第 1 行：将变量 score 的值赋值为 session 中的 score 的值
第 2 行：判断变量 score 的值是否大于等于 90，如果是，显示您的成绩优秀
第 3 行：判断变量 score 的值是否大于等于 60，如果是，显示您的成绩处于平均水平
第 4 行：如果变量 score 的值在 60 分以下，则显示您的成绩不及格

综合程序题1（共4题，40.0分）

2.（12.0分）下列Action中的execute()方法以非IoC方式访问Servlet，说明下列带标号语句的功能，并指出代码中两种方式设置数据的区别。

```
public String execute() throws Exception
{
    if(getUserName().equals("tester")&&getPassWord().equals("tester "))
    {
        HttpServletRequest request = ServletActionContexst.getRequest();    ①
        request.setAttribute("userName", this.userName);    ②
        request.setAttribute("passWord", this.passWord);

        HttpSession session = request.getSession();    ③
        session.setAttribute("userName", this.userName);    ④
        session.setAttribute("passWord", this.passWord);
        return SUCCESS;
    }else{
        return INPUT;
    }
}
```

答：

1 获取 HttpRequest 对象

2 把 userName 当做 key，this.userName 当做 value 放入 request 范围

3 通过 request 对象获取 session 对象

4 把 passWord 当做 key，this.passWord 当做 value 放入 session 范围

这两种方式设置的数据作用域范围不同：

request 中的数据只能在一次请求中使用，request 中的属性在一次请求后就会过期。session 中的数据可以在用户之间共享，建立在一次会话(用户和服务器之间的长连接)中，多次请求，直到用户退出会话

3.（10.0分）请编写一个登录页面对应的业务控制器，如果登录成功（用户名和密码正确，假设用户名和密码分别为“tester”、“tester”），同时设置tip的值为“登录成功！”，并返回SUCCESS。如果登录失败（用户名密码不正确），同时设置tip的值为“登录失败！”，并返回INPUT。

```
public class test1 extends actionsupport{
    private String userName;
    private String userPassName;
    private String tip;
    public String getUserName() {
        return userName;
    }
    public void setUserName(String userName) {
        this.userName = userName;
    }
    public String getUserPassName() {
        return userPassName;
    }
    public void setUserPassName(String userPassName) {
        this.userPassName = userPassName;
    }
    public String getTip() {
        return tip;
    }
    public void setTip(String tip) {
        this.tip = tip;
    }
    public String execute()throws Exception{

        请将此处代码补充完整
    }
}
```

```java
public String execute() throw Exception{
    if(userName.equals("tester") && password.equals("tester")) {
        setTip("登录成功！");
        return SUCCESS;
    }else {
        setTip("登录失败！");
        return INPUT;
    }
}
```

4.（10.0分）下面是前置通知的一个实例，请分析其功能，在①、②、③、④和⑤标号的位置补充完整配置文件及代码。

Reception.java:

```
package beforeadviceexample;
public interface Reception {
    void serveCustomer(String customerName);//为客户提供的服务方法
}

```

ConcreteReception.java:

```
package beforeadviceexample;
public class ConcreteReception implements Reception{
    public void serveCustomer(String customerName){//为客户服务
        System.out.println("我正在为客户服务："+customerName+"。");
    }
}

```

GettingBeforeAdvice.java:

```
package beforeadviceexample;
import java.lang.reflect.Method;
import org.springframework.aop.MethodBeforeAdvice;
public class GettingBeforeAdvice implements MethodBeforeAdvice{
    public void before(Method method,Object[] args, Object target) throws Throwable{
        String customerName=(String)args[0];
        System.out.println("很高心见到您："+customerName+",请跟我来！");
    }
}

```

//配置文件（applicationContext.xml）

```
…
<beans>
    <bean id="gettingBeforeAdvice"
        class="beforeadviceexample.GettingBeforeAdvice"></bean>
    <!--使用 Spring 代理工厂配置一个代理-->
    <bean id="reception"
        class="org.springframework.aop.framework.ProxyFactoryBean">
        <!--指定代理接口，如果有多个接口可以使用 list 元素指定-->
        <property name="proxyInterfaces"  value=" ① "></property>
        <!--指定通知-->
        <property name="interceptorNames" value=" ② "></property>
        <!--指定目标对象-->
        <property name="target" ref=" ③ "></property>
    </bean>
    <bean id="target" class="beforeadviceexample.ConcreteReception"></bean>
</beans>

