



Wszystkie poniższe relacje są relacjami typu bidirectional.

1. Jeden do jednego:

@Entity

@Table(name="client", schema="examscam")

public class **Client** implements Serializable{

private Long id;

private **ClientDetail** clientDetail;

@OneToOne(cascade = CascadeType.ALL, fetch = FetchType.LAZY)

@JoinColumn(name = "detail_id")

public **ClientDetail** getClientDetail() {

return clientDetail;

}

@Id

@GeneratedValue

@Column(name = "id")

public Long getId() {

return id;

...

@Entity

@Table(name="client_detail", schema="examscam")

public class **ClientDetail** {

private **Client** client;

@OneToOne(cascade= CascadeType.ALL, fetch= FetchType.LAZY)

@JoinColumn(name="detailId")

public **Client** getClient() {

return client;

}

...

2. Jeden do wielu:

```
@Entity
@Table(name="client", schema="examscam")
public class Client implements Serializable{

    private List<Address> addresses = new Vector<Address>();

    @OneToMany(mappedBy="client", fetch= FetchType.EAGER,
        cascade= {CascadeType.ALL}, orphanRemoval=true)
    public List<Address> getAddresses() {
        return addresses;
    }

    public void addAddress(Address address) {
        address.setClient(this);
        addresses.add(address);
    }
    ...
}
```

```
@Entity
@Table(name="address")
public class Address implements Serializable {

    private Client client;

    @ManyToOne()
    @JoinColumn(name="client_id")
    public Client getClient() {
        return client;
    }
    ...
}
```

3. Wiele do wielu

```
@Entity
@Table(name="client", schema="examscam")
public class Client implements Serializable{

    private List<Skill> skills = new Vector<Skill>();

    @ManyToMany(cascade= CascadeType.PERSIST)
    @JoinTable(name="client_skill",
        joinColumns={@JoinColumn(name="client_id")},
        inverseJoinColumns={@JoinColumn(name="skill_id")})
    public List<Skill> getSkills() {
        return skills;
    }
    ...
}
```

```
@Entity
@Table(name="skill", schema="examscam")
public class Skill implements Serializable {

    private Set<Client> clients;

    @ManyToMany(cascade= CascadeType.PERSIST)
    @JoinTable(name="client_skill",
        joinColumns={@JoinColumn(name="skill_id")},
        inverseJoinColumns={@JoinColumn(name="client_id")})
    public Set<Client> getClients() {
        return clients;
    }
}
```