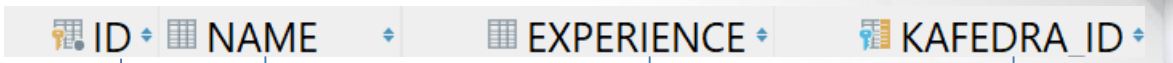
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Пример модели



НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Пример отображения




```
@Entity
public class Teacher {
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Id
    private int id;

    private String name;

    private int experience;

    @ManyToOne(optional = true)
    private Kafedra kafedra;
```

# Что такое Entity?

- Простой Java класс (Plain Old Java Object POJO)
  - Создается как обычный Java класс – при помощи new
  - Нет необходимости реализовывать интерфейсы в отличии от EJB 2.1 entity beans
- Может содержать перманентные и не перманентные данные
  - Не перманентные данные помечаются transient или @Transient
- Может расширять другие entity и не-entity классы

# Идентификация Entity

- Любой Entity имеет перманентный идентификатор
  - Он отображается в первичный ключ в таблице
- Идентификатор — примитивный тип
  - @Id—одиночное поле/свойство в Entity классе
  - @GeneratedValue—значение может генерироватся автоматически, используя различные стратегии (SEQUENCE, TABLE, IDENTITY, AUTO)
- Идентификатор – пользовательский класс
  - @EmbeddedId—одиночное поле/свойство в Entity классе
  - @IdClass—соответствует множеству полей в Entity классе

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Отображение отношений

- Аннотации для описания отношений между Entity
  - @OneToOne
  - @OneToMany
  - @ManyToOne
  - @ManyToMany

# EntityManager

- Управляет жизненным циклом Entity объектов
  - persist() - помещает объект в БД
  - remove() - удаляет объект из БД
  - merge() - синхронизирует с БД состояние отсоединенного объекта
  - refresh() - обновляет из БД состояние объекта

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Операция Persist

```
public Teacher createNewTeacher(String name, int experience) {
    // Создаем новый объект
    Teacher teacher = new Teacher(name, experience);

    // После вызова метода persist() объект меняет свой
    // статус на управляемый. Во время очередной
    // операции записи в БД объект будет помещен в БД.
    em.persist(teacher);

    return teacher;
}
```

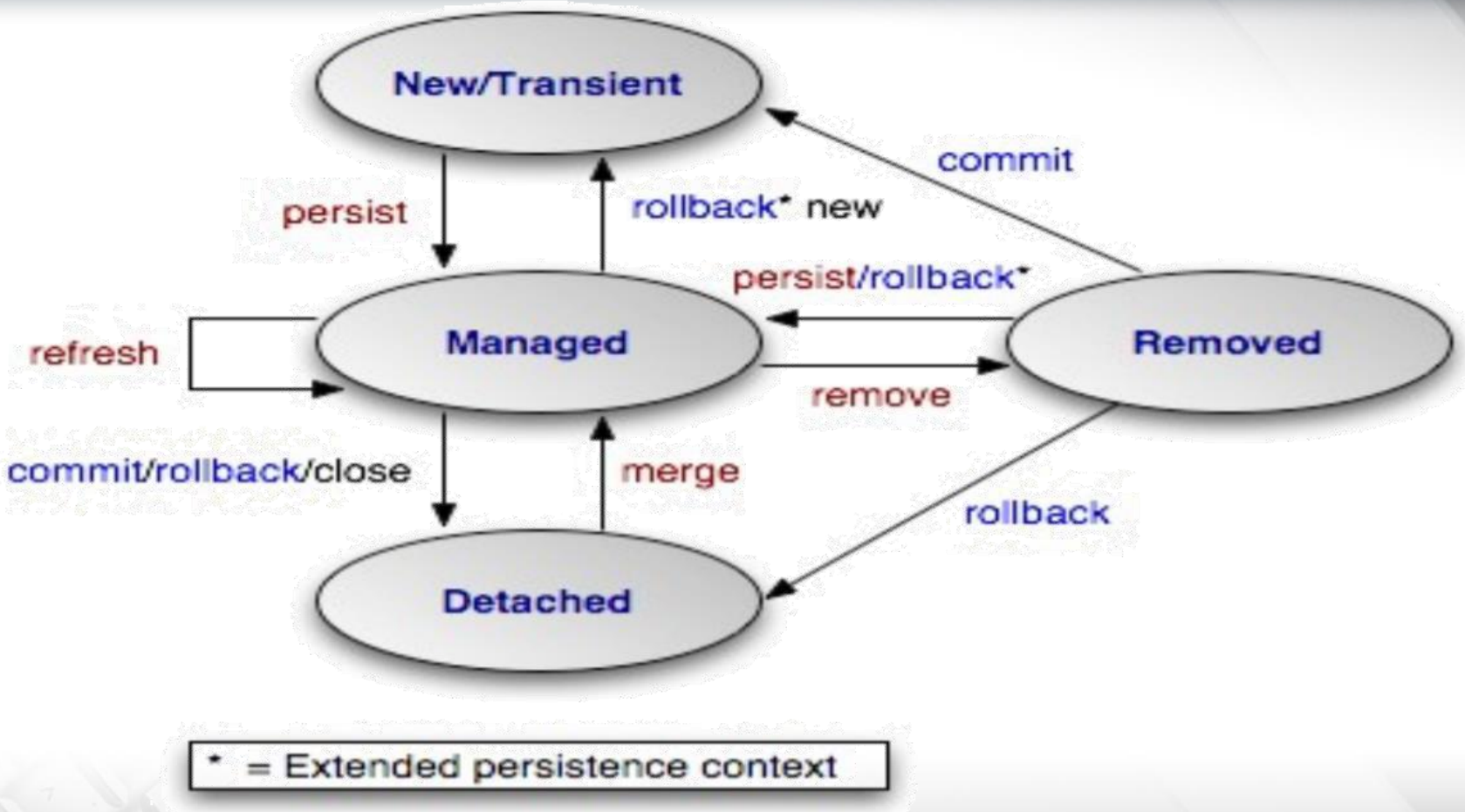
# Операции Find и Remove

```
public void removeTeacher(int teacherId) {

    Teacher teacher = em.find(Teacher.class, teacherId);

    // Объект будет удален из БД при очередной
    // операции записи в БД. Доступ к удаленному
    // объекту приводит к непредсказуемым
    // результатам
    em.remove(teacher);
}
```

