

本来不打算写经验的。但是感觉挺不好意思的，因为我的脚下算是踩了一些人头。

个人情况：

本科湖南衡阳南华大学，作为一个在 12 年才升一本的大学，应该是一本里最差的了。数学类的信息与计算科学专业，但是可以把我看成计算机专业的，因为计算机的大部分课程我们专业都上了。其实作为数学专业，我数学挺强的。但是种种原因，导致跪了，数学 79 分。初试总分 311，算上和我重分的那个人，我和他应该是并列倒数第二(后面算了下每个人的分数，不是倒数第二，是并列倒数第三)。不过我代码能力不错，复试成绩是正过来的第二(93.15)，和初试分数加权后就排到第 6 了。踩了 10 个人左右。

初试的渣渣分数就不说了，只说一些导致我数学懵逼的注意事项，1.不要天天待寝室里复习(从九月底开始，我一直没怎么出去见过太阳光，吃饭点的外卖)；2.注意作息(我每天是十一点左右起床点外卖，吃完中饭，十二点半左右开始复习，晚上 12 点上床，每天复习 9~10 个小时左右，直到考前我想调作息都调不动，考试的两晚都没睡着)；3.在考前仍然要每天做数学(考前一周没碰数学)。总结原因就是，考试时手生，然后作息没调好，脑袋好像没醒，看着题目思维都不转，再加之待寝室久了，一下子出寝室还跑这么远，肚子一直不舒服(考试中途上了个大号)。幸亏下午考 408 之前，先往厕所跑了一趟，解决了肚子问题，神清气爽，408 才没跪太惨。

其实上面一段只是我想吐一吐苦水而已，也没什么人会像我这样备考。直接说一下复试吧。

机试70%，我报考华工的一个很大理由，只要能进复试，如果不出意外(像数学那样的不是意外的意外)我就能保证我考上。没出院线之前，一直没准备，直到3月7号，群里打电话问到院线，我才开始准备。虽然没学C#这门课，但是我毕业设计选择了用WPF，也算是为复试做了一点小打算，万一上线了，可以不用从头学C#。刚开始本来打算用WPF来复试，但是看见论坛里说机子慢，版本低，各种乱七八糟的问题。就还是用相对低端但速度快一些的WinForm来练习了。从09到15，没有10，练了一遍，这一遍数据呈现用的DataGridView().DataSource=DataTable，比较方便，但是怕出问题。第二遍全部用ListView实现，比DataGridView要多几分钟(写个填充ListView的通用函数)，而且目标框架改为.NetFramework 2.0，这次没有练习11年的，11年4个表都要维护，内容太多。第一次6年，第二次5年，总共练习了11遍(一天一遍，差不多到第五遍的时候，不算数据库建立插入，其余部分一般一个小时，具体一点，数据表维护半小时，其他半小时)。

到了18号，合租的小伙伴要去学校，我就也去了，就背了个书包，没有带电脑，20号晚上找小伙伴的电脑练习了一遍14年的题目。第二天下午就考完了。考试过程有点懵，首先数据库连接不上，开服务，开协议，各种能开的都开，还是不能连上，开考一两分钟后还没连好，只好把服务器名字改了(之前用的是"，后面改成浏览更多里面的数据库引擎的第一个了，这个里面带上了实例名，应该是他们安装的时候改了实例名，所以"连接不上，必须要带上实例名)，连接正常后，就开始数据库了，数据库建立与插入15分钟左右(比平时快几分钟，我平时用的输入法是"智能ABC"，我考试用的是"微软拼音-简捷 2010")，插入时比平时练习时多了一步，就是对着insert语句又检查了一遍数据看有没有打错。主键，外键，级联，非空约束，唯一约束，检查约束这些都是基本东西，也没什么说的。

这里还有个小程序，因为维护的是学生表，增加学生的时候又要对另一张表的班级人数进行操作，要用到触发器(或者在C#中用事务)。但是我并没有在增加学生的时候增加对应班级人数(因为学生表中的记录并没有这么多，也就是学生记录数量没有与班级表中的班级人数对应。两者对应关系并不明确，增加的学生记录到底是在班级中已存在的学生(只是没记录)，还是新插班进来的学生，并不清楚。我就没考虑这个小程序)。这样在数据表维护时也不会遇到插入学生时班级人数超过30人。

