PDP
IT–ACADEMY

Cascade types. Eng muhim annotatsiyalar

REJA:

1. Cascade types
2. Eng muhim annotatsiyalar

JPA DA CASCADE TURLARI:

PERSIST

MERGE

REMOVE

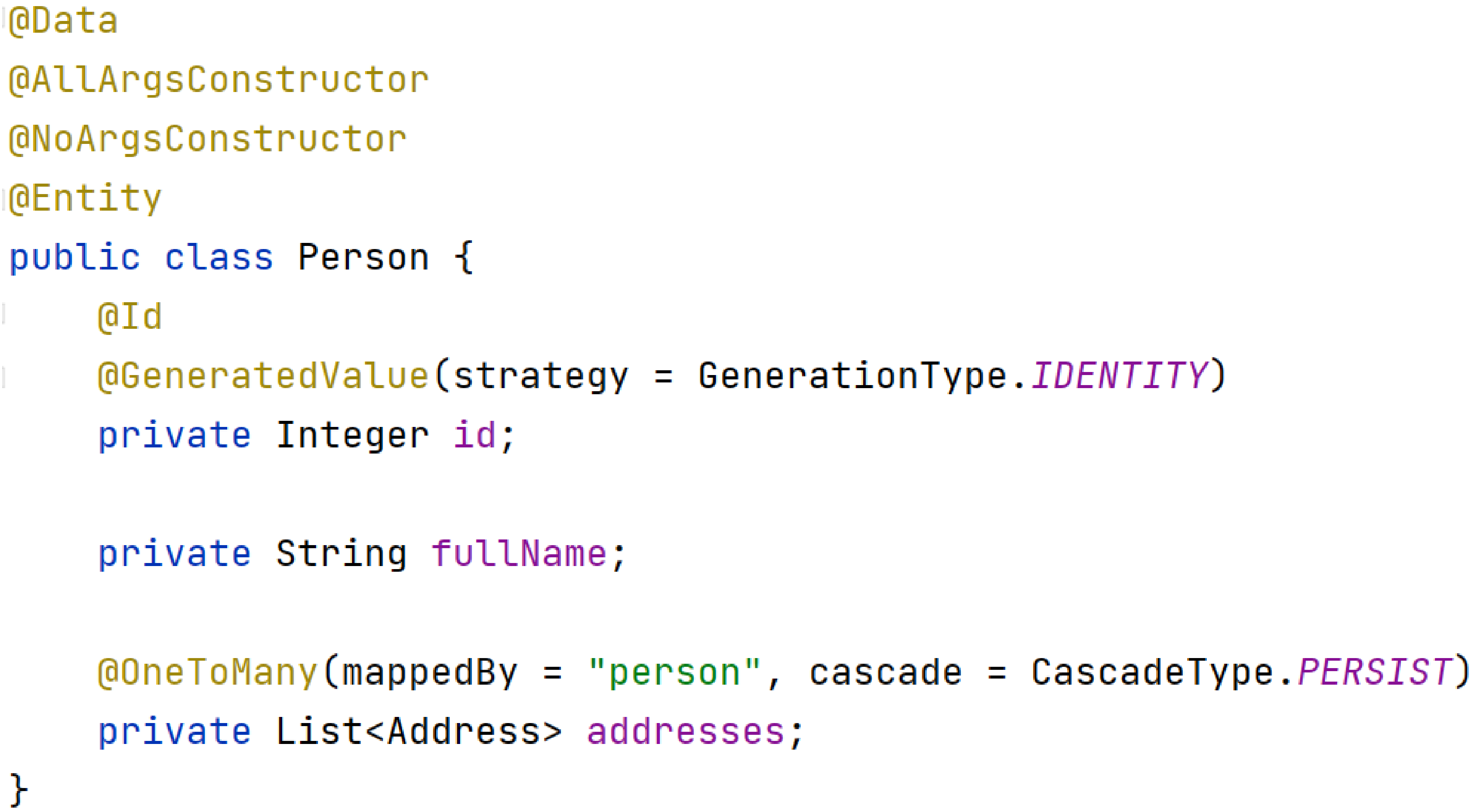
ALL

CascadeType.PERSIST

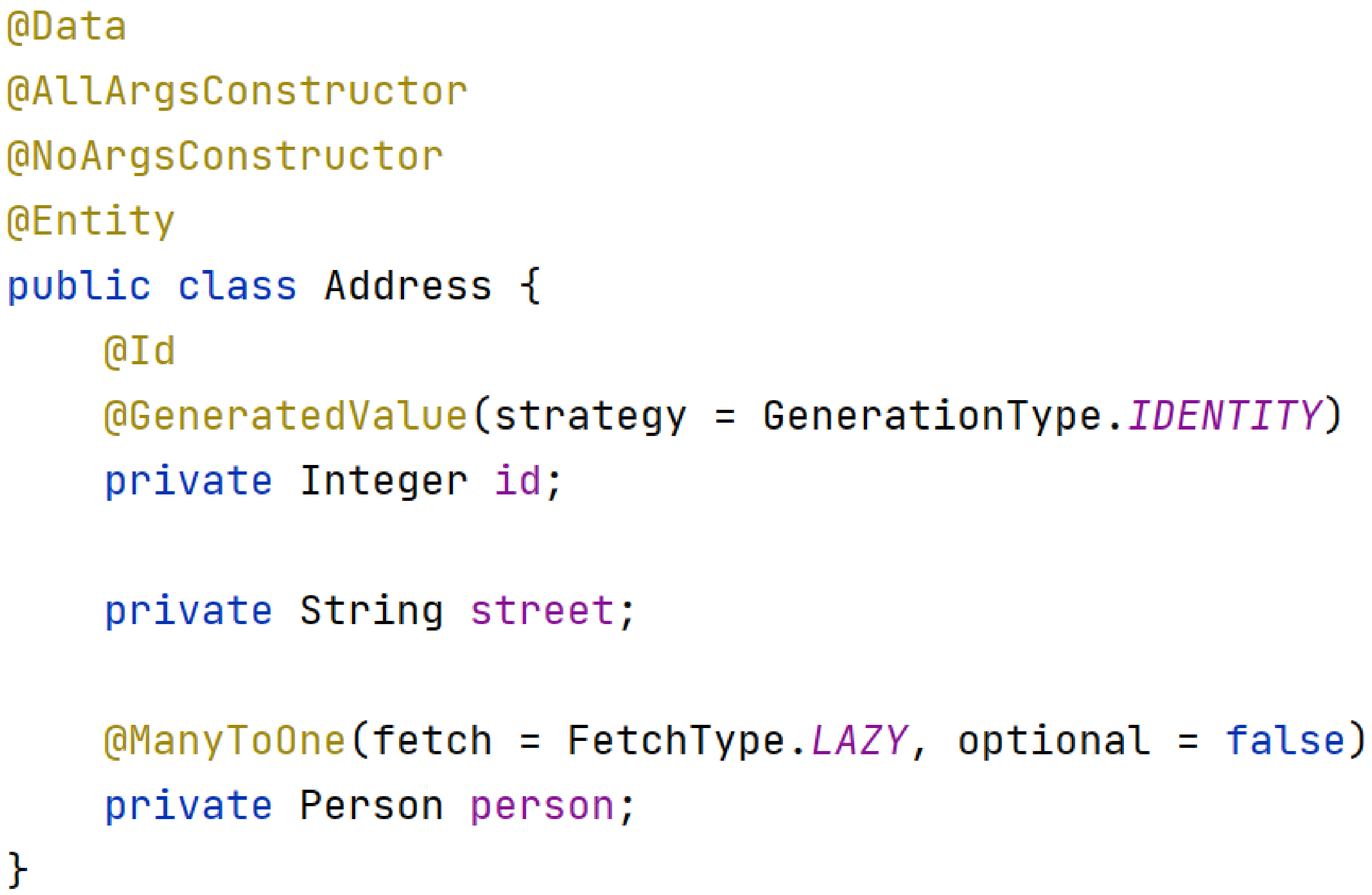
CascadeType.PERSIST buyrug’i vaqtinchalik nusxani doimiy qilib ma’lumotlar omboriga saqlaydi.

CascadeType.PERSIST doimiy operatsiyani otadan bolaga tarqatadi.

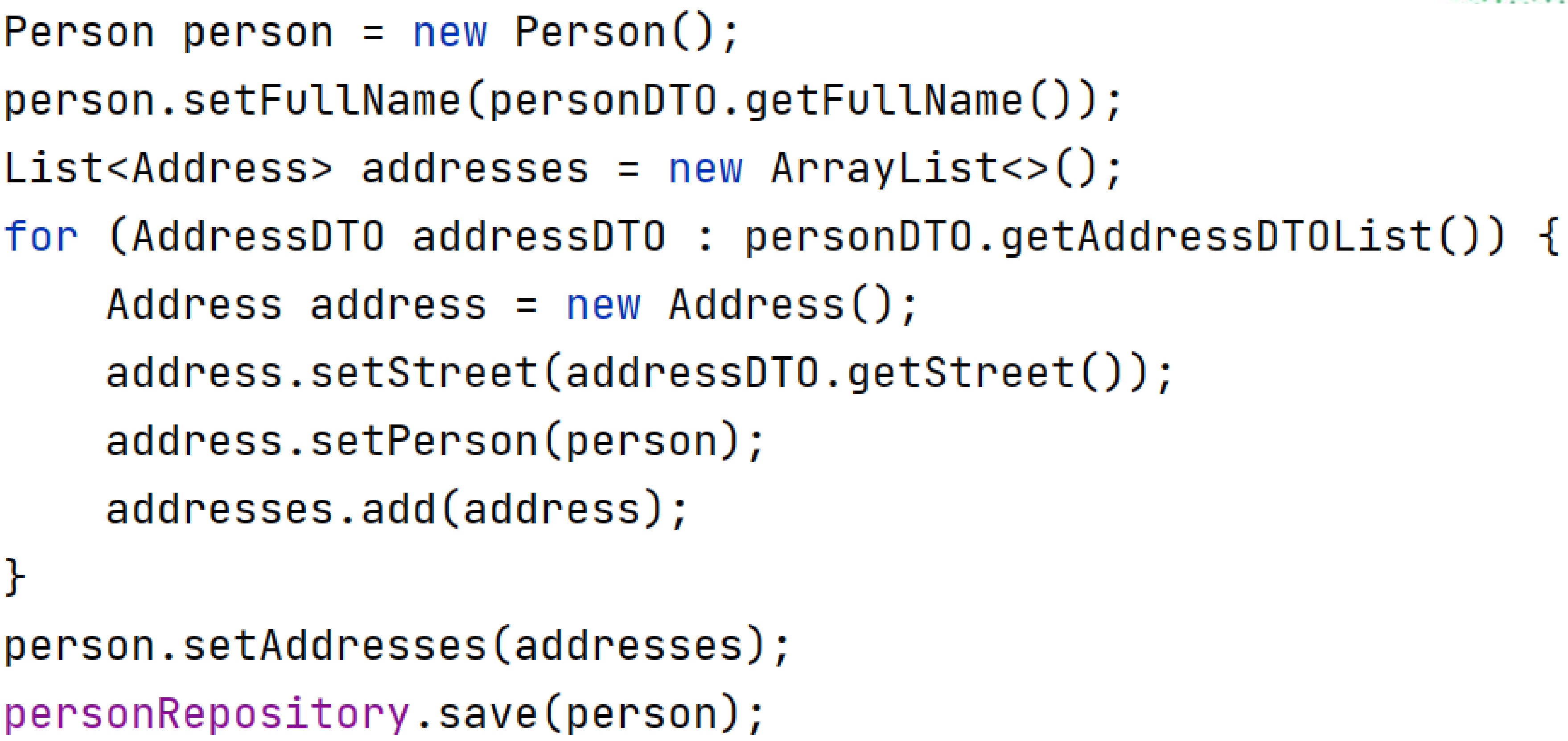
Biz PERSON ob’ektini saqlaganimizda, ADDRESS ob'ekti ham saqlanib qoladi.



```java
@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
public class Person {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;

    private String fullName;

    @OneToMany(mappedBy = "person", cascade = CascadeType.PERSIST)
    private List<Address> addresses;
}
```



```java
@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
public class Address {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;

    private String street;

    @ManyToOne(fetch = FetchType.LAZY, optional = false)
    private Person person;
}
```