# Операция Merge

```
public Teacher updateTeacher(Teacher teacher) {
    // Метод merge возвращает управляемую копию
    // переданного отсодиненного объекта. Если состояние
    // отсоединного объекта было изменено, то изменения
    // будут отражены в возвращаемой копии
    return em.merge(teacher);
}
```

# Жизненный цикл Entity



# Persistence контекст и EntityManager

- Persistence контекст
  - Множество управляемых Entity объектов во время работы приложения
  - “Объект со статусом управляемый” означает что он принадлежит определенному persistent контексту
- EntityManager
  - Выполняет операции связанные с жизненным циклом Entity объекта – управляет persistent контекстом

# Persistence контекст и EntityManager

- Persistence контекст напрямую не доступен разработчику напрямую
  - Программного доступа к Persistence контекст нет — в этом нет необходимости
  - Доступ Persistence контекст осуществляется через EntityManager
- Тип EntityManager определяет как будет persistence контекст будет создаваться и удаляться

# Типы EntityManager

- Управляемый контейнером EntityManager (Java EE)
- Управляемый приложением EntityManager (Java SE)

# Как создать EntityManager

- Разные типы EntityManager создаются по разному
  - Управляемый контейнером EntityManager (Java EE) создается контейнером и становится доступным для приложения через механизм инъекций

*Используется аннотация @PersistenceContext*

  - Управляемый приложением EntityManager (Java SE) создается и закрывается (уничтожается) приложением.

# Persistence Unit

- Все Entity объекты управляемые определенным EntityManager определяются при помощи Persistence Unit
- persistence.xml определяет один или несколько Persistence Unit

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Persistence Unit

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.1" xmlns="http://xmlns.jcp.org/xml/ns/persistence"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">
  <persistence-unit name="WebAppPU" transaction-type="JTA">
    <jta-data-source>jdbc/demojpa</jta-data-source>
    <exclude-unlisted-classes>false</exclude-unlisted-classes>
    <properties>
      <property name="javax.persistence.schema-generation.database.action"
                value="create"/>
      </properties>
  </persistence-unit>
</persistence>
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# payara-resources.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE resources PUBLIC "-//GlassFish.org//DTD GlassFish Application Server 6.2023
Resource Definitions//EN" " ">
<resources>
    <jdbc-connection-pool
         allow-non-component-callers="false" associate-with-thread="false"
         connection-creation-retry-attempts="0"
         connection-creation-retry-interval-in-seconds="10"
         connection-leak-reclaim="false" connection-leak-timeout-in-seconds="0"
         connection-validation-method="auto-commit"
         datasource-classname="org.mariadb.jdbc.MariaDbDataSource"
         fail-all-connections="false" idle-timeout-in-seconds="300"
         is-connection-validation-required="false" is-isolation-level-guaranteed="true"
         lazy-connection-association="false" lazy-connection-enlistment="false"
         match-connections="false" max-connection-usage-count="0" max-pool-size="32"
         max-wait-time-in-millis="60000"  name="nuospool"
         non-transactional-connections="false" pool-resize-quantity="2"
         res-type="javax.sql.DataSource"
         statement-timeout-in-seconds="-1" steady-pool-size="8"
         validate-atmost-once-period-in-seconds="0" wrap-jdbc-objects="false">
        <property name="User" value="student"/>
        <property name="Password" value="123"/>
        <property name="URL“
              value="jdbc:mariadb://localhost:3306/nuos?createDatabaseIfNotExist=true"/>
        <property name="driverClass" value="org.mariadb.jdbc.Driver"/>
    </jdbc-connection-pool>
    <jdbc-resource enabled="true"
              jndi-name="java:app/jdbc/nuos" object-type="user" pool-name="nuospool"/>
</resources>
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Q & A

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

# Java EE / Jakarta Persistence (JPA)


Беркунский Е.Ю., кафедра ИУСТ, НУК
eugeny.berkunsky@gmail.com
http://www.berkut.mk.ua