

《面向对象程序设计》课程设计

题目：小型公司工资管理系统设计

年级：20级计算机

开发组序号：第2组

开发组情况：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 姓名 | 学号 | 班级 | 分工 | 自评分 |
| 董福江 | 201911040033 | 2班 | 数据库数据传输 | A |
| 刘立源 | 202011040004 | 1班 | 制作操作界面 | B |
| 王伟业 | 202011040005 | 1班 | 制作操作界面 | B |
| 平征 | 202011040007 | 1班 | 数据库建立 | B |

指导教师：王倩

开发日期：2021.6.25

1概述

目的与意义：通过设计一个小系统，进一步熟悉c++中类的概念、类的封装和继承的实现方式。了解系统开发的需求分析、类层次设计、模块分解、编码测试、模块组装与整体调试的全过程，加深对C＋＋的理解与Visual C＋＋环境的使用；逐步熟悉程序设计的方法，并养成良好的编程习惯。

主要完成的任务：存储并管理公司四类员工的基本信息，输入各项信息后可将其以表格形式导出，并得到管理人员工资表。

解决的主要问题：数据库的建立与连接、对数据的存储与操作、用c++语言实现 对程序的封装和对数据库的操作。

人员分工：董福江负责数据库数据传输

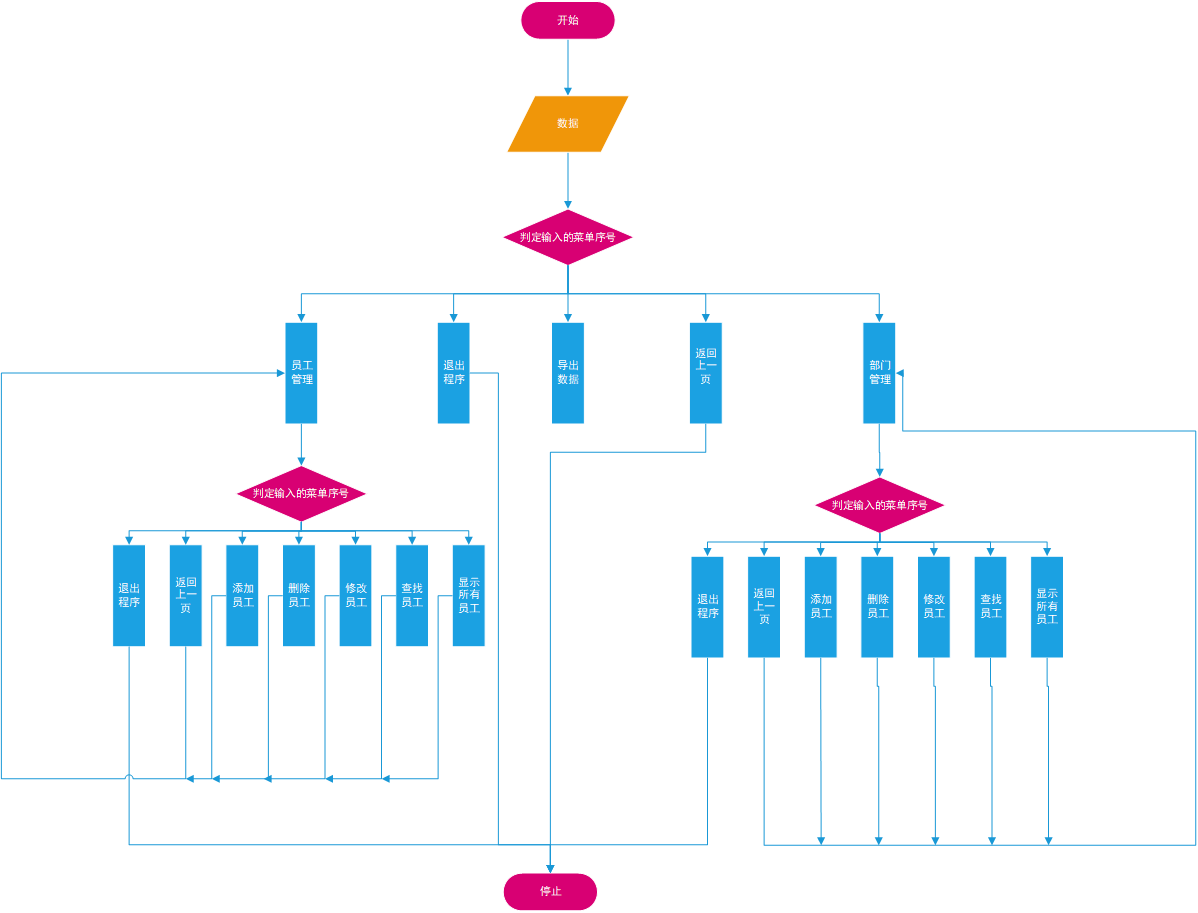
刘立源、王伟业制作操作界面

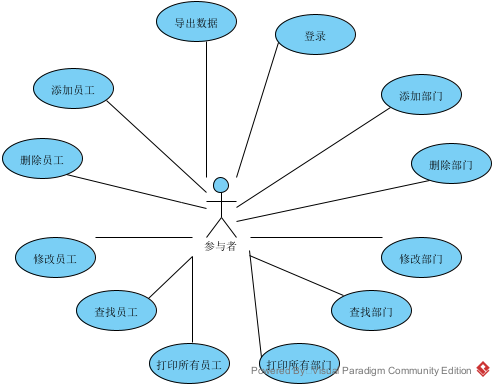
平征负责数据库的建立

开发计划、开发环境：使用MySQL数据库，并在vs2019上使用c++语言对其进 行操作。

2总体设计

图1软件的总体结构、模块功能及关系流程图





3详细设计

图2模块内部流程图

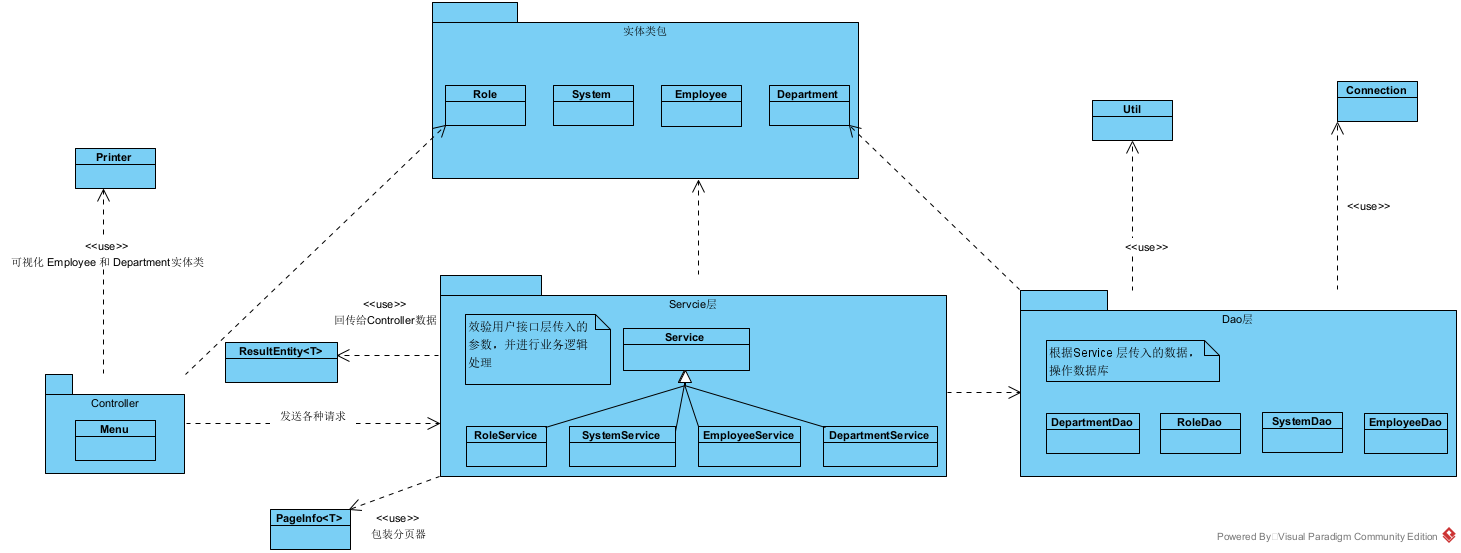
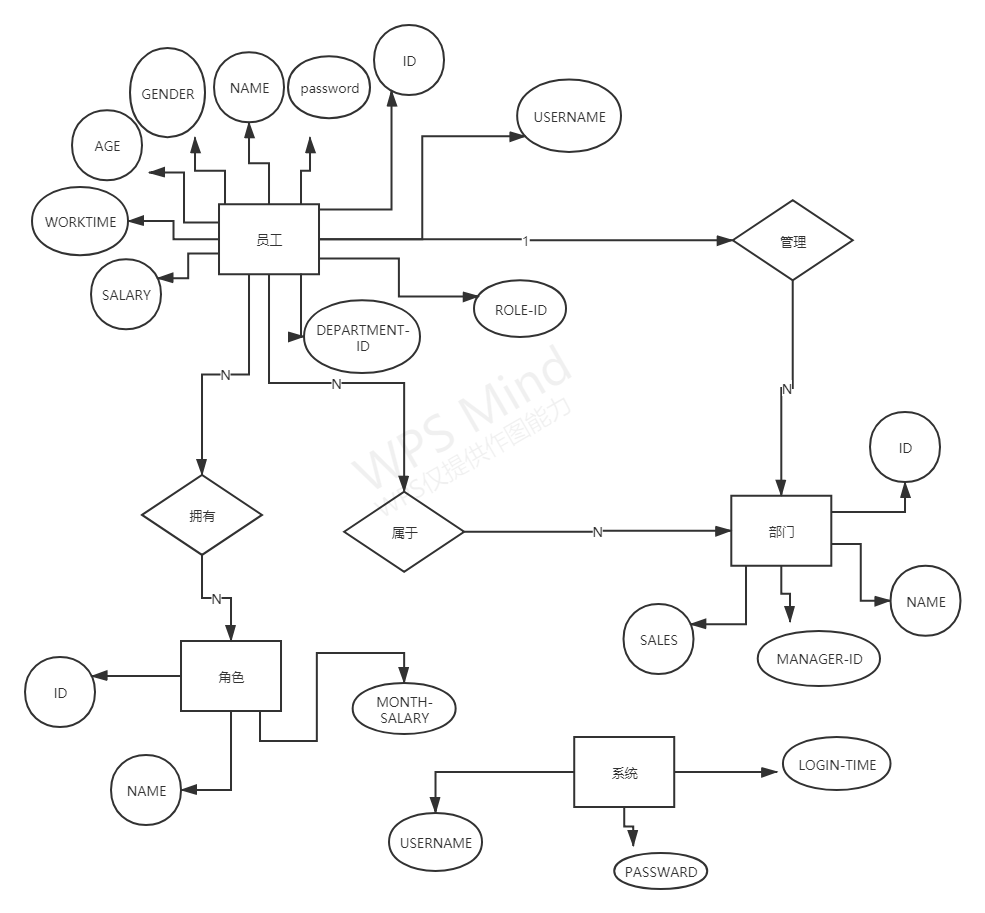


