Name : Ikhwan Fikri Nur Afra

ID : 2001586224

**Old Notepad Program**

**Description :**

Notepad is the most universal text editor program in every way. It can create any documents from any text file and can also read them. This program will recreate the most classic feeling of the old notepad and how does notepad looks like in DOS version. Before GUI (Graphical User Interface) operating system was created, most people always using keyboard for operating any programs on their computer especially notepad. Usually, any businessman, children even typewritter were using this program for typing word document such as letter, story and so on. Even, we can create a code in notepad. It comes from typewritter and it screen-based text editors. It’s invented in 1967. Finally, this program shows notepad program in the old ways. Literally, there still has a limited features in this program but, at least it almost contains the main features of notepad.

|  |
| --- |
| menu |
| -Input: input |
| +NewDocument(): void  +OpenDocument(): void  +EditDocument(): void  +FindWords(): void  +CopyText(): void  +ShapeGenerator(): void |

**Design/Plan:**

|  |
| --- |
| Shape |
| -Input: input  -x: int  -y: int  -i: int  -j: int |
| +Square(file:string, i:int): void  +Pyramid(file:string, i:int): void  +Rectangle(file:string, i:int): void  +Parallelogram(file:string, i:int): void  +Trapezium(file:string, i:int ):void  +InvertedTriangle(file:string, i:int): void  +Triangle(file:string, i:int): void  +Diamond(file:string, i:int): void |

|  |
| --- |
| Input |
| +Myfile: ofstream  +Mfile: ifstream  +file: string  +otherfile: string  +text: string  +line: string  +word: string  +find: string |

|  |
| --- |
| word |
| SortWords(text[]:string,count:int): void |

**Explanation Class and Function in this Project:**

1. Class input

This class contains main variables that used to every class in this program except class words. This class contains no functions.

1. Class menu

This class contains any functions that will be used in main function as a menu function. This menu function is the features in this Program.

* Function NewDocument()

This function can create text file and can replace the text in another file that has already been made

* Function OpenDocument()

This function can open the file and show the text inside the file including newline in the text file

* Function EditDocument()

This function can append the text in the end of file.

* Function FindWords()

This function can find the words in text file and can show how many words and how many words that matches from your search’s word.

* Function CopyText()

This function can copy the text from source file into another file. This can copy one line and more lines into another file.

* Function ShapeGenerator()

This function generate shape into text file even if the text still doesn’t created yet.

1. Class word

This class used inside the function FindWords(). This function only contains the process of selection sort

* Function SortWords(string text[],int count)

1. Class shape

This class is used to print the shape into text file. This class is used by functions ShapeGenerator() for printing several shapes.

* Function Square(string file,int i)

This function prints square into text file

* Function Pyramid(string file,int i)

This function prints pyramid into text file

* Function Rectangle(string file,int i,int j)

This function prints rectangle into text file

* Function Parallelogram(string file,int i)

This function prints parallelogram into text file

* Function Trapezium(string file,int i)

This function prints trapezium into text file

* Function InvertedTriangle(string file,int i)

This function prints Inverted Triangle into text file

* Function Triangle(string file,int i)

This function prints triangle into text file

* Function Diamond(string file,int i)

This function prints diamond into text file

**Everything that i have learned during the process :**

1. File Operation (Create a file, Open a File, Copy a file)
2. String class
3. Basic Matters(Loop, If-else, and iostream)
4. Sorting Algorithms
5. Nested Function
6. Class and Data structure
7. Hardwork, and Consistency
8. Thinking Positive

**Problem that i have overcome :**

1. How to find words in text file with array
2. How to print shape into text file by using two classes
3. How to type and the text file including newline

**Code**

Document.hpp

#ifndef document\_hpp

#define document\_hpp

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

class input {

public:

ofstream Myfile;

ifstream Mfile;

string file;

string otherfile;

string text;

string line;

string word;

string find;

};

#endif

Menu.hpp

#ifndef menu\_hpp

#define menu\_hpp

#include "document.hpp"

#include "shape.hpp"

using namespace std;

class menu{

input Input;

public:

void NewDocument();

void OpenDocument();

void EditDocument();

void FindWords();

void CopyText();

void ShapeGenerator();

};

class word{

public:

void SortWords(string[],int);

};

#endif

Menu.cpp

#include "menu.hpp"

#include "shape.hpp"

using namespace std;

void menu::NewDocument(){

cout << "===============New Document================" << endl;

cout << "Please Input your Filename with your format(.txt,.doc,.cpp, and so on)" << endl;

cin >> Input.file;

cin.ignore();

Input.Myfile.open(Input.file);

do{

getline(cin,Input.text);

Input.Myfile << Input.text <<endl;

}while(Input.text!="end");

cout << "file saved" << endl;

Input.Myfile.close();

}

void menu::OpenDocument(){

cout << "===============Open Document================" << endl;

cout << "Please Input your Filename that you want to open" << endl;

cin >> Input.file;

Input.Mfile.open(Input.file, ios\_base::in);

while(getline(Input.Mfile, Input.line)){

cout << Input.line << endl;

}

}

void menu::EditDocument(){

cout << "===============Edit Document================" << endl;

cout << "Please Input your Filename that you want to open" << endl;

cin >> Input.file;

Input.Myfile.open(Input.file, ios\_base::app);

cin.ignore();

do{

getline(cin,Input.text);

Input.Myfile << Input.text <<endl;

}while(Input.text!="end");

cout << "file saved" << endl;

Input.Myfile.close();

}

void menu::FindWords(){

int count=0;

int total=0;

string text[99999];

word words;

cout << "===============Find Words================" << endl;

cout << "Please Input your Filename that you want to open" << endl;

cin >> Input.file;

