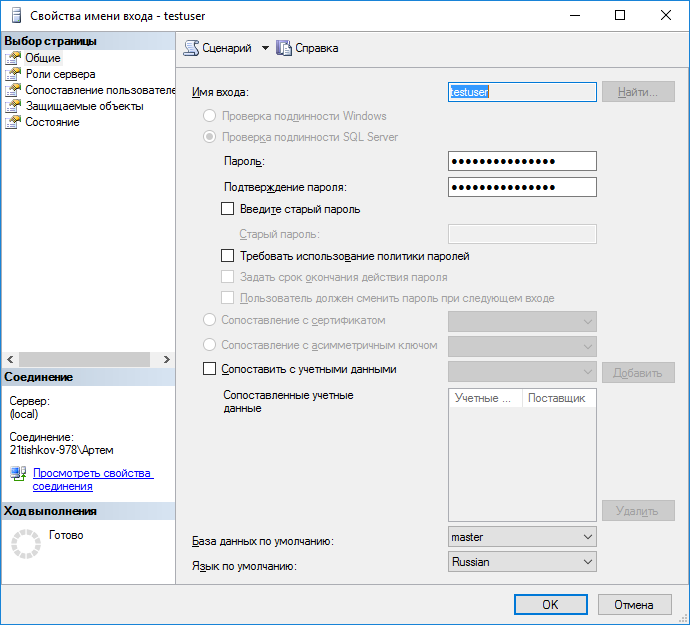
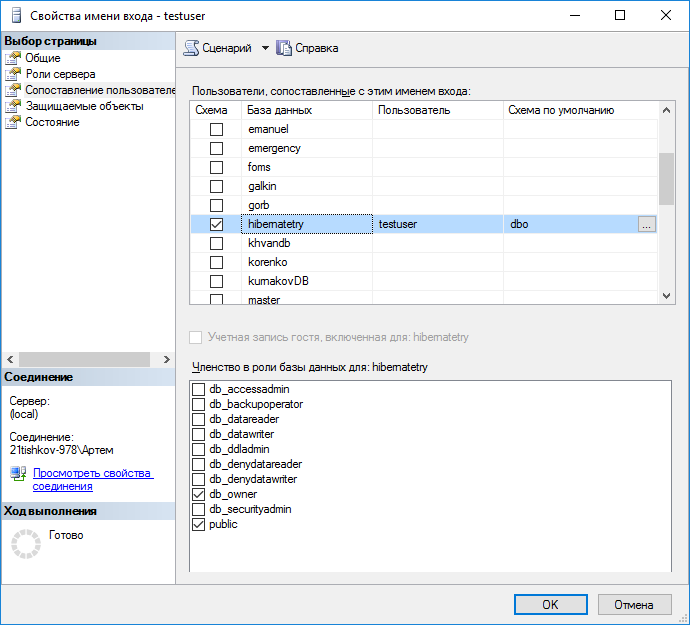
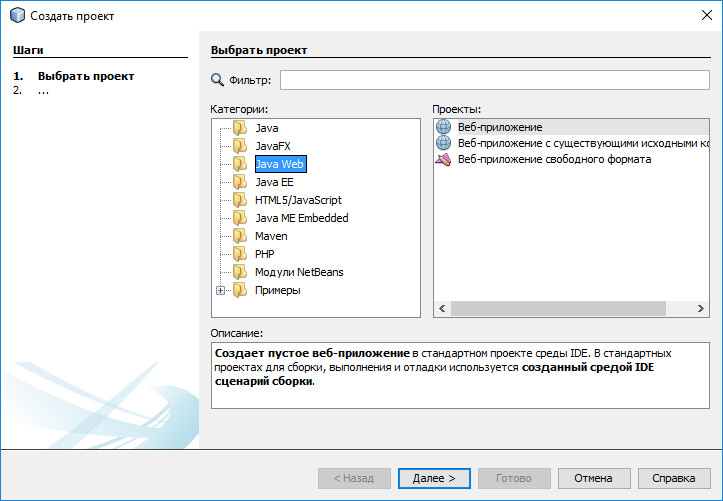
create table tTest (