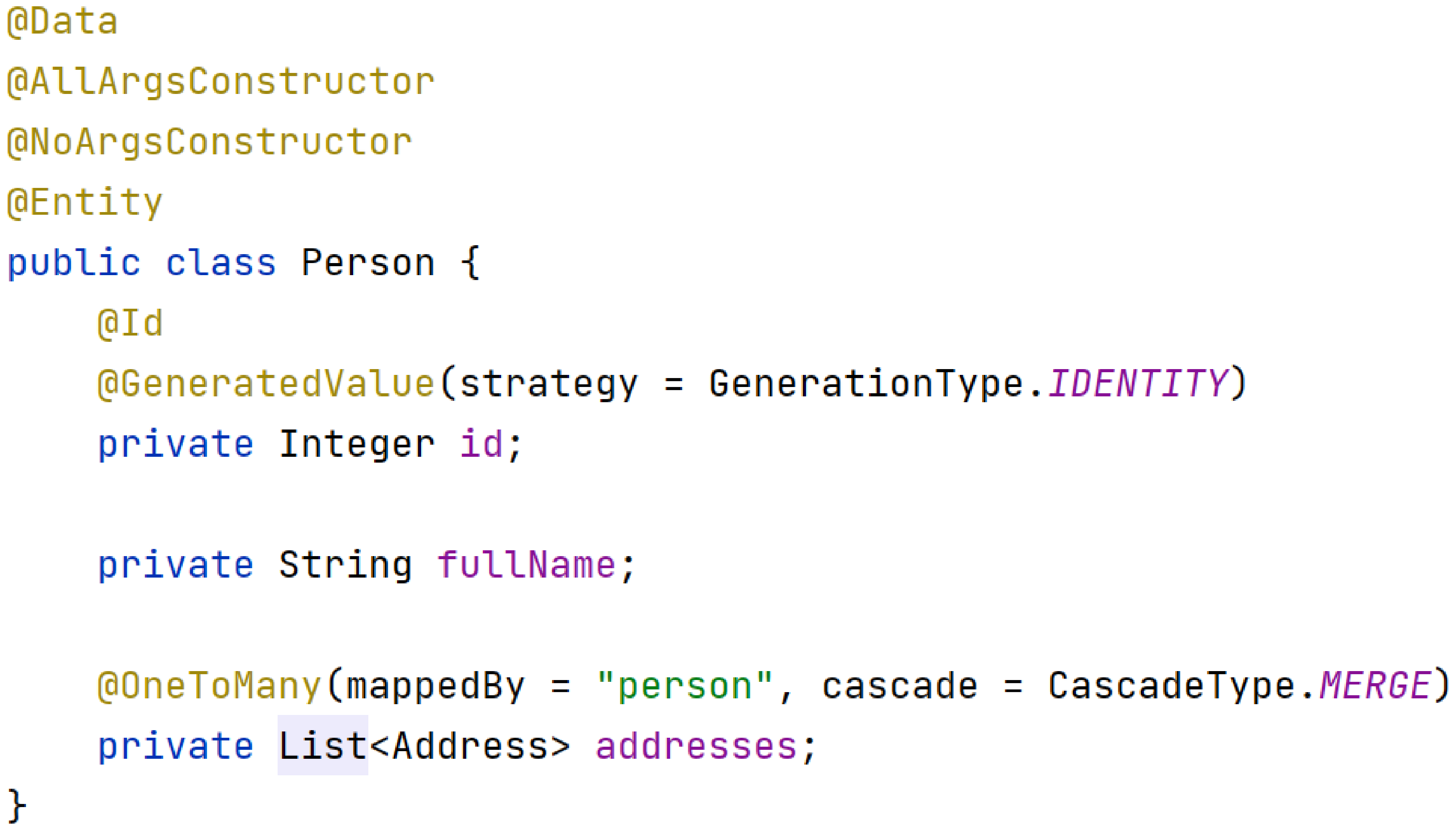
```java
Person person = new Person();
person.setFullName(personDTO.getFullName());
List<Address> addresses = new ArrayList<>();
for (AddressDTO addressDTO : personDTO.getAddressDTOList()) {
    Address address = new Address();
    address.setStreet(addressDTO.getStreet());
    address.setPerson(person);
    addresses.add(address);
}
person.setAddresses(addresses);
personRepository.save(person);
```

