哈夫曼编码译码软件需求分析及概要设计

·项目背景：

哈夫曼编码（Huffman Coding）是一种广泛使用的数据压缩算法，它利用字符在文本中出现的频率差异，通过构建哈夫曼树来为不同的字符分配不等长的二进制编码，以达到压缩数据的目的。本项目旨在实现一个完整的哈夫曼编码系统，支持用户通过键盘或文件输入构建哈夫曼树、对指定文件进行编码和解码，以及将编码后的二进制数据按指定方式输出。

一、需求分析：

·问题1：能够通过键盘或者纯文本文件读入字符集的大小n，以及n个字符和权值来建立哈夫曼树，并且把建立好的哈夫曼树存入到HuffmanTree.txt中去。

分析：键盘输入采用cin,而文本文件输入考虑使用<fstream>库中的函数，建立哈夫曼树的过程参考数据结构课中所学内容，数据存储考虑使用<fstream>库中的函数。

·问题2：利用已经建立好的哈夫曼树，对文件中的正文进行编码，将结果存入到文件HuffmanCode.txt中。

分析：根据哈夫曼树，向左侧分支走，编码加一个“0”，向右侧分支走，编码加一个“1”，按照此规则，得到每个字符的编码，然后对正文进行遍历，用相应字符的对应编码替换原本正文中的字符，将结果存入目标文件。

·问题3：利用已经建立好的哈夫曼树将HuffmanCode.txt中的哈夫曼编码进行译码，结果存入到HuffmanText.txt中。

分析：同问题2，首先得到对应字符的编码，对HuffmanCode.txt中的编码进行遍历，如果为“0”，树往左侧分支走，如果为“1”，树往右侧分支走，当走到叶子节点时，即可输出该节点字符，解码成功，往复进行上述操作即可。

·问题4：能够按照垂直输出二叉树的方式，将存储在HuffmanTree.txt纯文本文件中的哈夫曼树垂直输出。并且在打印哈夫曼编码时，要求字符与编码之间是一一对应的。

分析：考虑对树的节点增加一个水平位置记号，设置根节点为0，往左一个节点-1，往右一个节点+1，对所有节点，按水平位置从小到大的顺序进行遍历输出（具体实现详见代码）。最后，注意输出时，字符与编码一一对应。

·问题5：在程序运行时，能够出现一个主选择菜单，用户能够自主选择功能：①建立哈夫曼树；②对哈夫曼树进行编码、译码等功能。

分析：考虑到用户的需要，要求设计一个可视化的主菜单界面，此处考虑引用一些图形库（来自于对Windows绘图API的封装）来实现对界面的绘制。菜单具体设计过程详见代码。

以上便是对于哈夫曼编码译码软件的需求分析。

二、概要设计

·框图：



垂直输出

文件译码

文件编码

构建哈夫曼树

主菜单界面

·模块分解设计思路：

1. 构建哈夫曼树：对于键盘输入，直接使用cin读取输入，然后依照数据结构所学知识，构建哈夫曼树（具体方法，详见代码）；对于文件读入，采取<fstream>库中的ifstream类，即可从文件中读取数据，构造树的方法同上。
2. 文件编码：通过构建的哈夫曼树，规定：往左侧走过一个节点 编码加一个“0”，往右侧则加一个“1”，当遍历的节点为叶子节点时，该节点对应字符的编码即为其哈夫曼编码，以此方式得到每个字符的编码。然后对目标文件进行遍历，直接将对应字符替换为它的编码即可，结果借助<fstream>库中的ofstream类，储存到指定文件。
3. 文件译码：遍历待处理文件，当遍历到“0”时，往数的左侧走一个节点，否则往右，当遍历到叶子节点时，即可将该节点对应的字符放在输出流中，反复此操作，直至待处理文件被遍历完。最后还是通过同样的文件操作，将输出流数据写入指定文件。
4. 垂直输出：对数的节点增设一个水平位置参数，默认根节点为0，往左侧走过一个节点-1，往右则+1，计算树中所有节点的水平位置。最后，按照水平位置从小到大的顺序，遍历树中所有节点即可。注意输出时，将字符与其编码一一对应。