# JAVA编程进阶上机报告

****

**学 院 智能与计算学部**

**专 业 软件工程**

**班 级 软件工程2班**

**学 号 3619058914**

**姓 名 周陈铮**

1. **实验要求**

**详细见实验报告**

1. **源代码**

**定义注解**

@Target(ElementType.***FIELD***)  
@Retention(RetentionPolicy.***RUNTIME***)  
**public** @**interface** Column {  
 String value();  
}

@Target(ElementType.***TYPE***)  
@Retention(RetentionPolicy.***RUNTIME***)  
**public** @**interface** Table {  
 String value();  
}

**定义实体类**

**package** entity;  
  
**import** annotation.Column;  
**import** annotation.Table;  
  
@Table( **"user"**)  
**public class** User {  
 @Column(**"userId"**)  
 **private int id**;  
 @Column(**"username"**)  
 **private** String **userName**;  
 @Column(**"email"**)  
 **private** String **email**;  
 @Column(**"telephone"**)  
 **private** String **telephone**;  
 @Column(**"age"**)  
 **private int age**;  
 省去了get，set方法

**关键类实现**

**package** entity;  
  
**import** annotation.Column;  
**import** annotation.Table;  
  
**import** javax.jws.Oneway;  
**import** java.lang.annotation.Annotation;  
**import** java.lang.reflect.Field;  
**import** java.util.List;  
  
**public class** SqlUtilImpl **implements** ISqlUtil {  
 **private** Class **c**;  
 **private** Column **id**;  
 **private** Column **userName**;  
 **private** Column **email**;  
 **private** Column **telephone**;  
 **private** Column **age**;  
  
 @Override  
 **public** String query(User user) {  
 StringBuilder sb = **new** StringBuilder();  
 **c** = user.getClass();  
 **boolean** table\_flag = **c**.isAnnotationPresent(Table.**class**);  
 **if**(!table\_flag){  
 **throw new** NullPointerException();  
 }  
 Table table = (Table)**c**.getAnnotation(Table.**class**);  
 sb.append(**" SELECT \* FROM "**).append(table.value());  
 **try** {  
 Field fieldId = **c**.getDeclaredField(**"id"**);  
 fieldId.setAccessible(**true**);  
 **id** = fieldId.getDeclaredAnnotation(Column.**class**);  
 **int** idValue = (Integer)fieldId.get(user);  
   
 Field fielduserName = **c**.getDeclaredField(**"userName"**);  
 fielduserName.setAccessible(**true**);  
 **userName** = fielduserName.getDeclaredAnnotation(Column.**class**);  
 String userNameValue = (String)fielduserName.get(user);  
 **if**(idValue != 0) {  
 sb.append(String.*format*(**"WHERE %s = %d"**, **id**.value(), idValue));  
 }**else** {  
 sb.append(String.*format*(**"WHERE %s LIKE %s"**, **userName**.value(), userNameValue));  
 }  
  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
  
 **return** sb.toString();  
  
 }  
  
 @Override  
 **public** String insert(User user) {  
 StringBuilder sb = **new** StringBuilder();  
 **c** = user.getClass();  
 Table table = **null**;  
 **if**(**c**.isAnnotationPresent(Table.**class**)) {  
 table = (Table) **c**.getAnnotation(Table.**class**);  
 }  
 **try** {  
 Field fieldEmail = **c**.getDeclaredField(**"email"**);  
 fieldEmail.setAccessible(**true**);  
 **email**= fieldEmail.getDeclaredAnnotation(Column.**class**);  
 String emailValue = (String)fieldEmail.get(user);  
  
 Field fielduserName = **c**.getDeclaredField(**"userName"**);  
 fielduserName.setAccessible(**true**);  
 **userName** = fielduserName.getDeclaredAnnotation(Column.**class**);  
 String userNameValue = (String)fielduserName.get(user);  
  
 Field fieldtelephone= **c**.getDeclaredField(**"telephone"**);  
 fieldtelephone.setAccessible(**true**);  
 **telephone** = fieldtelephone.getDeclaredAnnotation(Column.**class**);  
 String telephoneValue = (String)fieldtelephone.get(user);  
  
 Field fieldage= **c**.getDeclaredField(**"age"**);  
 fieldage.setAccessible(**true**);  
 **age** = fieldage.getDeclaredAnnotation(Column.**class**);  
 **int** ageValue = (**int**)fieldage.get(user);  
 sb.append(String.*format*(**"INSERT INTO %s ( %s, %s, %s, %s) VALUES"**, table.value(), **userName**.value(), **email**.value(), **telephone**.value(), **age**.value()));  
 sb.append(String.*format*(**" ( %s, %s, %s, %s ) "**, userNameValue, telephoneValue, emailValue, ageValue));  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
  