CascadeType.MERGE

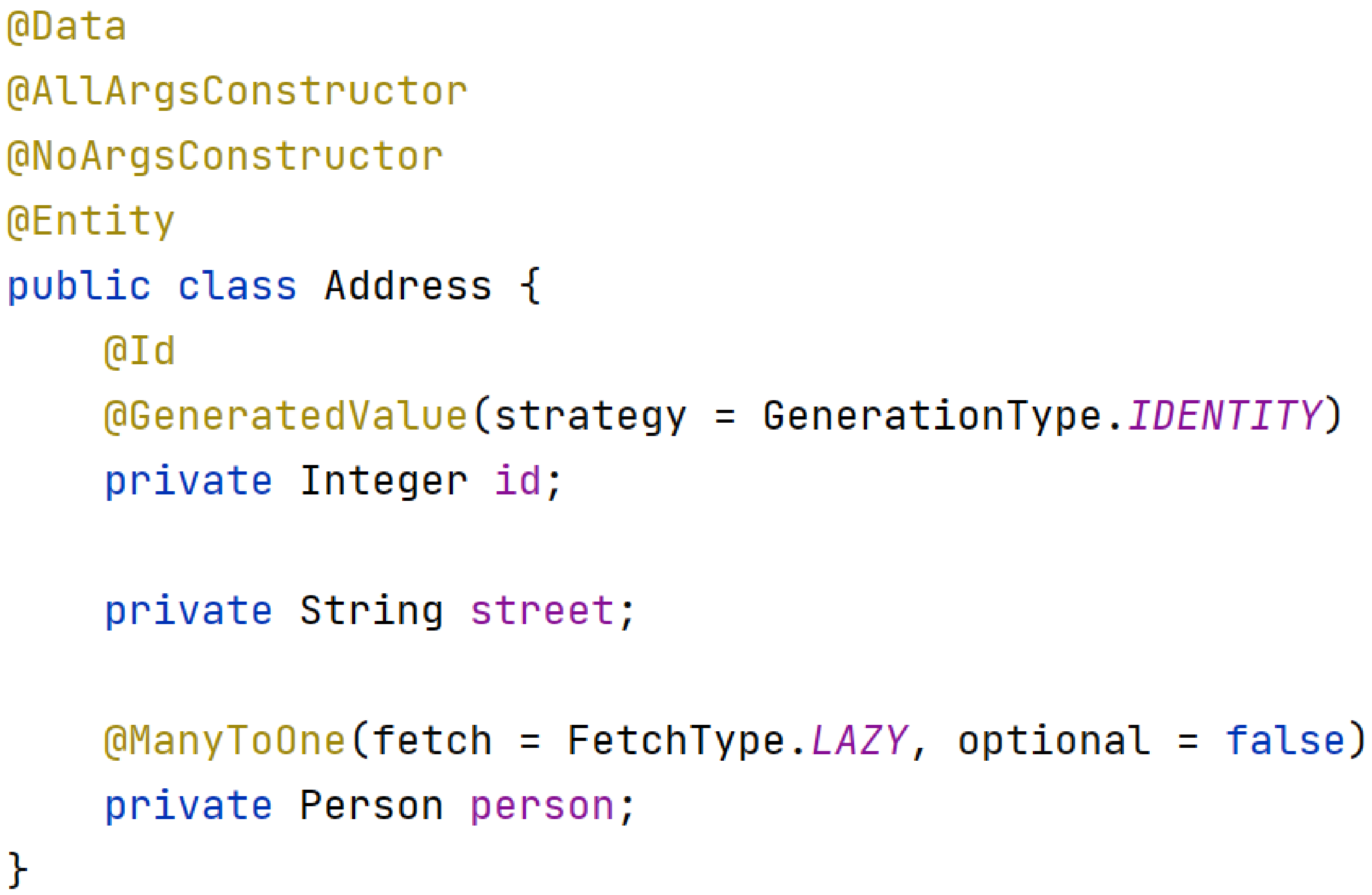
CascadeType.MERGE buyrug’I vaqtinchalik nusxani doimiy qilib ma’lumotlar omboriga saqlaydi.

