**Министр науки и высшего образования Российской Федерации**

**Федеральное государственное автономное образовательное учреждение высшего образования**

**«Национальный исследовательский университет ИТМО»**

Факультет информационных технологий и программирования

Лабораторная работа № 2

Выполнил студент группы № M32071

Реброва Татьяна Ивановна

Проверил:

Бутенко Олег Романович

Санкт-Петербург  
2022

FriendDao.java  
  
package dao;  
  
import entities.Friend;  
import entities.Kotik;  
import org.hibernate.Session;  
import org.hibernate.Transaction;  
import utils.HibernateSessionFactoryUtil;  
import java.util.List;  
  
public class FriendDao extends mainDao{  
  
 public Friend findById(int id) {  
 return HibernateSessionFactoryUtil.getSessionFactory().openSession().get(Friend.class, id);  
 }  
  
 public Kotik findKotikById(int id) {  
 return HibernateSessionFactoryUtil.getSessionFactory().openSession().get(Kotik.class, id);  
 }  
  
 public List<Friend> findAll() {  
 List<Friend> friends = (List<Friend>) HibernateSessionFactoryUtil.getSessionFactory().openSession().createQuery("From Friend").list();  
 return friends;  
 }  
}  
  
  
  
  
KotikDao.java  
  
package dao;  
  
import entities.Kotik;  
import entities.Owner;  
import org.hibernate.Session;  
import org.hibernate.Transaction;  
import utils.HibernateSessionFactoryUtil;  
import java.util.List;  
  
public class KotikDao extends mainDao{  
  
 public Kotik findById(int id) {  
 return HibernateSessionFactoryUtil.getSessionFactory().openSession().get(Kotik.class, id);  
 }  
  
 public Owner findOwnerById(int id) {  
 return HibernateSessionFactoryUtil.getSessionFactory().openSession().get(Owner.class, id);  
 }  
  
 public List<Kotik> findAll() {  
 List<Kotik> kotiki = (List<Kotik>) HibernateSessionFactoryUtil.getSessionFactory().openSession().createQuery("From Kotik").list();  
 return kotiki;  
 }  
}  
  
  
  
OwnerDao.java  
  
package dao;  
  
import entities.Kotik;  
import entities.Owner;  
import org.hibernate.Session;  
import org.hibernate.Transaction;  
import utils.HibernateSessionFactoryUtil;  
import java.util.List;  
  
public class OwnerDao extends mainDao{  
 public Owner findById(int id) {  
 return HibernateSessionFactoryUtil.getSessionFactory().openSession().get(Owner.class, id);  
 }  
  
 public Kotik findKotikById(int id) {  
 return HibernateSessionFactoryUtil.getSessionFactory().openSession().get(Kotik.class, id);  
 }  
  
 public List<Owner> findAll() {  
 List<Owner> owners = (List<Owner>) HibernateSessionFactoryUtil.getSessionFactory().openSession().createQuery("From Owner").list();  
 return owners;  
 }  
}  
  
  
  
mainDao.java  
  
package dao;  
  
import org.hibernate.Session;  
import org.hibernate.Transaction;  
import utils.HibernateSessionFactoryUtil;  
  
public abstract class mainDao<T> {  
 public void save(T owner) {  
 Session session = HibernateSessionFactoryUtil.getSessionFactory().openSession();  
 Transaction tx1 = session.beginTransaction();  
 session.save(owner);  
 tx1.commit();  
 session.close();  
 }  
  
 public void update(T owner) {  
 Session session = HibernateSessionFactoryUtil.getSessionFactory().openSession();  
 Transaction tx1 = session.beginTransaction();  
 session.update(owner);  
 tx1.commit();  
 session.close();  
 }  
  
 public void delete(T owner) {  
 Session session = HibernateSessionFactoryUtil.getSessionFactory().openSession();  
 Transaction tx1 = session.beginTransaction();  
 session.delete(owner);  
 tx1.commit();  
 session.close();  
 }  
}  
  
  
  
Color.java  
  
package entities;  
  
import java.util.HashMap;  
import java.util.Map;  
  
public enum Color {  
 BLACK("black"),  
 WHITE("white"),  
 RED("red"),  
 GREY("grey"),  
 GREEN("green"),  
 YELLOW("yellow");  
  
 private final String name;  
  
 Color(String name) {  
 this.name = name;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 private static final Map<String, Color> LOOKUP\_MAP = new HashMap<>();  
  
 static {  
 for (Color env : values()) {  
 LOOKUP\_MAP.put(env.getName(), env);  
 }  
 }  
  
 public static Color getTypeByName(String url) {  
 return LOOKUP\_MAP.get(url);  
 }  
}  
  
  
  
Friend.java  
  
package entities;  
  
import javax.persistence.\*;  
import java.security.Timestamp;  
import java.time.LocalDate;  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.List;  
  
