**第46届世界技能大赛移动应用开发项目中国集训队**

**集训日志（选手） 朱姚飞**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **9-26** | **指导老师** | **马平川** |
| **训练任务** | **Sqlite使用** | | | |
| **训练内容：**  练习 SQLite3 CLI ,Sqlite的各种API 和 SQL 语句使用  **训练要求：**  熟悉 Sqlite的使用和Sqlite3 adb 的调试方法  **过程记录：**  class MyDBHelper(context: Context, factory: SQLiteDatabase.CursorFactory?) : SQLiteOpenHelper(context, "myDb", factory, 1) {   override fun onCreate(p0: SQLiteDatabase?) {  Log.i("autopDev", "create DB");  p0?.execSQL("CREATE Table `${TableMap.user}`( ${UserModel.Column.id} integer primary key autoincrement, ${UserModel.Column.username} text not null , ${UserModel.Column.passwd} text not null , ${UserModel.Column.role} integer not null );");  }   override fun onUpgrade(p0: SQLiteDatabase?, p1: Int, p2: Int) {  }  }  class TableMap {  companion object {  val user = "user";  } }   class UserModel {   class Column {  companion object {  val id = "id";  val username = "username";  val passwd = "passwd";  val role = "role";  }  }   companion object {  fun insertUser(db: SQLiteDatabase, username: String, passwd: String, role: Int): Boolean {  val result = db.rawQuery("SELECT \* from `${TableMap.user}` WHERE `${Column.username}`=?", arrayOf(username));  return if (result.count == 0) {  val values = ContentValues()  values.put(Column.username, username)  values.put(Column.passwd, passwd)  values.put(Column.role, role)  db.insert(TableMap.user, null, values)  result.close();  true  } else {  result.close();  false  }  }   fun testUser(db: SQLiteDatabase, username: String, passwd: String, role: Int): Boolean {  val result = db.rawQuery("SELECT \* from `${TableMap.user}` WHERE `${Column.username}`=? AND ${Column.passwd}=? AND ${Column.role}=${role}", listOf(username, passwd).toTypedArray());  val count = result.count  result.close()  return count == 1;  }   } }  **分析总结：**  学习和训练了SQLITE的使用 和ADB 调试SQLITE 数据库的方法 | | | | |

**填写人：朱姚飞**