因为习惯从数据表维护开始，我就先扫题，扫到第三题，看见维护学生表几个字，然后就开始做了，碰到第一个问题，入学年份，从ListView的当前行中取出来的DateTime类型的时间文本，Convert.ToDateTime(文本);竟然会格式解析出错。改来改去改不好，最后想到了不在ListView里取DateTime文本，仿照DataGridView一样，直接根据主键在数据库里取DateTime对象(即用SqlCommand().ExecuteScalar()取出对应的Object)，然后强转为C#中的DateTime再进行处理。就这一个问题就改了我十几二十分钟。

做完学生表维护，统计部分，没什么难度。看到第一题就傻眼了，然后再回头看一下第二题和第三题，然后我才发现这是08年的题目。之前没仔细看题，以为和普通年份的差不多，慢悠悠做下来，还把每个控件都命名规范。到这里，就开始急了，因为08年题目数据不全，我就没做，只看过题目，想过用checkbox来做，但是一直没做。我首先在SQL Server里把含三个条件的查询的语句写出来。然后把变化的部分挖出，也就是三个条件，我就想到了拼接sql语句，但是我一直拼不过字符串(我写Sql语句都是习惯了参数化条件防止Sql注入的，一直拼不过sql语句)，中间一老出错，少逗号，多逗号，少空格，多换行什么的。错误的时候还不跳转到错误的地方，程序直接无响应。我只好断点调试，在执行sql语句之前断下，取出sql语句，自己到SQL Server中运行调错，调了几次后，顺利通过，最后还有一个条件，按系编号降序，专业名称升序，学生名降序，按多个字段排序我不记得了，第一次试的是order by

deptName desc order by className asc order by SName desc 报错order by 附近错误，第二次试的是把后面的两个 order by 去掉，order by deptName desc className asc SName desc 继续错误，试了几次，最后一次没报错了order by deptName desc, className asc, SName desc 到这个时候就剩下2分钟了，快速写完readme.txt，再写Info.doc还没写完就到时间了，但是我还是站着写了一两分钟，把它写完了，虽然学长扯了我一下，但是看到我不是在写代码，也就没太阻止我。考完还是有点忐忑的，毕竟没怎么测试。

我的机试流程大概就是这样了，可以看出，尽管复试前我感觉不会有什么太大意外，但是中间还是有这么多曲折，大家练习还是要保证在一个半小时以内做完。这样的话，在考试中遇到没处理过的问题或突发状况时，能够有一个小时来处理。每年的题目练习时间应该为：09 和 14 年 1 小时 30 分；11 年 3 小时；12，13 和 15 年一小时 20 分；16 年 1 小时 50 分。

再说一下关于机试的练习：

想看书可以在图书馆借<<C#高级编程>>，我找的书里面就这本书写的比较好，写的内容并不高级，可以算是入门书籍，只是特别厚，有一千五百页左右，想看视频可以搜杨中科的视频，我的做毕业设计时就用的这本书和他的视频。

一星期可以解决 C#语法，ado.net 操作，没学过数据库的还要几天时间看数据库。这些都没学过的，但动手写过代码写得多一点的人，10 天以内搞定，没怎么写过代码的最多 20 天。当然，这些都学过的同学就一个星期都不用就可以直接复试了。这天数也不是死的，你一天花的时间多，自然天数就缩短了，我每天花的时间在两小时以内。

练习的次数自己把握，满足我上面的时间就差不多了，只需要练习 09，11，16 年这三年的题目就行了，也可以再从 12 到 15 年里挑一年出来。练习这 4 年的题目就行了。