CascadeType.MERGE doimiy operatsiyani otadan bolaga tarqatadi.

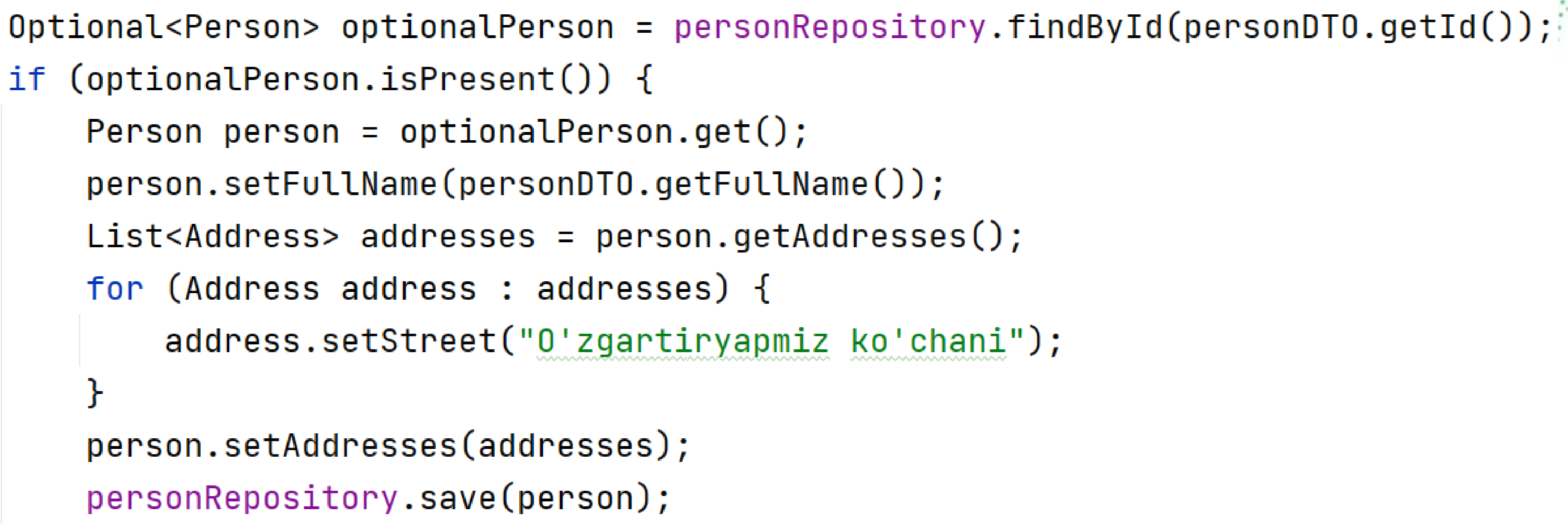
Biz PERSON ob’ektini o’zgartirganimizda, ADDRESS ob'ekti ham saqlanib qoladi.



```java
@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
public class Person {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;

    private String fullName;

    @OneToMany(mappedBy = "person", cascade = CascadeType.MERGE)
    private List<Address> addresses;
}
```



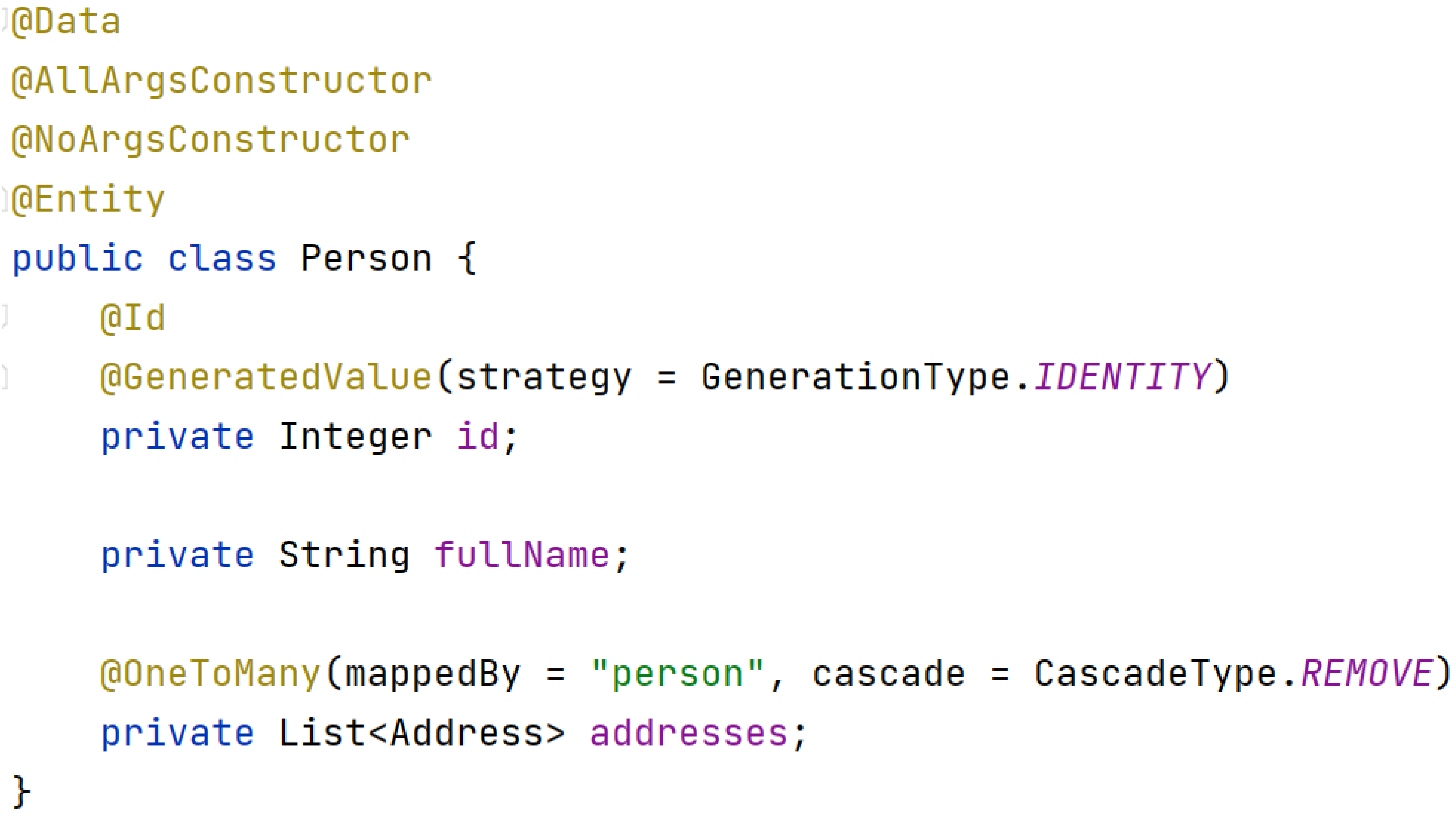
```java
@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
public class Address {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;

    private String street;

    @ManyToOne(fetch = FetchType.LAZY, optional = false)
    private Person person;
}
```



