Lecture 12: Embeddables - Complex Data Types

Creating an Embeddable Class

```java
@Embeddable
public class Laptop {
    private String brand;
    private String model;
    private int ram;

    // Constructors
    public Laptop() {}

    // Getters and Setters
    public String getBrand() { return brand; }
    public void setBrand(String brand) { this.brand = brand; }

    public String getModel() { return model; }
    public void setModel(String model) { this.model = model; }

    public int getRam() { return ram; }
    public void setRam(int ram) { this.ram = ram; }
}
```

Using Embeddable in Entity

```java
@Entity
public class Alien {
    @Id
    private int aid;
    private String aname;
    private String tech;
    private Laptop laptop;  // Embedded object

    // Constructors, getters, and setters
}
```

Key Concepts

- **@Embeddable**: Marks a class as embeddable (value type)
- **Composition**: Alien "has-a" Laptop relationship
- **Table Structure**: Laptop fields become columns in Alien table
- **No Separate Table**: Embeddable objects don't get their own table

Benefits of Embeddables

TELUSKO

1. **Code Organization**: Group related fields together
2. **Reusability**: Same embeddable can be used in multiple entities
3. **Type Safety**: Better than using primitive types for complex data
4. **Maintenance**: Changes to embeddable affect all using entities

TELUSKO