@Entity  
@Table (name = "kotiki\_friends")  
public class Friend{  
 @Id  
 @GeneratedValue(strategy = GenerationType.IDENTITY)  
 private int id;  
 @ManyToOne(fetch = FetchType.LAZY)  
 @JoinColumn(name = "kotik")  
 private Kotik kotik;  
 @OneToOne  
 @JoinColumn(name = "friend")  
 private Kotik friend;  
  
 public Friend() {  
 }  
  
 public Friend(Kotik kotik, Kotik friend) {  
 this.kotik = kotik;  
 this.friend = friend;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public Kotik getKotik() {  
 return kotik;  
 }  
  
 public void setKotik(Kotik kotik) {  
 this.kotik = kotik;  
 }  
  
 public Kotik getFriend() {  
 return friend;  
 }  
  
 public void setFriend(Kotik friend) {  
 this.friend = friend;  
 }  
}  
  
  
Kotik.java  
  
package entities;  
  
import tools.KotikiException;  
  
import javax.persistence.\*;  
import java.time.LocalDate;  
import java.util.ArrayList;  
import java.util.List;  
  
@Entity  
@Table (name = "kotiki")  
public class Kotik{  
 @Id  
 @GeneratedValue(strategy = GenerationType.IDENTITY)  
 private int id;  
  
 private String name;  
 private LocalDate birthday;  
 private String breed;  
  
 @Column(name = "color")  
 private Color color;  
  
 @ManyToOne(fetch = FetchType.LAZY)  
 @JoinColumn(name = "owner")  
 private Owner owner;  
  
 @OneToMany(mappedBy = "kotik", orphanRemoval = true)  
 private final List<Friend> friends = new ArrayList<>();  
  
 public Kotik() {  
 }  
  
 public Kotik(String name, LocalDate birthday, String breed,  
 Color color) {  
 this.name = name;  
 this.birthday = birthday;  
 this.breed = breed;  
 this.color = color;  
 //this.ownerId = ownerId;  
 }  
  
 public Friend addFriend(Kotik kotik) throws KotikiException {  
 if(kotik == null) {  
 throw new KotikiException("Sorry not found kotik");  
 }  
 if(kotik == this) {  
 throw new KotikiException("Sorry you can't be frendship with yourself I must to say your owner about your schiza");  
 }  
 Friend newFriend = new Friend(this, kotik);  
 friends.add(newFriend);  
 return newFriend;  
 }  
  
 public void removeFriends(Friend friend) {  
 friends.remove(friend);  
 }  
  
 public Integer getId() {  
 return id;  
 }  
  
 public void setId(Integer id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public LocalDate getBirthDay() {  
 return birthday;  
 }  
  
 public void setBirthDay(LocalDate birthDay) {  
 this.birthday = birthDay;  
 }  
  
 public String getBreed() {  
 return breed;  
 }  
  
 public void setBreed(String breed) {  
 this.breed = breed;  
 }  
  
 public Color getColorId() {  
 return color;  
 }  
  
 public void setColorId(Color color) {  
 this.color = color;  
 }  
  
 public Owner getOwnerId() {  
 return owner;  
 }  
  
 public void setOwnerId(Owner owner) {  
 this.owner = owner;  
 }  
  
}  
  
  
  
Owner.java  
  
package entities;  
  
import javax.persistence.\*;  
import java.time.LocalDate;  
import java.util.ArrayList;  
import java.util.List;  
  
@Entity  
@Table (name = "owners")  
public class Owner{  
 @Id  
 @GeneratedValue(strategy = GenerationType.IDENTITY)  
 private int id;  
 @Column(name = "name")  
 private String name;  
 private LocalDate birthday;  
  
 @OneToMany(mappedBy = "owner", cascade = CascadeType.ALL, orphanRemoval = true)  
 private List<Kotik> kotiki;  
  
 public Owner() {  
 }  
  
 public Owner(String name, LocalDate birthday) {  
 this.name = name;  
 this.birthday = birthday;  
 kotiki = new ArrayList<>();  
 }  
  
 public void addKotik(Kotik kotik) {  
 kotik.setOwnerId(this);  
 kotiki.add(kotik);  
 }  
  
 public void removeKotik(Kotik kotik) {  
 kotiki.remove(kotik);  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public LocalDate getBirthDay() {  
 return birthday;  
 }  
  
 public void setBirthDay(LocalDate birthDay) {  
 this.birthday = birthDay;  
 }  
  
 public List<Kotik> getKotiki() {  
 return kotiki;  
 }  
}  
  
  
  