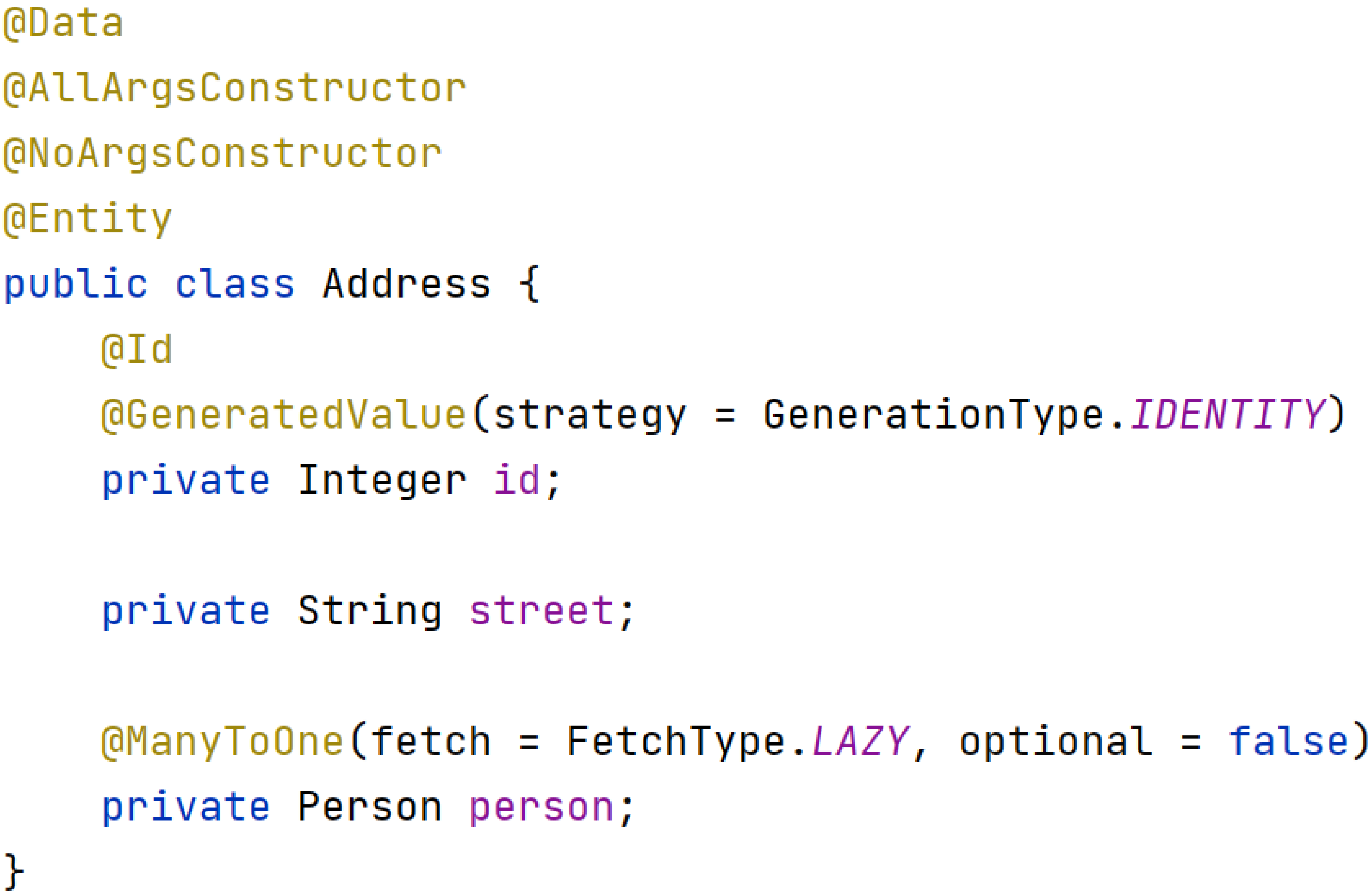
```java
Optional<Person> optionalPerson = personRepository.findById(personDTO.getId());
if (optionalPerson.isPresent()) {
    Person person = optionalPerson.get();
    person.setFullName(personDTO.getFullName());
    List<Address> addresses = person.getAddresses();
    for (Address address : addresses) {
        address.setStreet("O'zgartiryapmiz ko'chani");
    }
    person.setAddresses(addresses);
    personRepository.save(person);
```