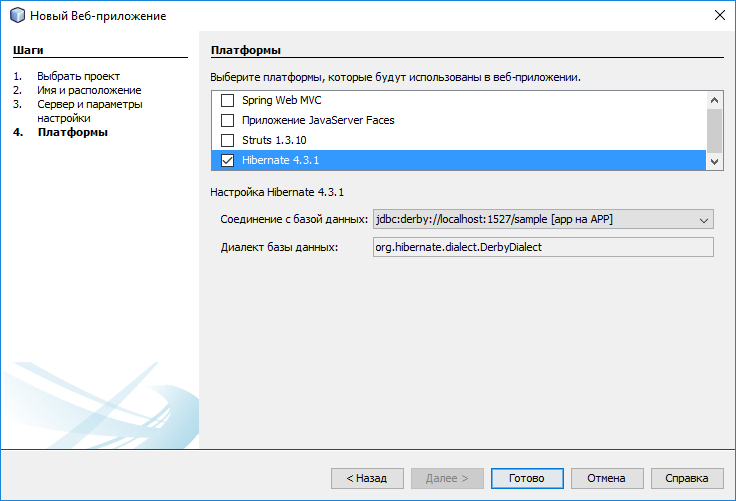
id int identity(1,1) constraint PK\_tTest primary key,

someText nvarchar(255)

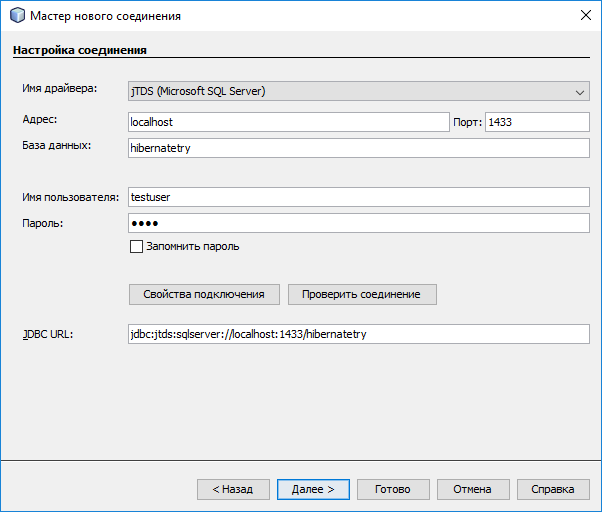
) 