```

```
//测试类 Test.java:
package beforeadviceexample;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.FileSystemXmlApplicationContext;
public class Test {
    public static void main(String[] args) {
        ApplicationContext ac=new
FileSystemXmlApplicationContext("src/beforeadviceexample/applicationContext.xml");
        Reception reception=(  ④  )ac.getBean(" ⑤ "); //生成代理对象
        reception.serveCustomer("小强"); //接待人员接到客户，提供服务
    }
}
```

① beforeadviceexample.Reception

② gettingBeforeAdvice

③ target

④ Reception

⑤ reception

5.（12.0分）近年来，随着全球新能源汽车市场蓬勃增长，中国新能源汽车企业在10余年间实现了“弯道超车”，一跃成为新能源汽车产量连续7年居世界第一的全球新能源汽车强国。现开发一个基于Spring框架的新能源汽车仿真系统，其中定义了汽车类型NewEnergyVehicle及其配置的电池接口Battery，并使用“设置注入”方式进行依赖注入。

NewEnergyVehicle.java：

```
package nev;
public class NewEnergyVehicle{
    Battery battery;
    ① 请完成实体类的代码

}
```

Battery.java：

```
package nev;
public interface Battery{
    void work();
}
```

CATLBattery.java：

```
package nev;
public class CATLBattery imlements Battery{
  public void work(){
    System.out.println("宁德时代三元锂电池正在驱动车辆");
  }
}
```

BYDBattery.java：

applicationContext.xml：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN 2.0//EN"
"http://www.springframework.org/dtd/spring-beans-2.0.dtd" [
<!ENTITY contextInclude SYSTEM "org/springframework/web/context/WEB-
INF/contextInclude.xml">]>
<beans>
    ② 请完成 applicationContext.xml 配置文件的代码

</beans>
```

Test.java：

```
public class Test {
    public static void main(String[] args) {
        ApplicationContext ac=new
FileSystemXmlApplicationContext("src/ nev /applicationContext.xml");
        NewEnergyVehicle car1 = (NewEnergyVehicle)ac.getBean("BYDCar");
        NewEnergyVehicle car2 = (NewEnergyVehicle)ac.getBean("XiaopengCar");
        System.out.println("启动比亚迪汽车");
        car1.run();
        System.out.println("启动小鹏汽车");
        car2.run();
    }
}
```

输出：

```
启动比亚迪汽车
弗迪磷酸铁锂电池正在驱动车辆
启动小鹏汽车
宁德时代三元锂电池正在驱动车辆
```

(简答题 4分) 请按照“设置注入”方式在①处补全实体类的代码。

① NewEnergyVehicle.java

```java
public void setBattery(Battery battery) {
  this.battery = battery;
}
public void run() {
  battery.work();
}
```

第1小题　　**第2小题**

(简答题 8分) 请根据“设置注入”方式及终端输出文字在②处补全applicationContext.xml配置文件的代码。

② applicationContext.xml

```
<bean id="BYDBattery" class="nev.BYDBattery"/>
<bean id="CATLBattery" class="nev.CATLBattery"/>

<bean id="BYDCar" class="nev.NewEnergyVehicle">
  <property name="battery" ref="BYDBattery"/>
</bean>
<bean id="XiaopengCar" class="nev.NewEnergyVehicle">
  <property name="battery" ref="CATLBattery"/>
</bean>
```

6. (15.0分) 《易经·系辞》上有“两仪生四象，四象生八卦”的说法，中国传统哲学将世间万物看作是基本事物的反复组合。“增、删、查、改”是数据库的四种基本操作，通过组合这些基本的操作，可以实现各种各样复杂的任务。基于上一题的电影信息管理系统的数据库定义和实体类定义，请补全代码完成以下的查询和新增操作。

FilmDao.java:

```
package DAO;
import PO.*;
import java.util.*;
public class FilmDao {
   pulic void showFilmsGThan80(){
      Configuration cfg=new Configuration().configure("hibernate.cfg.xml");
      SessionFactory sf=cfg.buildSessionFactory();
      Session session=sf.getCurrentSession();

      ①完成执行 HQL 查询操作的代码

   }

   public void addFilm(Film film) {
      Configuration cfg=new Configuration().configure("hibernate.cfg.xml");
      SessionFactory sf=cfg.buildSessionFactory();
      Session session=sf.getCurrentSession();

      ②完成执行新增电影操作的代码

      }

   public void addCategory(Category category) {
      Configuration cfg=new Configuration().configure("hibernate.cfg.xml");
```

```
        SessionFactory sf=cfg.buildSessionFactory();
        Session session=sf.getCurrentSession();

        ③完成执行新增类别操作的代码

    }
}
```

FilmTest.java:

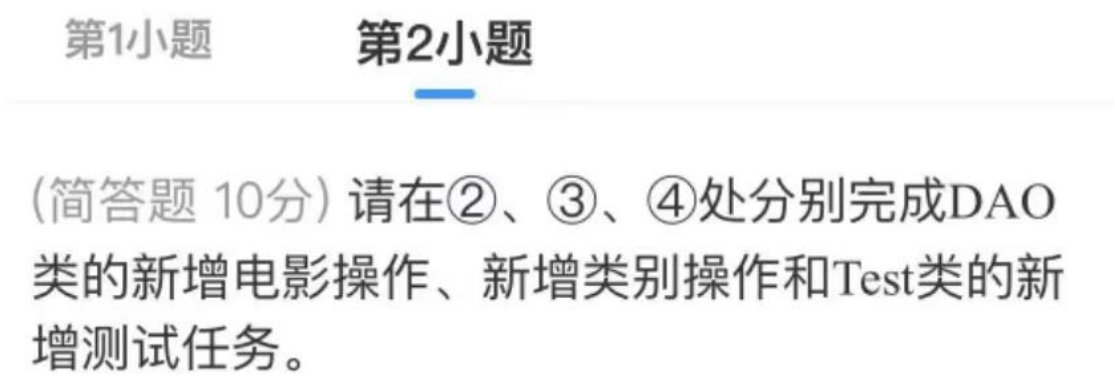
```
import PO.*;
import DAO.*;
import java.util.*;
public class FilmTest {
    public void test() {
        Film film = new Film(new Integer(1001), "长津湖", 85.0);
        Category category1 = new Category(new Integer(301), "战争片");
        Category category2 = new Category(new Integer(401), "国产片");

        ④完成新增测试任务，在数据库中新增电影《长津湖》，新增电影类别"战争片"和"国
          产片"，并建立《长津湖》与两个类别的关联关系

        // 显示系统中所有票价大于 80 元的电影名
        dao.showFilmsGThan80();
    }
}
```

第1小题 第2小题

(简答题 5分) 请在①处完成DAO类的查询操作，要求使用HQL语句查询系统中所有票价大于80元的电影，并将电影的名字输出到终端上。

① 执行HQL 查询操作

```
String sql = "from Film where price > 80";
try{
  Query query = sf.createQuery(sql);
  List<Film> films = query.list();
  for(Film film : films) {
    System.out.println(film.getName());
  }
} catch(HibernationException ex) {
  // 注意这里不能直接向上throw new HibernationException,
  // 这样会导致方法中也需要向上 throw HibernationException
  // 所以直接打印异常堆栈信息
  ex.printStackTrace();
} finally {
  session.close();
}
```



答：

② 执行新增电影操作

```
try{
  Transaction tx = session.beginTransaction();
  session.saveOrUpdate(film);
  tx.commit();
} catch (HibernateException ex) {
  ex.printStackTrace();
} finally {
  session.close();
}
```

3

③ 执行新增类别操作

```
try{
  Transaction tx = session.beginTransaction();
  session.saveOrUpdate(category);
  tx.commit();
} catch (HibernateException ex) {
  ex.printStackTrace();
} finally {
  session.close();
}
```

4 题目不全，关系未知

④