图2-1



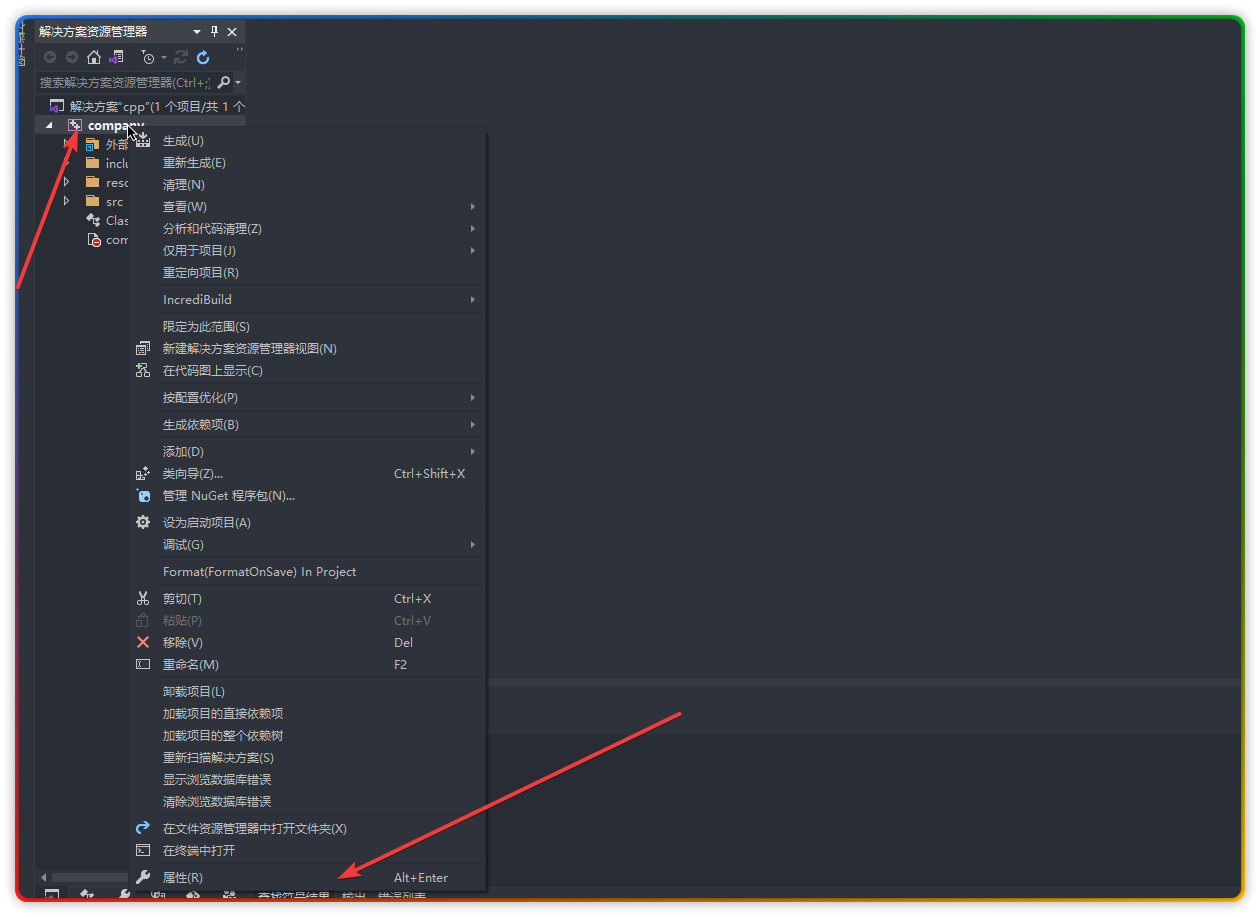
4编码设计

1开发环境的设置和建立

软件：VS 2019、Mysql 8

第三方库：[JSON for Modern C++](https://github.com/nlohmann/json)