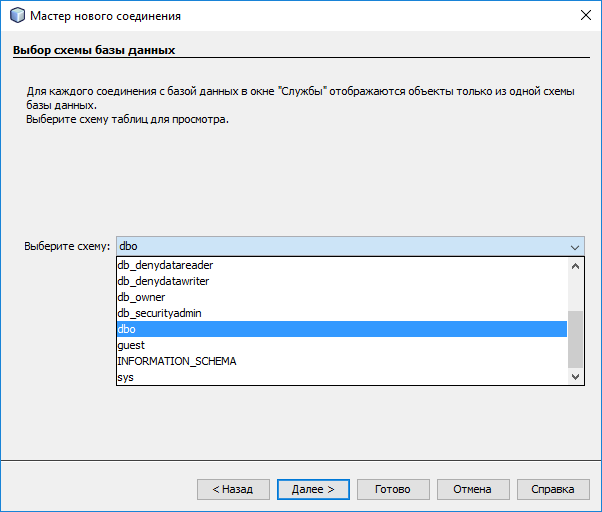


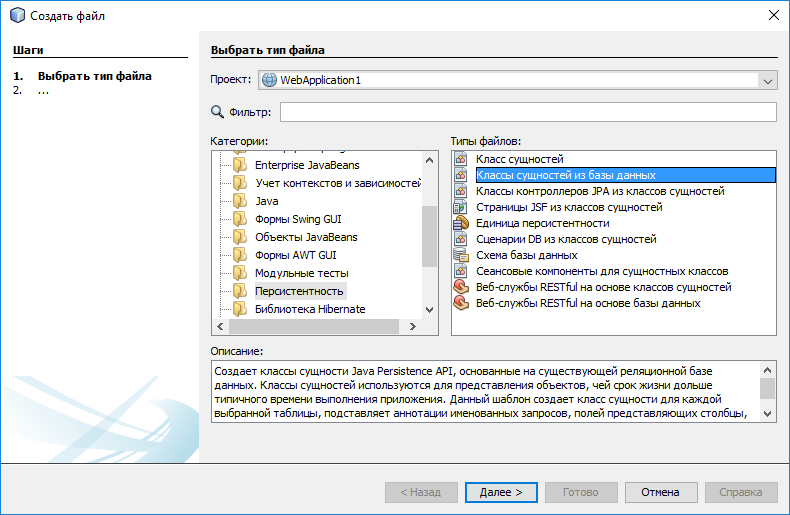


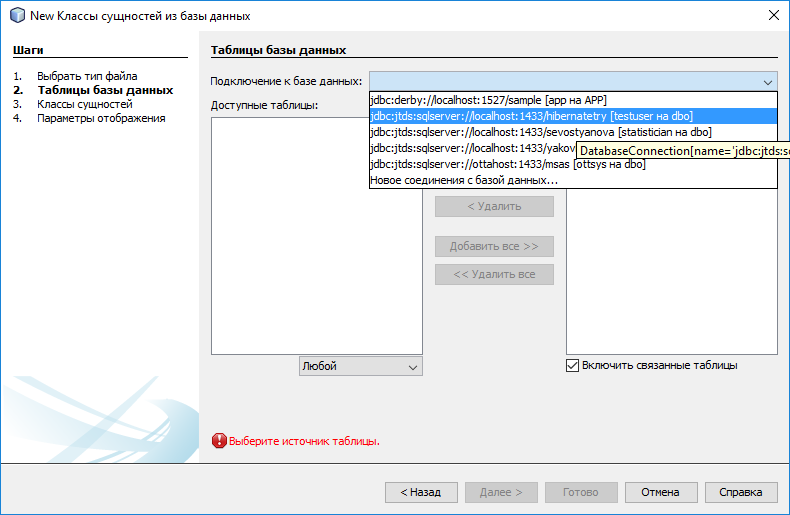


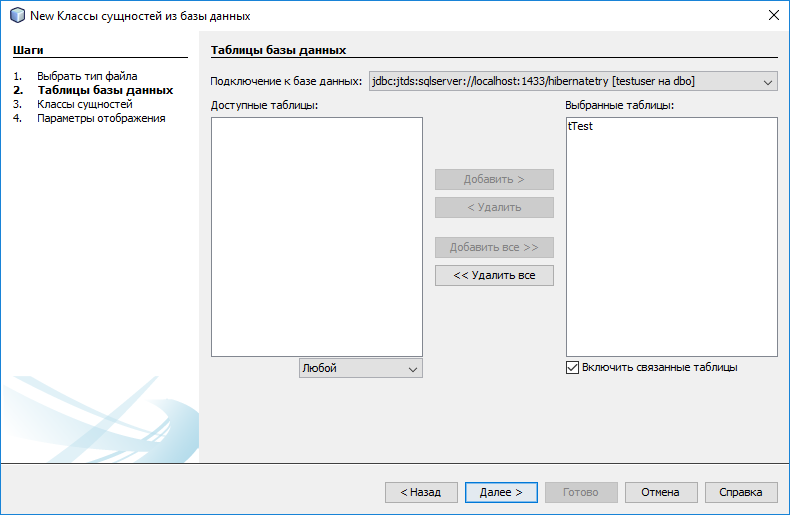
<https://sourceforge.net/projects/jtds/files/>