```
FilmDao filmDao = new FilmDao();
filmDao.addFilm(film);
filmDao.addCategory(category1);
filmDao.addCategory(category2);
```

7.（13.0分）近年来，国产电影越来越受到中国观众们的欢迎，扭转了以往“外国大片”主导票房份额的局面。国产电影的优势主要体现在题材多样化、技术快速追赶和更贴近民生等方面，成为国产电影票房的保证。现针对国内某电影院的需求，开发专用的电影信息管理系统，其中电影表t_film和类别表t_category分别给出电影的信息和电影所属类别的信息。由于电影和类型是多对多的关联关系，因此定义一个单独的关联表t_film_category来记录这种关联关系。下面分别给出了三个表的设计方案。

t_film 表

名	类型	长度	小数点	非空	键
film_id	int	4		√	√
film_name	varchar	50			
ticket_price	decimal	4	2		

t_category 表

名	类型	长度	小数点	非空	键
category_id	int	4		√	√
category_name	varchar	50			

t_film_category 表

名	类型	长度	小数点	非空	键
id	int	4		√	√
fid	int	4			film_id 外键
cid	int	4			category_id 外键

Film.java：

```
package PO;
```

①请完成 **Film** 类的定义

}

Category.java：

```
package PO;
import java.util.*;
public class Category {
    private Integer categoryId;
    private String categoryName;
    private Set<Film> films = new HashSet<Film>();

    public Category(Integer id, String name) {
    this.categoryId = id;
    this.categoryName = name;
    }
    public Integer getCategoryId() {
        return categoryId;
    }
    public void setCategoryId(Integer categoryId) {
        this.categoryId = categoryId;
    }
    public String getCategoryName() {
        return categoryName;
    }
    public void setCategoryName(String categoryName) {
        this.categoryName = categoryName;
    }
    public Set<Film> getFilms() {
        return films;
    }
    public void setFilms(Set<Film> films) {
        this.films = films;
    }
}
```

Film.hbm.xml：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC
 "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
 "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="PO">
<class name= "Film" table= "t_film">
    <id column="film_id" name="filmId" type= "integer">
      <generator class="increment"/>
    </id>
    <property column="film_name" name="name" type= "string"/>
    <property name="price" type= "float"/>
      <column name="ticket_price" sql-type='decimal(6, 2)'/>
    </property>
    <--多对多关联映射-->
    <set table="t_film_category" name="categories" cascade="save-update"
inverse="false">
        <key column="fid"></key>
        <many-to-many column="cid" class="PO.Category"></many-to-many>
    </set>
</class>
</hibernate-mapping>
```

Category.hbm.xml：

```
< ?xml version="1.0" encoding="UTF-8"? >
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping package= "PO">
```

②请完成 **Category.hbm.xml** 映射文件的定义

```
</hibernate- mapping>
```

第1小题　　第2小题

(简答题 7分) 请在①处根据数据表的结构完成实体类的定义。

① Film类

```
// 注意和 Film.hdm.xml 的映射关系对应
public class Film {
  private Integer filmId;
  private String name;
  private Float price;
  private Set<Category> categories=new HashSet<>;

  public Integer getFilmId() {
    return filmId;
  }
  public void setFilmId(Integer filmid) {
    this.filmId=filmId;
  }
  public String getName() {
    return name;
  }
  public void setName(String name) {
    this.name=name;
  }
  public Float getPrice() {
    return price;
  }
  public void setPrice(Float price) {
    this.price=price;
  }

  public Set<Category> getCategories() {
    return categories;
  }
  public void setCategories(Set<Category> categories) {
    this.categories=categories;
  }
}
```

第1小题　　**第2小题**

（简答题 6分）请在②处按照多对多关联关系完成映射文件的定义。

② Category.hdm.xml 映射文件

```
<class name="Category" table="t_category" >
  <id column="category_id" name="categoryId" type="integer">
    <generator class="increment"/>
  </id>
  <property columb="category_name" name="categoryName" type="string"/>
  <!--多对多关系映射    参考上面的 Film.hdm.xml-->
  <set table="t_film_category" name="films" cascade="save-update" inverse="false">
    <key column="cid"></key>
    <many-to-many column="fid" class="PO.Film"></many-to-many>
  </set>
</class>
```