1. 安装Mysql 8、VS2019、Connector/C++
   1. 使用Mysql installer 安装；
2. 引入 Connector/c++ 头文件和 mysqlcppconn8.lib、mysqlcppconn8-2-vs14.dll
   1. 图3引入include头文件



* 1. 图4引入include头文件

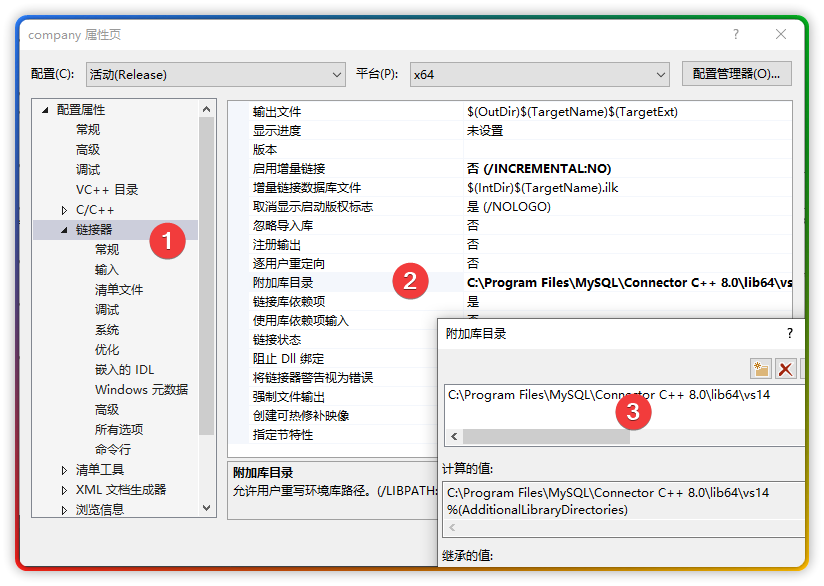


图5引入 mysqlcppconn8.lib

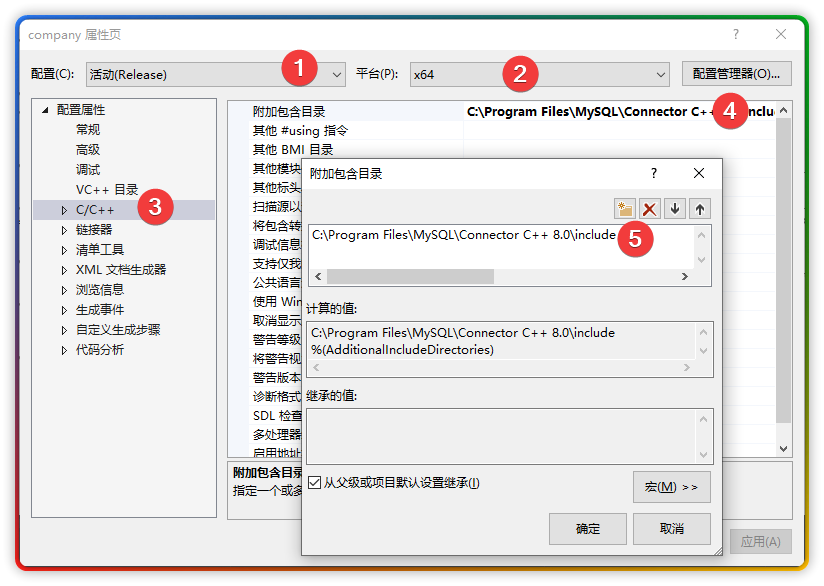