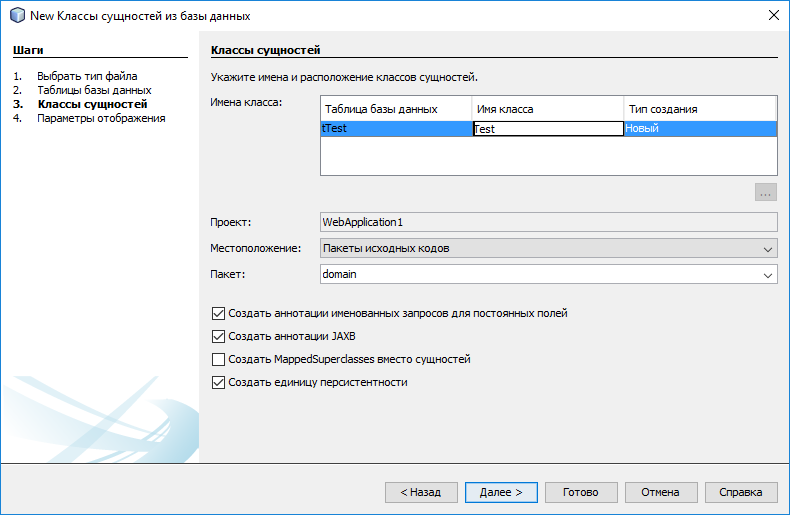


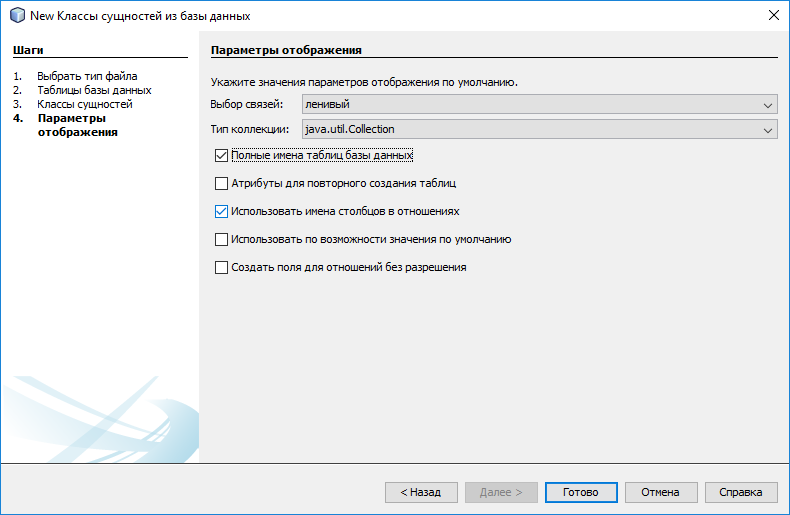


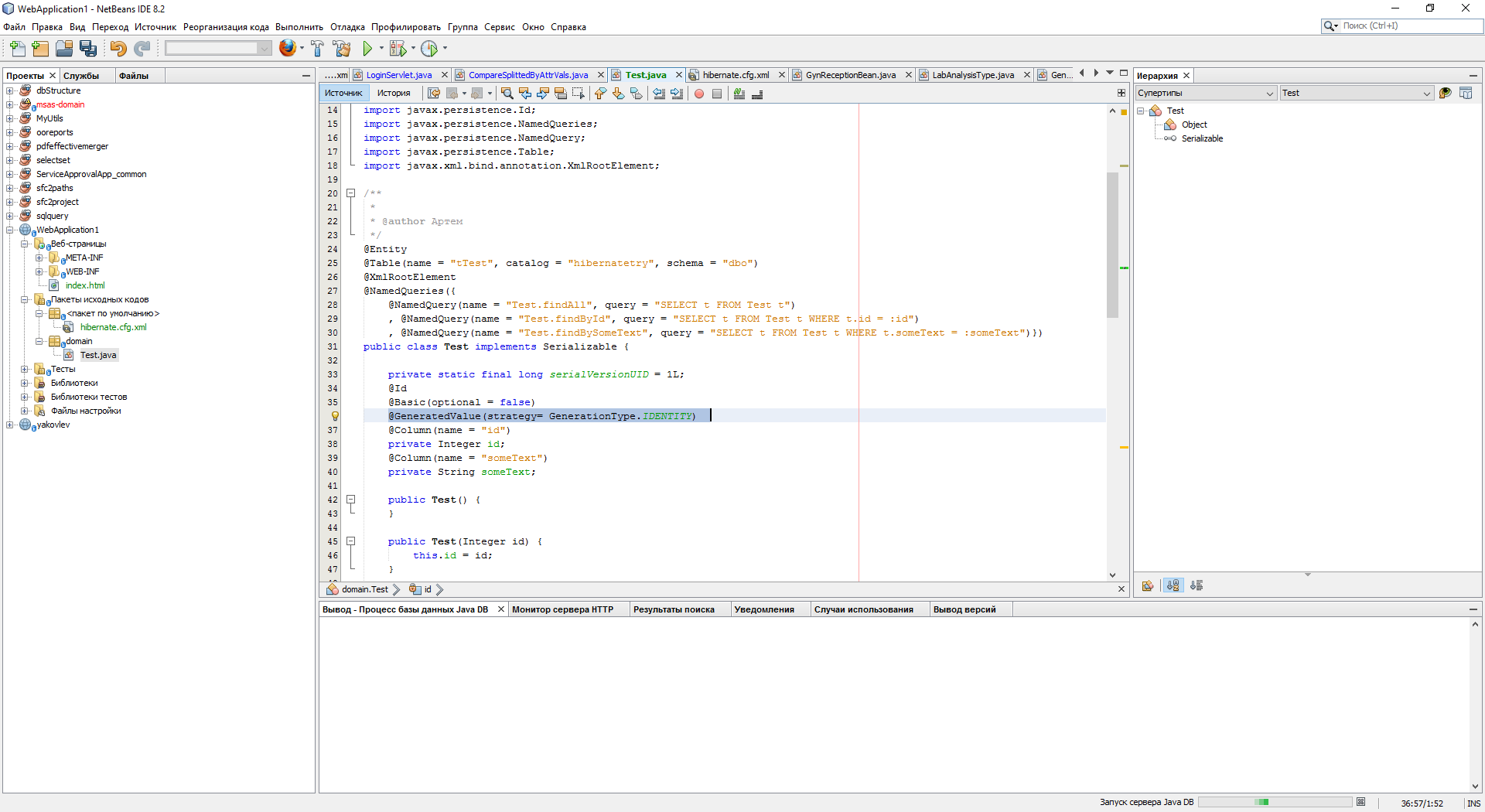


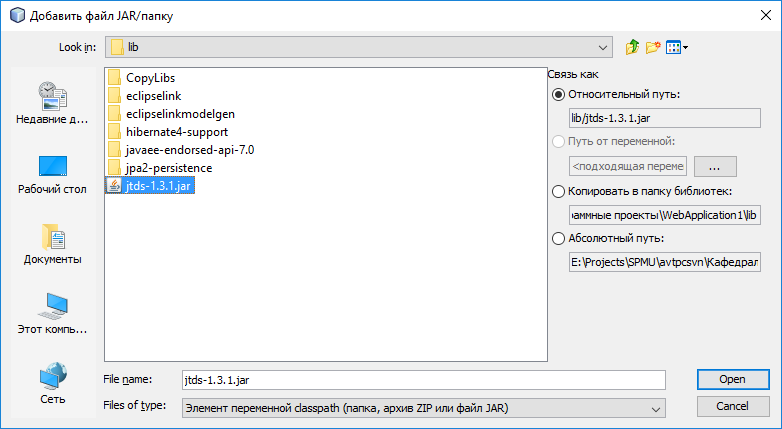












<hibernate-configuration>

<session-factory>

<property name="hibernate.connection.driver\_class">net.sourceforge.jtds.jdbc.Driver</property>

<property name="hibernate.connection.url">jdbc:jtds:sqlserver://localhost:1433/hibernatetry</property>

<property name="hibernate.connection.username">testuser</property>

<property name="hibernate.connection.password">test</property>

<mapping class="domain.Test"/>

</session-factory>

</hibernate-configuration>

private static final long serialVersionUID = 1L;

@Id

@Basic(optional = false)

@GeneratedValue(strategy= GenerationType.IDENTITY)

@Column(name = "id")

private Integer id;

@Column(name = "someText")

private String someText;

public static void main(String[] args) {

Configuration conf = new Configuration();

conf.configure();

SessionFactory sessionFactory = conf.buildSessionFactory();

Session s = sessionFactory.openSession();

/\*

Test test = (Test) s.get(Test.class, 1);

System.out.println(test.getSomeText());

test.setSomeText("Some another text");

s.beginTransaction();

s.saveOrUpdate(test);

s.getTransaction().commit();

s.close();

\*/

s = sessionFactory.openSession();

Test test2 = (Test) s.get(Test.class, 1);

System.out.println(test2.getSomeText());

s.close();

s = sessionFactory.openSession();

s.beginTransaction();

test2.setId(null);

test2.setSomeText("Some text 2");

s.saveOrUpdate(test2);

s.getTransaction().commit();

s.close();

s = sessionFactory.openSession();

test2 = (Test) s.get(Test.class,6);

s.beginTransaction();

s.delete(test2);

s.getTransaction().commit();

s.close();

}

create table tMarks (

id int identity(1,1) constraint PK\_tMarks primary key,

fname nvarchar(255)

)

create table tStudents (

id int identity(1,1) constraint PK\_tStudents primary key,

fio nvarchar(255),

lnkMark int

constraint FK\_tStudents\_tMarks

foreign key (lnkMark)

references tMarks(id),

)

insert into tMarks (fname) values ('îòëè÷íèê'),('õîðîøèñò'),('òðîå÷íèê'),('ðàçãèëüäÿé')