CascadeType.REMOVE

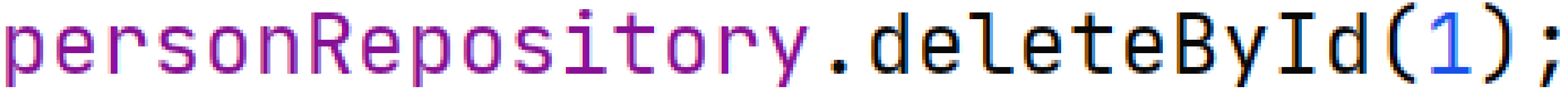
Nomidan ko'rinib turibdiki, REMOVE operatsiyasi ma'lumotlar bazasidan va doimiy kontekstdan ob'ektga mos keladigan qatorni olib tashlaydi. CascadeType.REMOVE olib tashlash operatsiyasini ota-onadan bolaga tarqatadi.



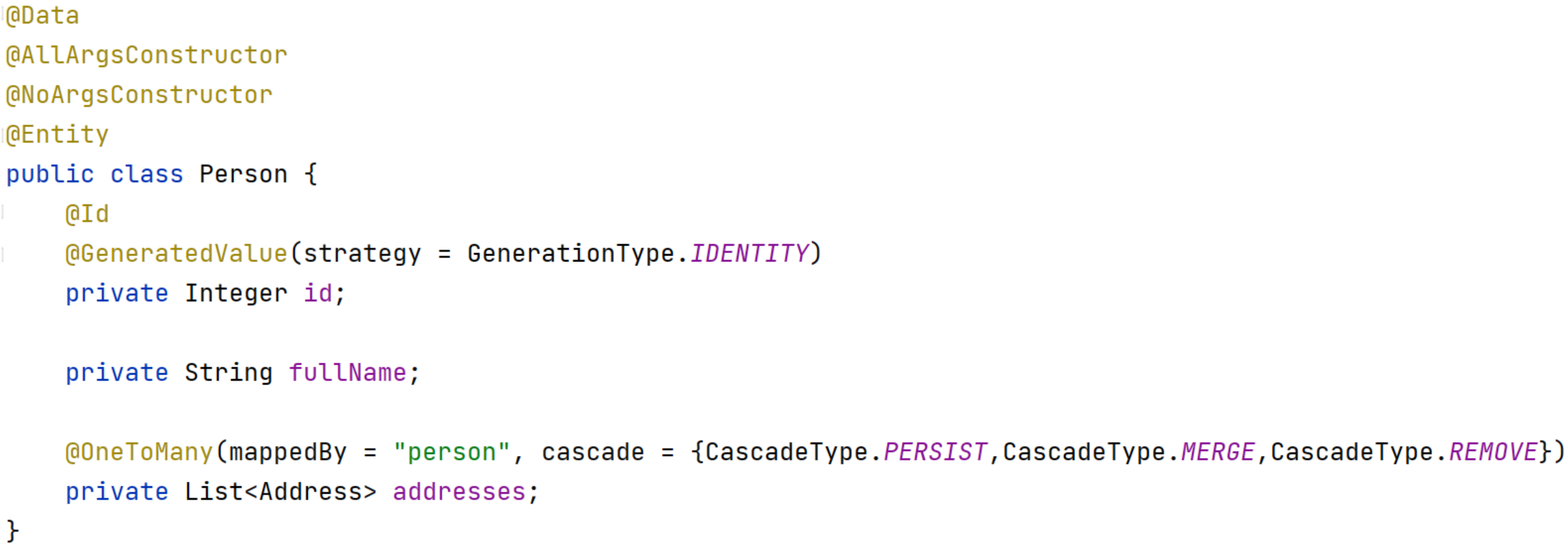
```java
@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
public class Person {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;

    private String fullName;

    @OneToMany(mappedBy = "person", cascade = CascadeType.REMOVE)
    private List<Address> addresses;
}
```