数据库连不上，可以直接举手，不要自己瞎折腾，浪费时间，我虽然折腾出来了，但是时间浪费了，也影响心情。数据库中建表，设主键，设外键(加上级联)，每个都非空约束，碰到非字符串类型的字段判断是否要用检查约束，根据一对多、多对多关系有时需要设置唯一性约束，其他东西，比如索引、视图、触发器，若题目没有明确要求可以不做(或者你考试时时间多，可以完善完善)。插入数据都写 SQL 语句。遇见关键字的字段名用[]括起来。数据库部分写完了，马上备份数据库。后面测试时随便增删改数据，最后要结束时，把数据库删了，重新从备份文件恢复，或者把表全删了，用之前写的 SQL 语句重新运行一遍，保证最后数据库的数据是初始模样。要是碰见 EXCEL 导入数据不会，可以在创建完表后，选择编辑前 200 行，然后在 EXCEL 中复制全部数据，到数据库中选中一行(必须选中一行，不能是选中一个单元格)Ctrl V 就全部数据复制过去了，一次一个表，比写 SQL 语句要快。

上面速度快点，不到 20 分钟就拿到 40 分了。

数据库辅助类 SqlHelper.cs 肯定是要第一步写的，我这个类中有五个静态函数，三个分别对应执行数据库的三个函数，两个用来填充界面数据，这两个其实是应该另外建一个 CommonHelp.cs 或者 Utils.cs 类的，但是我嫌麻烦就放在一起了。

//我的连接字符串是写在配置文件中的

//你们可以直接写在这里 connStr="server=.;integrated security=sspi;database=test2016;";

private static string connStr =

System.Configuration.ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;

//public static int ExecuteNonQuery(string,params SqlParameter[]);执行增删改

//public static Object ExecuteScalar(string,params SqlParameter[]);获取数据库中的单个对象

//public static DataTable ExecuteTable(string,params SqlParameter[]);执行查询，获得虚拟表

//public static void FillListView(ListView,DataTable,params string[]);用DataTable中的数据填充ListView，需要提供行头字符串，调用案例:SqlHelper.FillListView(listView1,dataTable1,"系编号","系名称");

//public static void FillComboBox(ComboBox,DataTable);用DataTable中的一列数据填充ComboBox

//不熟悉 SqlParameter 可以直接掉这个参数，把下面的 cmd.Parameters.AddRange(param);这句也删掉。

//直接拼接完 Sql 语句，传进来就是，其实不删也没事，params 参数可以不传参，直接 ExecuteNonQuery(sql);

```
public static int ExecuteNonQuery(string sql, params SqlParameter[] param) {  
    using (SqlConnection conn = new SqlConnection(connStr)) {
```

```

        conn.Open();
        using (SqlCommand cmd = conn.CreateCommand()) {
            cmd.CommandText = sql;
            cmd.Parameters.AddRange(param);
            return cmd.ExecuteNonQuery();
        }
    }
}

public static Object ExecuteScalar(string sql, params SqlParameter[] param) {
    using (SqlConnection conn = new SqlConnection(connStr)) {
        conn.Open();
        using (SqlCommand cmd = conn.CreateCommand()) {
            cmd.CommandText = sql;
            cmd.Parameters.AddRange(param);
            return cmd.ExecuteScalar();
        }
    }
}

public static DataTable ExecuteTable(string sql, params SqlParameter[] param) {
    using (SqlConnection conn = new SqlConnection(connStr)) {
        conn.Open();
        using (SqlCommand cmd = conn.CreateCommand()) {
            cmd.CommandText = sql;
            cmd.Parameters.AddRange(param);
            SqlDataAdapter adapter = new SqlDataAdapter(cmd);
            DataTable dt = new DataTable();
            adapter.Fill(dt);
            return dt;
        }
    }
}

public static void FillListView(ListView listView, DataTable dt, params string[] headers) {
    listView.Clear();
    listView.BeginUpdate();
    for (int i = 0; i < dt.Columns.Count; i++) {
        listView.Columns.Add(headers[i], listView.Width / headers.Length - 2);
    }
    for (int j = 0; j < dt.Rows.Count; j++) {
        ListViewItem item = new ListViewItem();
        item.Text = dt.Rows[j][0] + "";
        for (int k = 1; k < dt.Columns.Count; k++) {
            item.SubItems.Add(dt.Rows[j][k] + "");
        }
        listView.Items.Add(item);
    }
    listView.EndUpdate();
}

public static void FillComboBox(ComboBox cb, DataTable dt) {
    for (int i = 0; i < dt.Rows.Count; i++) {