系统分析设计题（共1题，20.0分）

8.（20.0分）为了深入学习宣传贯彻党的二十大精神，充分发挥高校思政育人功能，使得青年一代把党的二十大精神内化于心，外化于行，坚定信念，脚踏实地，将自身理想与国家前途命运紧密结合，某大学要开发“聚焦党的二十大”网站。该网站主要包含注册登录、党的二十大精神宣讲模块、党的二十大知识答题模块、党的二十大知识对抗赛模块、党的二十大热议论坛等。请用本门课程所学的框架对该网站进行系统分析和设计。

第1小题　第2小题

（简答题 8分）请描述该系统的功能需求（具体说明系统应有哪些角色，每一种角色应有哪些功能）。

答：该网站具有的功能需求描述如下：

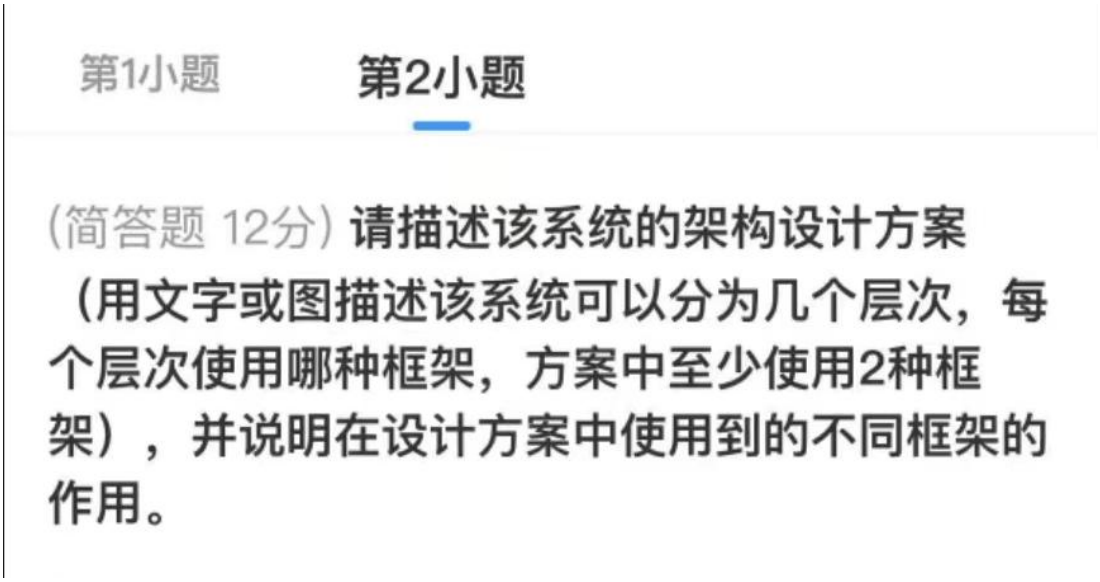
1 系统注册登录，用户可以注册位系统会员，然后可以使用注册后的账号和密码登录系统

2 十二大精神宣传，管理员可以发布精神宣传的视频或者文稿，普通用户可以浏览观看

3 十二大知识答题模块，管理员可以根据十二大有关精神和内容，创建知识问答题库

4 十二大知识对抗赛模块，会员可以参加知识问答对抗赛，并设置一定的奖励机制

5 十二大热议论坛模块 ，会员可以参加十二大热议，发表个人观点，其他人也可以回复，把回复最多的置顶



该系统采用 SSH 框架设置，具体分层为：

1 视图层，采用 jsp 技术完成，主要完成数据的展示和录入，包括会员和管理员

2 控制层，采用 struts2+spring 来完成，主要完成业务功能的实现，以及每次用户操作完成的视图跳转

3 持久层，采用 hibernate 框架来完整，主要完成数据的持久化，包括增删改查四大类操作

该框架还可以使用 SpringBoot 来实现，具体分层为：

1 前端展示层，采用 Thymeleaf 框架来完成，主要是数据的录入和展

示，提供用户操作界面

2 业务逻辑层，采用 SpringBoot 来完成，主要完成业务逻辑实现以及页面跳转

3 数据访问层，采用 mybatis 来完成，主要完成数据的新增，删除，修改和查询操作。