图6引入 mysqlcppconn8.lib

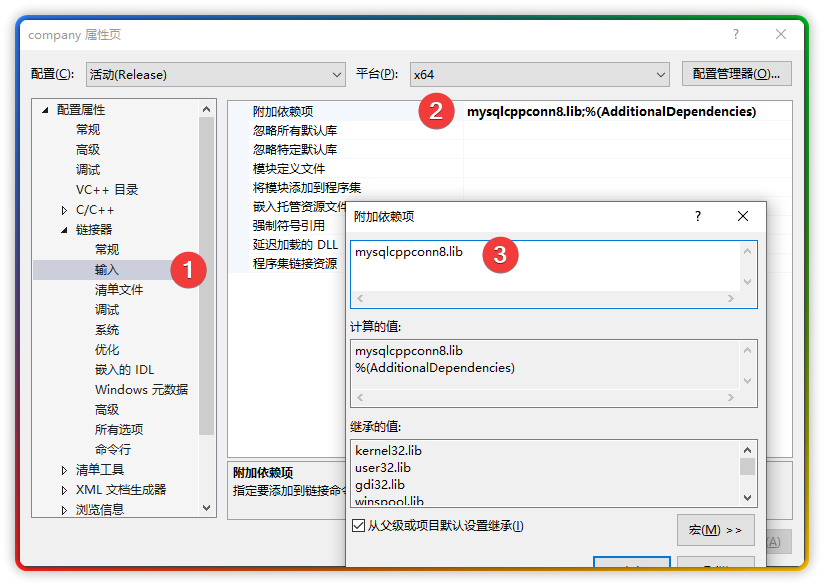
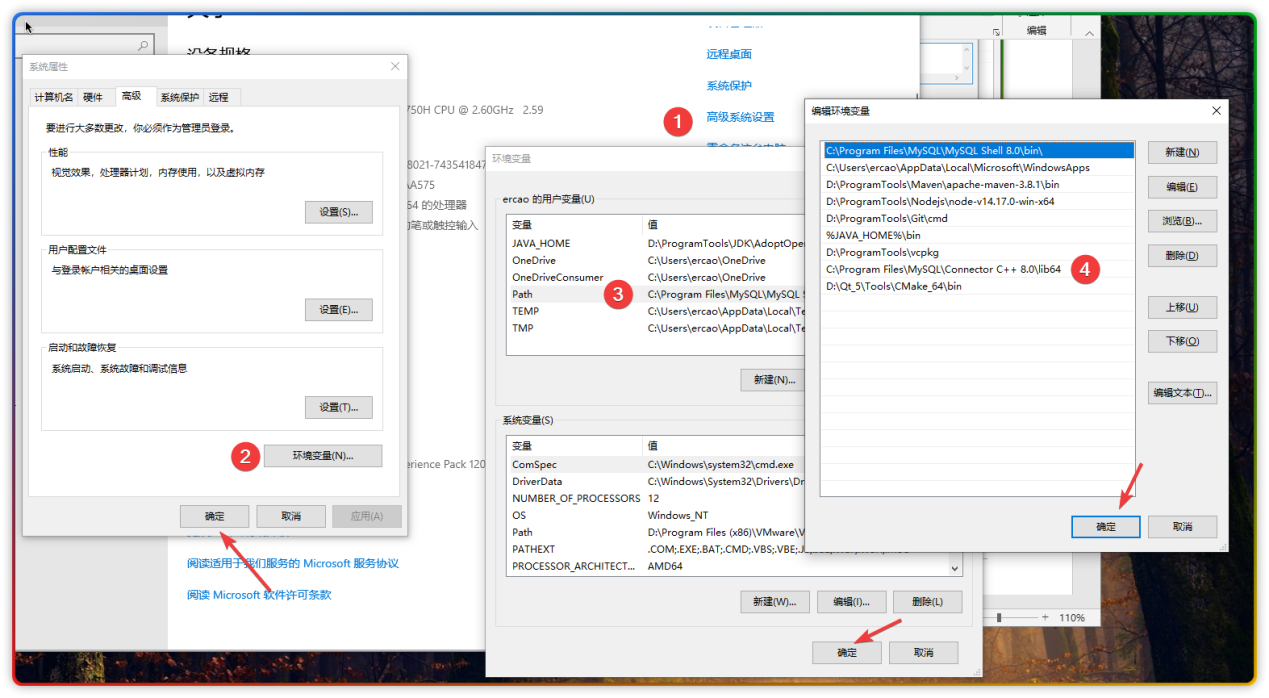


图7引入 mysqlcppconn8-2-vs14.dll

重启

1. 创建数据库和表结构，与初始化数据

导入 [company.sql](https://github.com/ercao/university/blob/main/cpp/company/company.sql)

