北京邮电大学软件学院

实验报告

（注意：此处填写实验指导手册中的实验名称）

**提交时，将本注释框删除**

课程名称： C语言程序设计课程设计

项目名称：

项目完成人：

姓名： 姜洪烨 学号： 2019211915

指导教师： 杨 谈

日 期： 2019 年 12 月 日

# 实验目的

（简要说明通过本实验希望达到的目的）

# 实验内容

（简要说明本实验的内容）

# 实验环境

操作系统：Windows 10 Enterprise LTSC

IDE：Visual Studio 2019 Community

# 附录

（附上实验文档，如：问题分析、设计方案、算法、设计图、程序、仿真结果、运行结果、调试心得等，具体内容根据实验要求来定。源代码请附在这里。源代码排版请特别注意，用5 号字体，行间距为单倍行距。注意节省空间，不要浪费纸张。）

## 问题分析

（说明你对这个问题的理解，包括，这个程序要解决什么问题、功能、性能、健壮性等）

### 目标

（说明这个程序要解决什么问题，达到的目标）

### 功能

（说明这个程序的各种功能，参见下例4.1.2.1和4.1.2.2。在撰写正文时，请将下面的例子删除）

#### 功能一：用鼠标画线

当用户选择此项时，

当手机接收到网络的振铃信号时，能够激活手机屏幕，并且通过手机扬声器或者耳机播放预先设定的声音；

当用户按挂机键或者接听键后，停止铃声；

若铃声持续2分钟后，用户没有按挂机键或者接听键，则向电话网络反馈信号“无人”

#### 功能二：查找联系人

当用户按“查找”按钮之后，程序显示输入框，等待用户输入。用户输入文字的同时，对用户的通信录按照字母或者汉语拼音进行检索

### 性能

（说明这个程序主要功能的性能参数，参见下例4.1.3.1。在撰写正文时，请删除例子）

#### 查找联系人的性能

在用户输入文字的同时，程序在每次输入后的0.4秒之内响应并将结果显示到屏幕上

### 健壮性

（说明这个程序应对错误输入、异常情况的能力，参见下例4.1.3.1。在撰写正文时，请删除例子）

#### 写入或读取存档时的异常处理

##### 第一层：考虑到在程序中调用了Windows API来让用户自由读取存档，因此有必要在用户选择错误的文件时报错。在本项目中通过在保存文件时添加校验位来标记该文件是本程序创建的。在读取时通过对检验位的比较来辨别用户选择的文件正确与否。在用户选择错误的文件时会弹出警告窗口来使用户重试。

##### 第二层：在写入或读取存档时会对成功写入或读取的数据块的数量与应该成功的数据块数量进行比较，在写入或读取时有数据块出错时弹出警告提醒用户重新操作。

#### 在用户键入多边形边数或坐标时的异常处理

##### 对于边数的异常处理：

###### 检查输入字符串是否全为数字，并提示错误点

作为边数，自然都必须为数字，程序将检查输入的字符串是否都为数字，再使用atoi()函数转换，否则警告"边数只能包含数字"

###### 检查边数在几何意义上的合理性，并提示错误点

从几何意义上说，多边形的边数必须大于等于3，若用户输入的边数小于3，则警告"不能构成多边形"

##### 对于坐标的异常处理

在按照 ”,” 分割输入的字符串后将检查：

###### 检查分割后的字符串是否全为数字，并提示错误点

作为坐标，自然都必须为数字，程序将检查输入的字符串是否都为数字，再使用atoi()函数转换，否则警告"坐标只能包含数字"

###### 检查坐标个数，并提示错误点

一个边数为sides的多边形只能有sides个坐标，若读取的坐标数量过多，则警告"坐标个数过多"；若读取的坐标数量过少，则警告"坐标个数过少"

###### 检查坐标范围，并提示错误点

若输入的坐标超出屏幕的显示范围，则警告"坐标范围无效"

## 设计方案

整个项目分为9个模块，对应9个C++源代码文件和9个头文件，每个模块有2-4个函数，共计21个函数。

（在此处，用文字描述整体的设计，包括，分成多少个模块，多少个文件，多少个函数，每个模块、函数、文件的大致功能； 此外，还需要包含一个程序整体的流程图，反映各个模块的执行次序和关系；但是避免绘制细节的流程）

### 模块、文件、函数的功能

### 运行简化主体流程图

## 重要算法

（在此处，用流程图描绘重要的算法，例如排序、查找等。一般的程序代码不必在此说明）

### 算法一：xxxxxx

static WORD tmp\_totalShapes;

static WORD nLines;

static WORD nCircles;

static WORD nRectangles;

static WORD nPolygons;

// count the number of each shape

if (tmp\_totalShapes != g\_nTotalShapes)

{

fileEdited = true;

nLines = 0;

nCircles = 0;

nRectangles = 0;

nPolygons = 0;

for (int i = 0; i < g\_nTotalShapes; i++)

{

switch (shapeData[i].shapeType)

{

case shape\_line:

nLines++;

break;

case shape\_circle:

nCircles++;

break;

case shape\_rectangle:

nRectangles++;

break;

case shape\_polygon:

nPolygons++;

break;

default:

break;

}

}

}

### 算法二：xxxxxx

## 实验结果

（说明实验完成情况，有需要文字回答的题目，以及实验结果截图，请写在这里）

（在此处，详细描述运行的结果。包括：不同输入对应的不同结果，异常情况的执行结果）

## 调试心得

（总结你在调试程序时的收获，对于某一类或者某几类警告、错误的处理方法。这些心得，能够对学习C语言程序设计的新手具有一些指导作用）