



CS544 EA

Hibernate

Collections

Collections

- So far we've only used java.util.List for collections
 - Other options are available
- Very important to P2I
 - Hibernate provides its **own implementations**
 - Do not create a setter method for a collection!

```
public void setCars(List<Car> cars) {  
    this.cars = cars;  
}
```

.setCars() can be used
to **overwrite**
Hibernate's collection
Implementation!

```
public boolean addCar(Car car) {  
    if (cars.add(car)) {  
        car.setOwner(this);  
        return true;  
    }  
    return false;  
}  
  
public boolean removeCar(Car car) {  
    if (cars.remove(car)) {  
        car.setOwner(null);  
        return true;  
    }  
    return false;  
}
```

4 Types of Collections

- Collections Types:
 - **Bag** (collection without restrictions)
 - **Set** (Bag without duplicates)
 - **List** (Bag with an given arbitrary order)
 - **Map** (Set of keys to a bag of values)
- JPA defaults `java.util.List` to be a Bag!
 - Every collection so far was **mapped as a Bag**

2 Types

- **Non-Indexed** Collections:
 - Don't need anything extra
 - Collections: **Bag** and **Set**
- **Indexed** Collection:
 - Have a separate Index column
 - Hibernate does not auto-generate db-index on column
 - Collections: **Map** and **List**



@OrderBy

- Hibernate's **Bag**, **Set**, and **Map** can be ordered
 - Based on a property of the elements

```
@Entity
public class ToolBox {
    @Id
    @GeneratedValue
    private Long id;
    private String model;
    private String manufacturer;
    @OneToMany(mappedBy="owner")
    @OrderBy("size ASC")
    private Set<Tool> tools
        = new HashSet<>();
}
```

@OrderBy
annotation

```
@Entity
public class Tool {
    @Id
    @GeneratedValue
    private int id;
    private String type;
    private String size;
    @ManyToOne
    private Toolbox owner;
}
```

Tools in the Toolbox
are sorted based
on this size in ascending
order

- List cannot (be re-ordered)
 - It is always ordered based on its order column