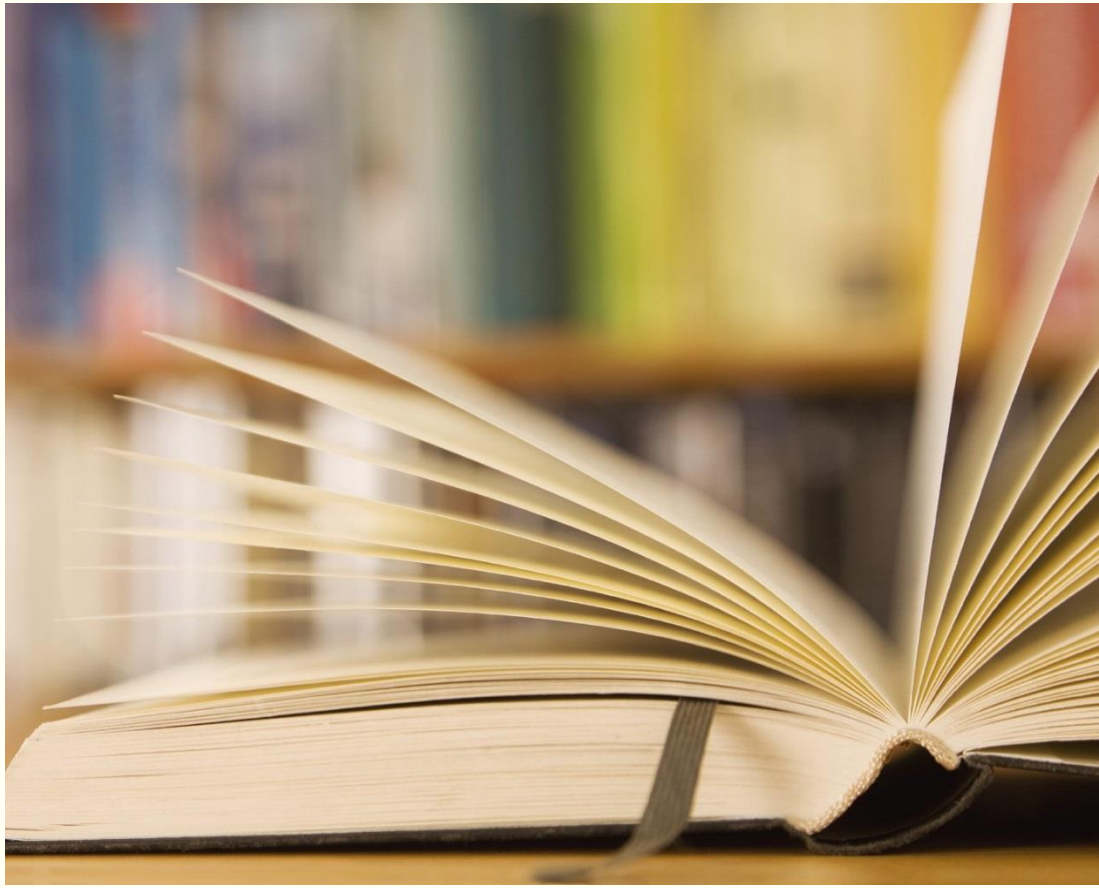
ENTERPRISE ARCHITECTURE

Najeeb Najeeb, PhD



LESSON 02 PERSISTENCE & RELATIONS

CONVENTION OVER CONFIGURATION

What is Convention over Configuration

- Configuration by Exception
- Programming by Exception
- Use Defaults
  - Configuration is the exception, not required.
- Cost
  - Defaults are not obvious sometimes.
  - Debugging may become hard, due to default behaviors.
- Defaults are not hiding things but supporting rapid development.
- Learn the defaults to help ease debugging and development.
  - It does not hurt to write comments in the beginning.

My First Entity

```
@Entity
public class Student {

        @Id private int id;
        private String name;
        private double gpa;

        public Student() {}
        public Student(int id) { this.id = id; }

        @Id public int getId() { return id; }
        public void setId(int id) { this.id = id; }
        public String getName() { return name; }
        public void setName(String name) { this.name = name; }
        public double getGpa() { return gpa; }
        public void setGpa(double gpa) { this.gpa = gpa; }

}
```

INTRODUCTION TO JPA

Entity Manager

- EntityManager is an Interface.
- Entities are managed by EntityManager.
- Entities becomes managed if
  - Read by EntityManger from DB.
  - Passed to EntityManager.
- Configured to manage specific types of objects.
- Configured to work with a given database.
- Configured to be implemented by a persistence provider.
- An EntityManager is created by an EntityManagerFactory.