1. 开始编码

2要设计的主要过程、类、对象

1.主要类

service 和 controller 交互使用的类：ResultEntity

分页信息：PageInfo

角色信息：Role

系统信息：System

部门类：Department

员工类：Employee

2主要结构体

管理者信息：Employee

部门信息：Department

角色信息：Role

3主要对象

Menu菜单

Employee Service

Department Service

Role Service

Employee dao

Department dao

Role dao

3程序设计时需要注意的事项

1. 所有变量定义位于函数开始处。  
   2.尽量避免全局变量，全局变量需于实现文件中定义，头文件中extern声明。  
   3.浮点零值比较采用区间值。  
   4.保证正确的前提下尽量使用移位运算。  
   5.If语句、switch语句优先处理出现概率大的情况。  
   6.指针与NULL比较反写如：NULL==p。  
   7.循环嵌套时最长的循环放内层短的放外层。  
   8.尽量不在循环体内修改循环变量。  
   9.凡是无返回值、无参数、任意类型指针参数的函数均使用void或void指针。  
   10.适当使用const，使用目的限于节省空间、提高效率、不允许修改或可能产生的函数参数修改。  
   11.多参数函数应使用结构体减少参数个数。  
   12.使用宏定义#define定义文件路径时候使用带双引号、双反斜杠的字符串。宏定义函数时候尽可能的使用括号。宏定义中的换行接续后和行起始处保证无任何字符。  
   13.一种变量名称代表一段有限空间，其内容称为值，c/c++中称之为从左值到右值的转换。另一种变量是一段有限空间的引用，同时规定空间的长度(数组)。  
   14.创建动态二维数组(指向指针的指针)一维使用new char\*[一维度] 或者 (char\*\*)malloc(一维度).二维循环于一维下动态创建new char[二维度] 或者(char\*)malloc(二维度)。释放动态二维数组刚好相反，先循环释放二维，再释放一维。  
   15.利用字符创的结束标识'\0' （其ascII 码是 0,0x00）而非空格(其ascII 码是 32，0x20)可判断字符串结束以否，并可以移动指针的方式计算字符串的大小。  
   16.free后的内存空间只是被标记为了可重新分配，但作为该进程的内存块他依然具有读写权限，为了避免内存混乱需要NULL指针避免野指针的存在。  
   17.减少程序与外存的交互次数可提速程序执行，考虑使用链表一次加载多次使用、内存块分配或者文件映射等方法。

4解决的技术难点、经常犯的错误

难点：用c++语言控制数据库、从数据库中将文件按所需格式导出

常犯错误：c++语言和MySQL编码不同导致程序不可运行

5有特点的算法

ResultEntity<int> SystemService::exportData(const string &filename) {

static const string &EXPORT\_FILE\_FAILD\_PREFIX = "导入数据失败：";

try {

// 输出到文件

ofstream ofs(filename, ios::out);

if (!ofs.is\_open()) {

throw string("文件打开失败");

}

// 导出数据

EmployeeService employee\_service;

auto employees = employee\_service.getPageInfo("1", to\_string(INT\_MAX));

DepartmentService department\_service;

auto departments = department\_service.getPageInfo("1", to\_string(INT\_MAX));

int employees\_count = 0, department\_count = 0;

// 转为json

using json = nlohmann::json;

json data;

// 保存部门信息

data["departments"] = json::array();

for (auto d : departments.data\_->data\_) {

json j;

j["id"] = d->id();

j["name"] = Util::string\_to\_utf8(d->name());

j["manager"] = Util::string\_to\_utf8(d->managerName());

j["sales"] = d->sale();

data["departments"].push\_back(j);

++department\_count;

}

// 保存官员信息

data["employees"] = json::array();

for (auto e : employees.data\_->data\_) {

json j;

// TODO

j["id"] = e->id();

j["username"] = Util::string\_to\_utf8(e->username());

j["password"] = Util::string\_to\_utf8(e->password());

j["name"] = Util::string\_to\_utf8(e->name());

j["gender"] = Util::string\_to\_utf8(e->gender());

j["age"] = e->age();

j["workedTime"] = e->workedTime();

j["salary"] = e->salary();

j["department"] = Util::string\_to\_utf8(e->departmentName());

j["role"] = Util::string\_to\_utf8(e->roleName());

j["createTime"] = e->createTime();

j["updateTime"] = e->updateTime();

data["employees"].push\_back(j);

++employees\_count;

}

ofs << data.dump();

ofs.close();

return ResultEntity<int>::successResultWidthoutData(

string("导出数据成功：").append(to\_string(department\_count))

.append(" 条部门信息记录，").append(to\_string(employees\_count)).append(" 条员工信息记录"));

} catch (const string &e) {

return ResultEntity<int>::faildResult(EXPORT\_FILE\_FAILD\_PREFIX + e);

} catch (const exception &e) {

return ResultEntity<int>::faildResult(EXPORT\_FILE\_FAILD\_PREFIX + e.what());

}

return ResultEntity<int>::faildResult("");