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package domain;

import java.io.Serializable;

import javax.persistence.Basic;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.ManyToOne;

import javax.persistence.NamedQueries;

import javax.persistence.NamedQuery;

import javax.persistence.Table;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

/\*\*

\*

\* @author Артем

\*/

@Entity

@Table(name = "tStudents", catalog = "hibernatetry", schema = "dbo")

@NamedQueries({

@NamedQuery(name = "Student.findAll", query = "SELECT s FROM Student s")

, @NamedQuery(name = "Student.findById", query = "SELECT s FROM Student s WHERE s.id = :id")

, @NamedQuery(name = "Student.findByFio", query = "SELECT s FROM Student s WHERE s.fio = :fio")})

public class Student implements Serializable {

private static final long serialVersionUID = 1L;

@Id

@Basic(optional = false)

@GeneratedValue(strategy= GenerationType.IDENTITY)

@Column(name = "id")

private Integer id;

@Column(name = "fio")

private String fio;

@JoinColumn(name = "lnkMark", referencedColumnName = "id")

@ManyToOne

private Mark mark;

public Student() {

}

public Student(Integer id) {

this.id = id;

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getFio() {

return fio;

}

public void setFio(String fio) {

this.fio = fio;

}

public Mark getMark() {

return mark;

}

public void setMark(Mark mark) {

this.mark = mark;

}

@Override

public int hashCode() {

int hash = 0;

hash += (id != null ? id.hashCode() : 0);

return hash;

}

@Override

public boolean equals(Object object) {

// TODO: Warning - this method won't work in the case the id fields are not set

if (!(object instanceof Student)) {

return false;

}

Student other = (Student) object;

if ((this.id == null && other.id != null) || (this.id != null && !this.id.equals(other.id))) {

return false;

}

return true;

}

@Override

public String toString() {

return "domain.Students[ id=" + id + " ]";

}

public static void main(String[] args) {

Configuration conf = new Configuration();

conf.configure();

SessionFactory sessionFactory = conf.buildSessionFactory();

Session s = sessionFactory.openSession();

/\*

Test test = (Test) s.get(Test.class, 1);

System.out.println(test.getSomeText());

test.setSomeText("Some another text");

s.beginTransaction();

s.saveOrUpdate(test);

s.getTransaction().commit();

s.close();

\*/

s = sessionFactory.openSession();

Mark mark = (Mark) s.get(Mark.class, 1);

Student student = new Student();

student.setFio("Иванов Иван Иванович");

student.setMark(mark);

s.beginTransaction();

s.saveOrUpdate(student);

s.getTransaction().commit();

s.close();

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package domain;

import java.io.Serializable;

import javax.persistence.Basic;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.NamedQueries;

import javax.persistence.NamedQuery;

import javax.persistence.Table;

/\*\*

\*

\* @author Артем

\*/

@Entity

@Table(name = "tMarks", catalog = "hibernatetry", schema = "dbo")

@NamedQueries({

@NamedQuery(name = "Mark.findAll", query = "SELECT m FROM Mark m")

, @NamedQuery(name = "Mark.findById", query = "SELECT m FROM Mark m WHERE m.id = :id")

, @NamedQuery(name = "Mark.findByFname", query = "SELECT m FROM Mark m WHERE m.fname = :fname")})

public class Mark implements Serializable {

private static final long serialVersionUID = 1L;

@Id

@Basic(optional = false)

@GeneratedValue(strategy= GenerationType.IDENTITY)

@Column(name = "id")

private Integer id;

@Column(name = "fname")

private String fname;

public Mark() {

}

public Mark(Integer id) {

this.id = id;

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getFname() {

return fname;

}

public void setFname(String fname) {

this.fname = fname;

}

@Override

public int hashCode() {

int hash = 0;

hash += (id != null ? id.hashCode() : 0);

return hash;

}

@Override

public boolean equals(Object object) {

// TODO: Warning - this method won't work in the case the id fields are not set

if (!(object instanceof Mark)) {

return false;

}

Mark other = (Mark) object;

if ((this.id == null && other.id != null) || (this.id != null && !this.id.equals(other.id))) {

return false;

}

return true;

}

@Override

public String toString() {

return "domain.Marks[ id=" + id + " ]";

}

}