EntityManagerFactory

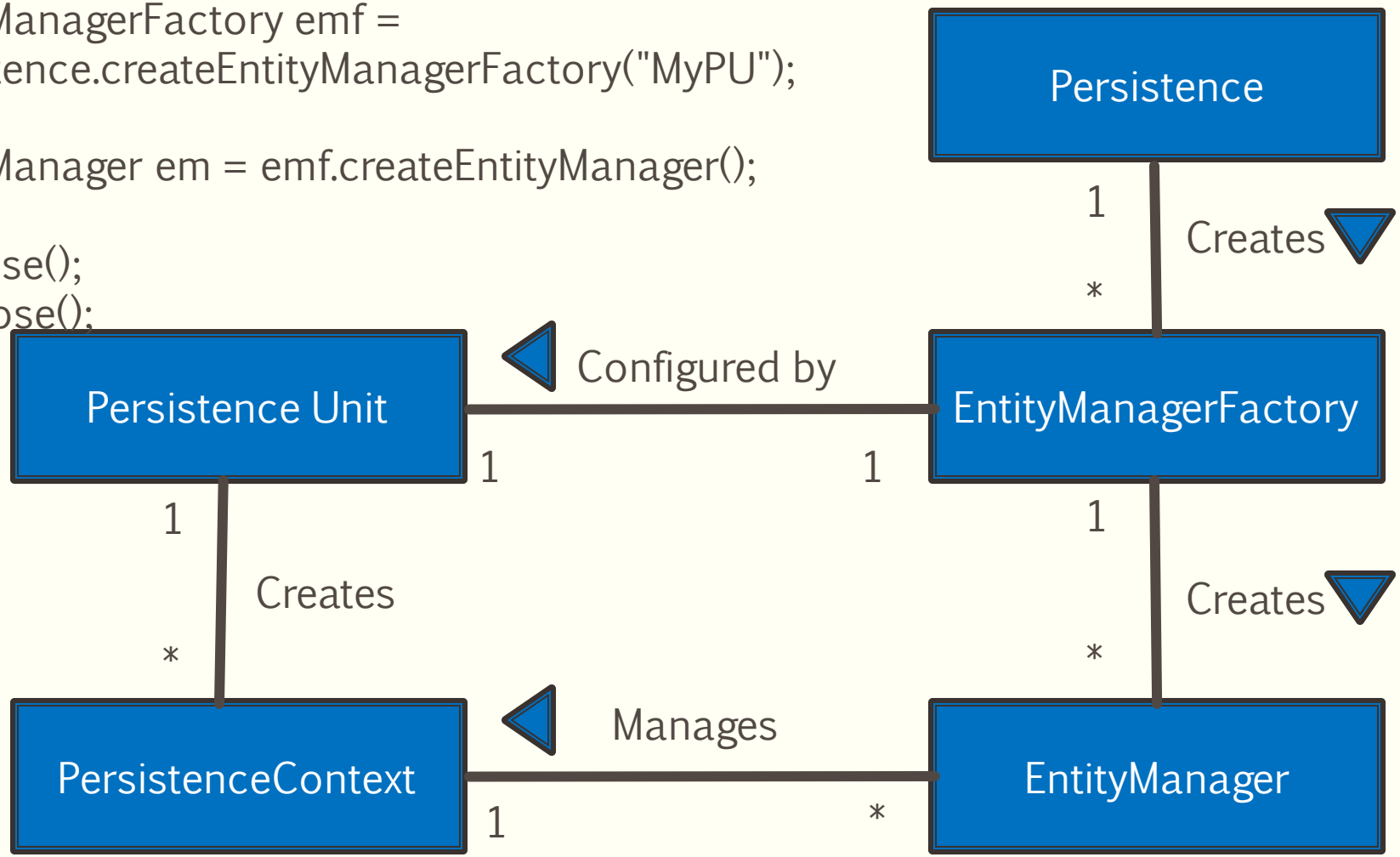
- A factory creates an EntityManager based on a persistence unit.
- Persistence Unit is the set of configurations that define the EntityManager.
- There is a one-to-one relationship between an EntityManagerFactory and a Persistence Unit.
- Persistence unit defines the database to be used, how to access it
- The factory may create several entity manager, but they will all have the same configuration.

JPA Concepts

```
EntityManagerFactory emf =
Persistence.createEntityManagerFactory("MyPU");

EntityManager em = emf.createEntityManager();

em.close();
emf.close();
```



Persistence Unit

- em from emf.
- emf from Persistence Unit.
- Stored in persistence.xml.

```
<persistence>
 <persistence-unit name="StudentService"
                   transaction-type="RESOURCE_LOCAL">
  <class> edu.miu.cs544.model.Student</class>
  <properties>
   <property name="javax.persistence.jdbc.driver"
             value= "org.apache.derby.jdbc.ClientDriver"/>
   <property name= "javax.persistence.jdbc.url"
             value= "jdbc:derby://localhost:1527/StudentDB;
             create=true"/>
   <property name="javax.persistence.jdbc.user"
             value= "userName"/>
   <property name= "javax.persistence.jdbc.password"
             value="password"/>
   </properties>
  </persistence-unit>
</persistence>
```

Class Mismatch

- How can an Object be represented in a Database?
- One Table, same names or different.
- Split across several tables.
- What else could be possible cases?
  - One table several classes.
  - Inner classes.
- Even a one-to-one mapping could be challenging
  - The needs of the database are more important than the needs of the application.
  - There could be several applications accessing the same database. So one application does not have the power.
  - The Object model must adapt to the database model. Especially since data outlives programs.

Ideal Java Solution

- Objects or Tables?
- Convenience or Ignorance?
- Unobtrusive or transparent?
- Legacy data but new Object.
- Just enough don't over-sell me.
- Local yet mobile.
- Standard yet replaceable.
- Impedance Mismatch

Other Persistence Topics

- Relationships
  - What are they in OO?
  - What are they in RDBMS?
- Inheritance
  - How important is it in OO?
  - How important is it in RDBMS? Is it even there?
- Methods
  - Should we persist OO methods? Why?
  - Is there such a thing in RDMS?
- Queries
  - Should we depend on the RDBMS for all queries?
  - Should we use

Main Point

- JPA is a solution to the Persistence problem in Java. JPA is a specification, the actual persistence is performed by an ORM Provider. We access the ORM provider through the entity manager (em). The em is part of the specifications and like much of JPA it utilizes convention over configuration.
- Everything we see in nature is a manifestation of the Unified Field. We can experience the UF with the regular practice of TM.

ENTITIES AND RELATIONSHIPS