HibernateSessionFactoryUtil.java  
  
package utils;  
  
import entities.Friend;  
import entities.Kotik;  
import entities.Owner;  
import entities.Color;  
import org.hibernate.SessionFactory;  
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;  
import org.hibernate.cfg.Configuration;  
  
public class HibernateSessionFactoryUtil {  
 private static SessionFactory sessionFactory;  
  
 private HibernateSessionFactoryUtil() {}  
  
 public static SessionFactory getSessionFactory() {  
 if (sessionFactory == null) {  
 try {  
 Configuration configuration = new Configuration().configure();  
 configuration.addAnnotatedClass(Owner.class);  
 configuration.addAnnotatedClass(Kotik.class);  
 configuration.addAnnotatedClass(Color.class);  
 configuration.addAnnotatedClass(Friend.class);  
 StandardServiceRegistryBuilder builder = new StandardServiceRegistryBuilder().applySettings(configuration.getProperties());  
 sessionFactory = configuration.buildSessionFactory(builder.build());  
  
 } catch (Exception e) {  
 System.out.println("Исключение!" + e);  
 }  
 }  
 return sessionFactory;  
 }  
}  
  
  
Main.java  
  
  
import entities.Color;  
import entities.Friend;  
import entities.Kotik;  
import entities.Owner;  
import services.MainService;  
  
import java.sql.SQLException;  
import java.time.LocalDate;  
  
public class Main {  
 public static void main(String[] args) throws Exception {  
 MainService mainService = new MainService();  
 Owner owner = new Owner("Masha", LocalDate.now());  
 mainService.saveOwner(owner);  
 Kotik kotik1 = new Kotik("Vudi1", LocalDate.of(2004, 05, 29), "No name", Color.GREY);  
 kotik1.setOwnerId(owner);  
 owner.addKotik(kotik1);  
 Kotik kotik2 = new Kotik("Vudi2", LocalDate.of(2004, 05, 29), "No name", Color.YELLOW);  
 kotik2.setOwnerId(owner);  
 owner.addKotik(kotik2);  
 mainService.updateOwner(owner);  
  
 Owner owner2 = new Owner("Dasha", LocalDate.now());  
 mainService.saveOwner(owner2);  
 Kotik kotikFriend = new Kotik("Kuki", LocalDate.of(2004, 05, 29), "No name", Color.GREY);  
 kotikFriend.setOwnerId(owner2);  
 owner2.addKotik(kotikFriend);  
 mainService.updateOwner(owner2);  
  
 owner.setName("Sasha");  
 mainService.updateOwner(owner);  
 kotik1.setName("Rubi");  
 mainService.updateKotik(kotik1);  
 Friend friend = kotik1.addFriend(kotik2);  
 mainService.saveFriend(friend);  
 Friend friend2 = kotik1.addFriend(kotikFriend);  
 mainService.saveFriend(friend2);  
 Owner findOwner = mainService.findOwnerById(kotik1.getOwnerId().getId());  
 System.out.println(findOwner.getName());  
  
  
 /\*mainService.deleteOwner(owner);  
 mainService.deleteOwner(owner2);\*/  
 }  
}  
  
  
MainService.java  
  
package services;  
  
import dao.FriendDao;  
import dao.KotikDao;  
import entities.Friend;  
import entities.Kotik;  
import entities.Owner;  
import dao.OwnerDao;  
  
import java.util.List;  
  
public class MainService {  
 private OwnerDao ownersDao = new OwnerDao();  
 private KotikDao kotikiDao = new KotikDao();  
 private FriendDao friendsDao = new FriendDao();  
  
 public Owner findOwner(int id) {  
 return ownersDao.findById(id);  
 }  
  
 public List<Owner> findAllOwners() {  
 return ownersDao.findAll();  
 }  
  
 public void saveOwner(Owner owner) {  
 ownersDao.save(owner);  
 }  
  
 public void deleteOwner(Owner owner) {  
 ownersDao.delete(owner);  
 }  
  
 public void updateOwner(Owner owner) {  
 ownersDao.update(owner);  
 }  
  
 public Kotik findKotikById(int id) {  
 return ownersDao.findKotikById(id);  
 }  
  
  
 public Kotik findKotik(int id) {  
 return kotikiDao.findById(id);  
 }  
  
 public void saveKotik(Kotik kotik) {  
 kotikiDao.save(kotik);  
 }  
  
 public void deleteKotik(Kotik kotik) {  
 kotikiDao.delete(kotik);  
 }  
  
 public void updateKotik(Kotik kotik) {  
 kotikiDao.update(kotik);  
 }  
  
 public List<Kotik> findAllKOtiki() {  
 return kotikiDao.findAll();  
 }  
  
 public Owner findOwnerById(int id) {  
 return kotikiDao.findOwnerById(id);  
 }  
  
  
 public Friend findFriend(int id) {  
 return friendsDao.findById(id);  
 }  
  
 public void saveFriend(Friend friend) {  
 friendsDao.save(friend);  
 }  
  
 public void deleteFriend(Friend friend) {  
 friendsDao.delete(friend);  
 }  
  
 public void updateFriend(Friend friend) {  
 friendsDao.update(friend);  
 }  
  
 public List<Friend> findAllFriends() {  
 return friendsDao.findAll();  
 }  
  
 public Kotik findKotikByIdInFriends(int id) {  
 return friendsDao.findKotikById(id);  
 }  
  
}