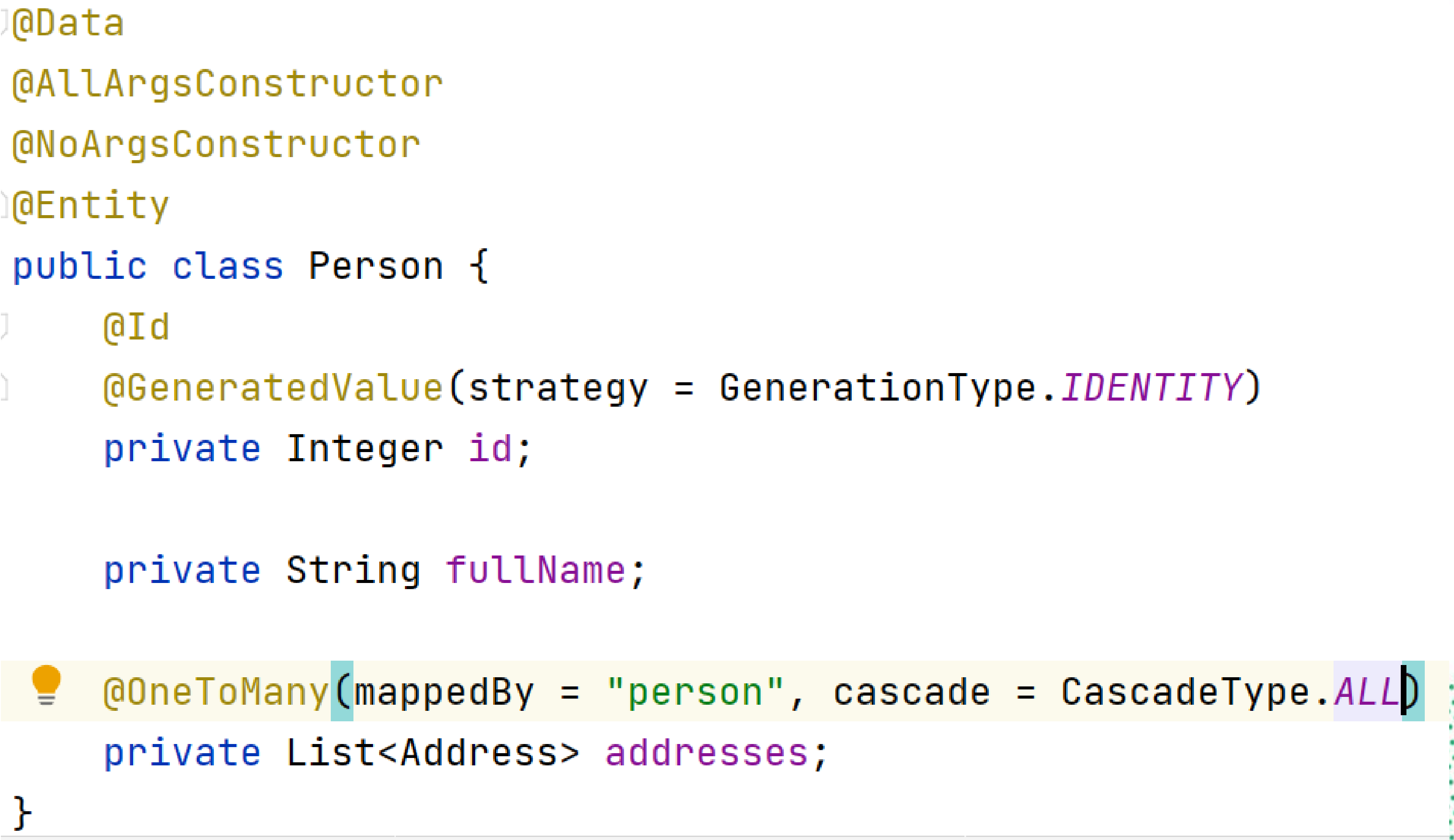
```java
@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
public class Address {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;

    private String street;

    @ManyToOne(fetch = FetchType.LAZY, optional = false)
    private Person person;
}
```



```java
personRepository.deleteById(1);
```



```java
@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
public class Person {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;

    private String fullName;

    @OneToMany(mappedBy = "person", cascade = {CascadeType.PERSIST,CascadeType.MERGE,CascadeType.REMOVE})
    private List<Address> addresses;
}
```



```java
@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
public class Person {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;

    private String fullName;

    @OneToMany(mappedBy = "person", cascade = CascadeType.ALL)
    private List<Address> addresses;
}
```

Eng muhim annotatsiyalar

```
import org.hibernate.annotations.OnDelete;
```

@OnDelete

```
import org.springframework.transaction.annotation.Transactional;
```

@Transactional

PDP
IT-ACADEMY

```
import javax.persistence.Transient;
```

@Transient

PDP
IT–ACADEMY

import org.springframework.data.repository.query.Param;

@Param

PDP
IT-ACADEMY

import org.springframework.data.repository.query.Param;

@Param

import com.fasterxml.jackson.annotation.JsonIgnore;

@JsonIgnore

```
import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
```

@JsonIgnoreProperties

PDP
IT-ACADEMY

```
import com.fasterxml.jackson.annotation.JsonIgnore;
```

@JsonIgnore

```
import javax.persistence.OrderBy;
```

@OrderBy

import org.hibernate.annotations.Where;

@Where

PDP
IT-ACADEMY

E'tiboringiz uchun rahmat.