}

6详细代码

IMG_256***https://github.com/ercao/university***

7测试时出现过的问题及其解决方法

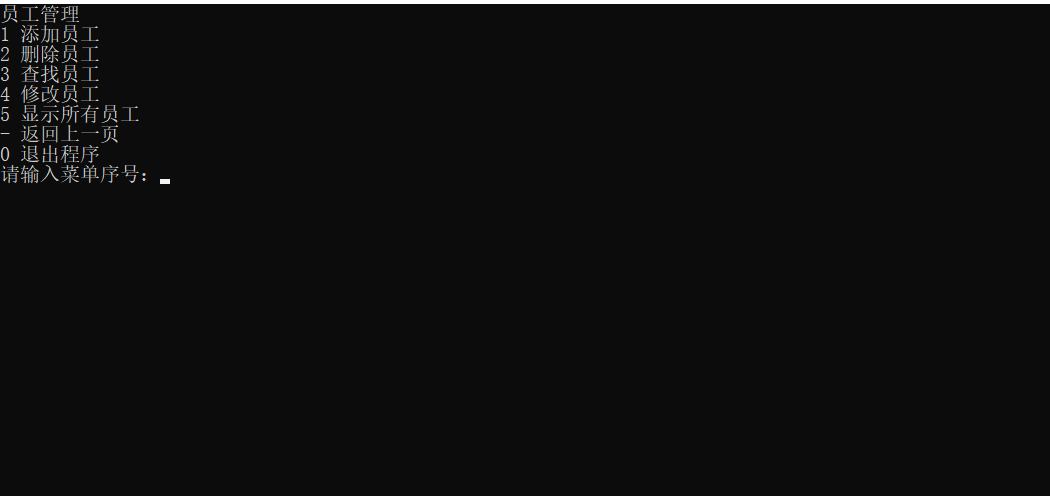
问题：员工营业额无法用c++语言按要求排序

解决方法：通过数据库对数据进行排序

6运行

图8运行菜单

图9员工管理界面

图10显示所有员工信息

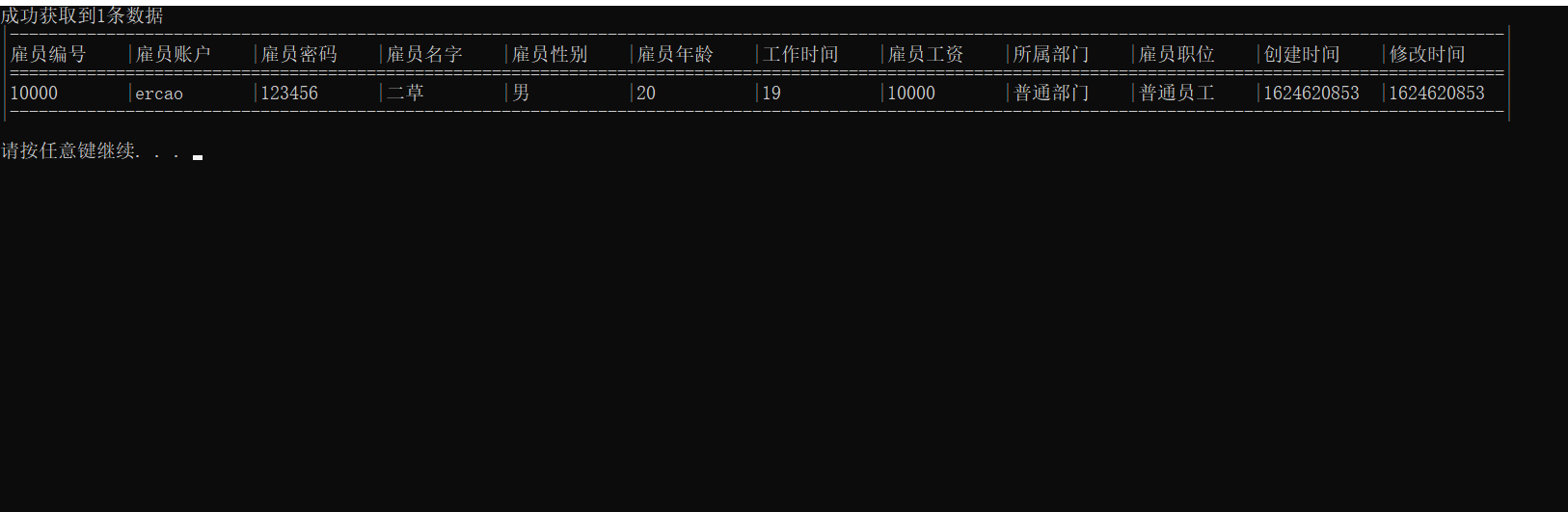
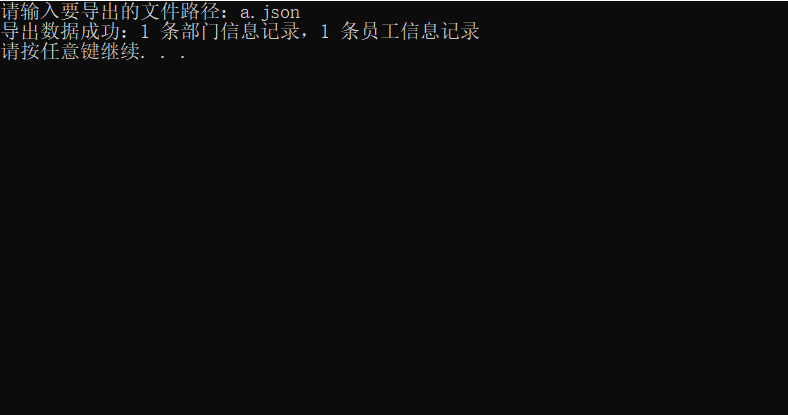