```

```

        cb.Items.Add(dt.Rows[i][0]);
    }
}

```

按照顺序来，首先是数据表的 CRUD，我一般是三个 Button(增修删)，一个 ListView，增加和修改再弹出一个编辑窗体 EditForm，窗体传值通过 EditForm 的成员变量来传。有一个状态字段 public string STATE; 点击增加按钮时设置为“add”，点击修改按钮时设置为“modify”，在 EditForm 中 Load 时判断 STATE 是否为“modify”，是的话就通过成员变量对 EditForm 上的 TextBox, RadioButton, ComboBox 等控件进行数据回显。在点击保存时，首先判断是内容是否为空，if(TextBox.Text.Trim().Length<1|...){MessageBox.Show(“所有信息都不能为空!”);return;}，第二步要对数据判断是否满足数据库中的检查约束，程序中不要用 try catch 来解决可以通过逻辑判断解决的问题，我只在用 Convert 类进行类型转换时用 try catch，因为这里有可能用户输入的格式错误，不是可以通过逻辑判断出来的(不要记那些正则表达式，当你 int age=Convert.ToInt32(string); 出错时自然是用户输入的不是整数了，try catch 提示用户按正确格式输就行了，用户输对了，自然就通过了，再判断该整数是否在某个区间就行了，简单又方便)。第三步，判断是否 STATE 是否等于“add”，是就先检测主键是否重复，if(0<(int)(SqlHelper.ExecuteScalar(“select count(*) from Student where SNo=@SNo”,new SqlParameter(“@SNo”,txtSNo.Text.Trim())))){MessageBox.Show(“该学生编号已存在!”);return;}否则就可以插入了;如果 STATE 等于“modify”，则先判断当前修改行的 SNo 与 txtSNo.Text.Trim() 是否相等，不相等则同样要像上面一样检测主键是否重复(不能修改为已经存在的主键)，相等则直接更新该主键对应的行。当含有外键要修改时，外键字段可以使用 ComboBox 来选择，而不是让用户输入。输入的经常会是外键不存在，放在 ComboBox 中的就肯定是数据库中存在的。还有 TextBox 中的 MaxLength 要设置为你数据库中对应字段的长度，不然的话，当你在数据库中编号长度是 5，可是 TextBox 中输入时长度超出 5 时，执行修改或增加，会出现数据库字段截断错误。

统计和查询，查询一般要用到 ComboBox，统计一般有表连接，嵌套子查询，分组，排序，就是考 Select 语句，这个自己练就是了。今年的查询是复杂一点，但是如果做过一次，就不会有什么问题。现在要我再做一遍应该可以 1 小时 40 分钟左右做完。

上机考试的东西也不多，如果都练过就比较轻松，08 年的我没练过，但是速度比较快，还是能节省时间出来解决问题。

面试就是那些经验里常见的问题了，也没什么说的。再就是联系导师了，这个对我来说不知道是重要还是不重要，因为本科学校差，感觉联系了也没人要。反正我一开始没联系，老师们对本科学校还是有些歧视的，我是录取分数出来后打的电话，问老师有没有名额，每个老师一开始都没有回答名额问题，而都是第一个问：“你本科学校哪的？”，当我说完后，每个人都是说：“我这边人选已经定好了，你再找找其他老师吧”。最后我就没联系了，等着剩下的哪个老师，我再联系哪个老师。差不多到中午左右，我和一个伙伴再去看已经确认的名单，还剩两个老师，伙伴联系了中间一个，我就打电话给这最后的张老师了。虽然张老师也是对我没啥想法，但是毕竟一个萝卜一个坑，后来许老师又说，张老师维护了一个比较大的系统，要代码能力强的，我说我动手能力还不错，复试是第二名，许老师就直接拍板帮张老师签了。所以本科学校还行的就快点联系老师，可以占住名额，本科差的也不用担心没人要，努力复试，上了就有坑让你入。

就这样啦，上线了的都要努力复试，分高的不要懈怠，分高容易碾压，分低的不要气馁，分低也可翻盘。当然，最后就是选择也很重要，胆子要大，我们专硕这边都挺保守的，许多都不敢报学硕，所以今年专硕情况就有些特殊了，应该是算有些惨烈。