7 Demonstrate component mapping in Hibernate.

**package** hibernate7;

**import** javax.persistence.\*;

@Entity

@Table(name = "employees")

**public** **class** Employee {

@Id

@GeneratedValue(strategy = GenerationType.***IDENTITY***)

**private** Long id;

**private** String name;

@Embedded

**private** Address address;

// Constructors, getters, and setters

**public** Employee() {

}

**public** Employee(String name, Address address) {

**this**.name = name;

**this**.address = address;

}

// Getters and setters

**public** Long getId() {

**return** id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** Address getAddress() {

**return** address;

}

**public** **void** setAddress(Address address) {

**this**.address = address;

}

}

**package** hibernate7;

**import** javax.persistence.Embeddable;

@Embeddable

**public** **class** Address {

**private** String street;

**private** String city;

**private** String state;

**private** String zipCode;

// Constructors, getters, and setters

**public** Address() {

}

**public** Address(String street, String city, String state, String zipCode) {

**this**.street = street;

**this**.city = city;

**this**.state = state;

**this**.zipCode = zipCode;

}

// Getters and setters

**public** String getStreet() {

**return** street;

}

**public** **void** setStreet(String street) {

**this**.street = street;

}

**public** String getCity() {

**return** city;

}

**public** **void** setCity(String city) {

**this**.city = city;

}

**public** String getState() {

**return** state;

}

**public** **void** setState(String state) {

**this**.state = state;

}

**public** String getZipCode() {

**return** zipCode;

}

**public** **void** setZipCode(String zipCode) {

**this**.zipCode = zipCode;

}

}

**package** hibernate7;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** org.hibernate.cfg.Configuration;

**public** **class** MainApp {

**public** **static** **void** main(String[] args) {

// Create a Hibernate configuration and configure it

Configuration configuration = **new** Configuration().configure();

// Build the session factory

SessionFactory sessionFactory = configuration.buildSessionFactory();

// Create a session from the session factory

Session session = sessionFactory.openSession();

// Create an Employee object

Employee employee = **new** Employee("John Doe", **new** Address("123 Main St", "New York", "NY", "10001"));

// Begin the transaction

Transaction transaction = session.beginTransaction();

Long generatedId = (Long) session.save(employee);

// Commit the transaction

transaction.commit();

// Close the session

session.close();

// Close the session factory

sessionFactory.close();

// Print the generated ID

System.***out***.println("Generated ID: " + generatedId);

}

}

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"<http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd>">

<hibernate-configuration>

<session-factory>

<property name=*"hibernate.connection.driver\_class"*>com.mysql.jdbc.Driver</property>

<property name=*"hibernate.connection.password"*>Sona@1234</property>

<property name=*"hibernate.connection.url"*>jdbc:mysql://localhost/db2</property>

<property name=*"hibernate.connection.username"*>root</property>

<property name=*"hibernate.dialect"*>org.hibernate.dialect.MySQLDialect</property>

<property name=*"hibernate.hbm2ddl.auto"*>update</property>

<mapping class=*"hibernate7.Employee"*/>

</session-factory>

</hibernate-configuration>