图10导出文件

图11导出的文件

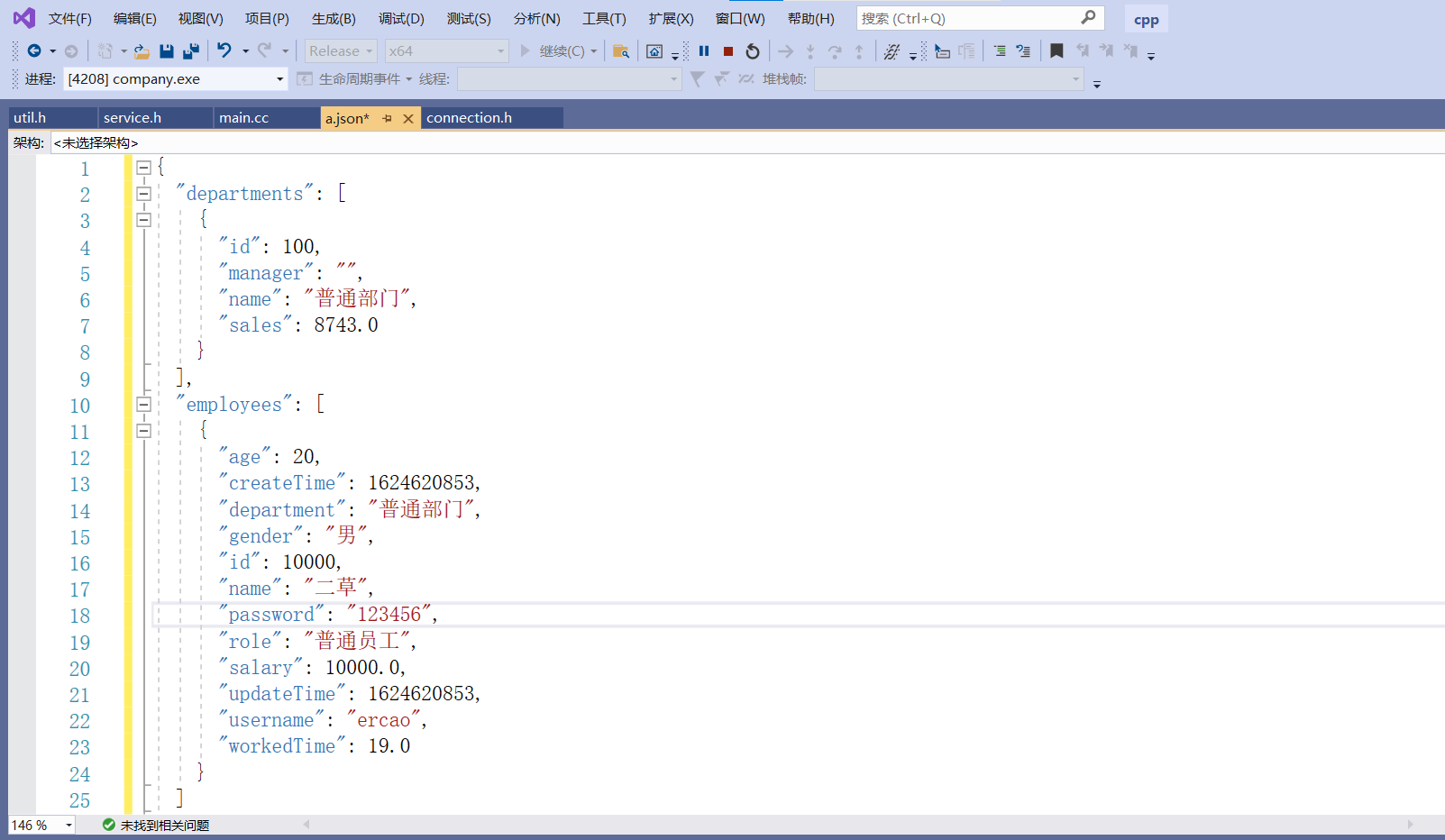
7总结

表1 完成与未完成情况

|  |  |
| --- | --- |
| 已经完成 | (1) 数据录入：输入各种数据；  (2) 数据统计：各销售经理的工资计算及最终按工资进行的冒泡排序；  (3) 数据打印：打印上述表格；  (4) 数据备份：把相关数据写入文件；  (5) 退出：退出本系统 |
| 还未完成 | 无 |

表2 小组成员分工和成绩自我评定及排序

|  |  |  |
| --- | --- | --- |
| 姓名 | 分工 | 成绩 |
| 董福江 | 数据库数据传输 | A |
| 刘立源 | 制作操作界面 | B |
| 王伟业 | 制作操作界面 | B |
| 平征 | 数据库建立 | B |

收获、经验、教训、感受：学会了使用c++语言操作数据库，对数据库有了大致的认识，掌握了新开发软件的基本操作每个编译软件都有其自身固定的逻辑和语言，想要对程序进行编译，就要先从语言开始学习，大部分的软件都有着差不多的语言模式，但也会有所不同，需要我们在学系时注意区别和记忆，很多软件都有其独到的一面，在学习时要理解它基本的运行方式，这样才能实现更加复杂的功能。而且，很多不同软件之间会存在不兼容问题，在使用时要注意，确保两个或多个软件之间能实现数据传输和交互，不会出现转码错误或语言逻辑错误。每一个被广泛使用的软件都会有深度，在短时间内无法完全学会，想要实现一个功能需要不断地研究和尝试，从平时开始积累知识和经验真的很重要，这样才不会在需要使用某些软件时没有头绪，手忙脚乱。

8参考文献