 **return** sb.toString();  
 }  
  
 @Override  
 **public** String insert(List<User> users) {  
 StringBuilder sb = **new** StringBuilder();  
 User user = users.get(0);  
 **c** = user.getClass();  
 Table table = **null**;  
 **if**(**c**.isAnnotationPresent(Table.**class**)) {  
 table = (Table) **c**.getAnnotation(Table.**class**);  
 }  
  
 **int** value = 0;  
 **for**(User user1: users) {  
 **try** {  
 Field fieldEmail = **c**.getDeclaredField(**"email"**);  
 fieldEmail.setAccessible(**true**);  
 **email** = fieldEmail.getDeclaredAnnotation(Column.**class**);  
 String emailValue = (String) fieldEmail.get(user);  
  
 Field fielduserName = **c**.getDeclaredField(**"userName"**);  
 fielduserName.setAccessible(**true**);  
 **userName** = fielduserName.getDeclaredAnnotation(Column.**class**);  
 String userNameValue = (String) fielduserName.get(user);  
  
 Field fieldtelephone = **c**.getDeclaredField(**"telephone"**);  
 fieldtelephone.setAccessible(**true**);  
 **telephone** = fieldtelephone.getDeclaredAnnotation(Column.**class**);  
 String telephoneValue = (String) fieldtelephone.get(user);  
  
 Field fieldage = **c**.getDeclaredField(**"age"**);  
 fieldage.setAccessible(**true**);  
 **age** = fieldage.getDeclaredAnnotation(Column.**class**);  
 **int** ageValue = (**int**) fieldage.get(user);  
 **if**(value == 0) {  
 sb.append(String.*format*(**"INSERT INTO %s ( %s, %s, %s, %s) VALUES"**, table.value(), **userName**.value(), **email**.value(), **telephone**.value(), **age**.value()));  
 }  
  
 sb.append(String.*format*(**" ( %s, %s, %s, %s ), "**, userNameValue, telephoneValue, emailValue,ageValue));  
 value++;  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 sb.deleteCharAt(sb.lastIndexOf(**","**));  
 **return** sb.toString();  
 }  
  
 @Override  
 **public** String delete(User user) {  
 StringBuilder sb = **new** StringBuilder();  
 **c** = user.getClass();  
 **boolean** table\_flag = **c**.isAnnotationPresent(Table.**class**);  
 **if**(!table\_flag){  
 **throw new** NullPointerException();  
 }  
 Table table = (Table)**c**.getAnnotation(Table.**class**);  
 **try** {  
 Field fieldId = **c**.getDeclaredField(**"id"**);  
 fieldId.setAccessible(**true**);  
 **id** = fieldId.getDeclaredAnnotation(Column.**class**);  
 **int** idValue = (Integer)fieldId.get(user);  
 sb.append(String.*format*(**"DELETE FROM %s WHERE %s = %d"**, table.value(), **id**.value(), idValue));  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
  
 **return** sb.toString();  
  
 }  
  
 @Override  
 **public** String update(User user) {  
 StringBuilder sb = **new** StringBuilder();  
 **c** = user.getClass();  
 **boolean** table\_flag = **c**.isAnnotationPresent(Table.**class**);  
 **if**(!table\_flag){  
 **throw new** NullPointerException();  
 }  
 Table table = (Table)**c**.getAnnotation(Table.**class**);  
 sb.append(String.*format*(**"UPDATE %s SET "**,table.value()));  
 **try** {  
 Field[] fields = **c**.getDeclaredFields();  
 Object idValue = **null**;  
 Object value = **null**;  
 String key = **null**;  
 **for**(Field field: fields){  
 field.setAccessible(**true**);  
 key = field.getDeclaredAnnotation(Column.**class**).value();  
  
 value = field.get(user);  
 **if**(((!key.equals(**"userId"**)) && ( value **instanceof** Integer ? (Integer) value != 0 : value != **null**))){  
 sb.append(**id**.value() + **" = "** + value);  
 }  
 **if**(key.equals(**"userId"**)){  
  
 **id** = field.getDeclaredAnnotation(Column.**class**);  
  
 idValue = value;  
 }  
  
 }  
 sb.append(String.*format*(**" WHERE %s = %d"**,**id**.value(), idValue));  
  
  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 **return** sb.toString();  
 }  
  
  
}

1. **实验结果**