Input.Mfile.open(Input.file);

cout << "Please Input the words that you want to find" << endl;

cin >> Input.find;

while (Input.Mfile>>Input.word){

count++;

}

cout << "this file had " << count << " words";

Input.Mfile.close();

Input.Mfile.open(Input.file);

for (int i=0;i<count;i++){

Input.Mfile>>text[i];

}

words.SortWords(text,count);

for(int i=0;i<count;i++){

if(text[i]==Input.find){

total++;

}

}

cout << " and there are " << total << " matches" << endl;

}

void word::SortWords(string text[],int count){

int startScan, minIndex;

string minValue;

for (startScan = 0; startScan < (count-1); startScan++)

{

minIndex = startScan;

minValue = text[startScan];

for(int index = startScan + 1; index < count; index++)

{

if (text[index] < minValue)

{

minValue = text[index];

minIndex = index;

}

}

text[minIndex] = text[startScan];

text[startScan] = minValue;

}

}

void menu::CopyText(){

string choice,choice2,line,lines3;

int lines,lines2;

int count=0;

cout << "===============Copy Text================" << endl;

cout << "Please Input your filename that you want to open" << endl;

cin >> Input.file;

Input.Mfile.open(Input.file);

cout << "Please Input the target file" << endl;

cin >> Input.otherfile;

cout << "do you want to copy the text or replace the text? (copy/replace)";

cin >> choice;

if (choice=="copy"){

Input.Myfile.open(Input.otherfile, ios::app);

cout << "which parts do you want to copy? (all/selected)";

}else if(choice=="replace"){

Input.Myfile.open(Input.otherfile, ios::ate);

cout << "which parts do you want to replace? (all/selected)";

}

cin >> choice2;

if (choice2=="all"){

Input.Myfile << Input.Mfile.rdbuf();

} else if (choice2=="selected") {

cout << "what number of line do you want to copy?";

cin >> lines;

cout << "do you only want to copy one line or more? (one/more)";

cin >> lines3;

if (lines3=="one"){

while(getline(Input.Mfile,line)){

if(count==lines){

Input.Myfile << line << endl;

}

count++;

}

}else if (lines3=="more"){

cout << "from line " << lines << " till where?";

cin >> lines2;

while(getline(Input.Mfile,line)){

if(count>lines-1 && count<lines2+1){

Input.Myfile << line << endl;

}

count++;

}

}

}

cout << "Copy Text Completed" << endl;

Input.Myfile.close();

Input.Mfile.close();

}

void menu::ShapeGenerator(){

int choice,size,width,length;

shape shape;

string file;

cout << "===============Shape Generator================" << endl;

cout << "Please Input your Filename with your format(.txt,.doc,.cpp, and so on)" << endl;

cin >> file;

cin.ignore();

cout << "Which Shape do you want to draw?" << endl

<< "1.Rectangle" << endl

<< "2.Pyramid" << endl

<< "3.Triangle" << endl

<< "4.Inverted Triangle" << endl

<< "5.Square" << endl

<< "6.Trapezium" << endl

<< "7.Diamond" << endl

<< "8.Parallelogram" << endl;

cin >> choice;

if(choice==1){

cout << "How long rectangle's width you want to draw?";

cin >> width;

cout << "How long rectangle's length you want to draw?";

cin >> length;

shape.Rectangle(file,width,length);

}

cout << "How much size do you want to print?" << endl;

cin >> size;

switch (choice){

case 2: shape.Pyramid(file,size);break;

case 3: shape.Triangle(file,size);break;

case 4: shape.InvertedTriangle(file,size);break;

case 5: shape.Square(file,size);break;

case 6: shape.Trapezium(file,size);break;

case 7: shape.Diamond(file,size);break;

case 8: shape.Parallelogram(file,size);break;

default:

cout << "invalid data" << endl;break;

}

cout << "Print Shape Completed !" << endl;

}

shape.hpp

#ifndef shape\_hpp

#define shape\_hpp

#include "document.hpp"

#include "menu.hpp"

using namespace std;

class shape{

input Input;

int x;

int y;

int i;

int j;

public:

void Square(string file,int i){

Input.Myfile.open(file, ios::app);

for (int y=1;y<=i;y++){

for(int x=1;x<=i;x++){

cout << "\*";

Input.Myfile << "\*";

}

cout << "\n";

Input.Myfile << "\n";

}

}

void Pyramid(string file,int i){

Input.Myfile.open(file, ios::app);

for (y=1;y<=i;y++){

for(x=1;x<=y-1;x++){

cout << " ";

Input.Myfile << " ";

}

for(x=y\*2-1;x<=i\*2-1;x++){

cout << "\*";

Input.Myfile << "\*";

}

cout << "\n";

Input.Myfile << "\n";

}

}

void Rectangle(string file,int i,int j){

Input.Myfile.open(file, ios::app);

cout << "How long rectangle do you want?" << endl;

for (int y=1;y<=i;y++){

for(int x=1;x<=j;x++){

cout << "\*";

Input.Myfile << "\*";

}

cout << "\n";

Input.Myfile << "\n";

}

}

void Parallelogram(string file,int i){

Input.Myfile.open(file, ios::app);

for (y=1;y<=i;y++){

for(x=1;x+y<=i;x++){

Input.Myfile << " ";

}

for(x=1;x<=i;x++){

Input.Myfile << "@";

}

Input.Myfile << "\n";

}

}

void Trapezium(string file,int i){

Input.Myfile.open(file, ios::app);

for (y=1;y<=i;y++){

for(x=1;x+y<=i;x++){

Input.Myfile << " ";

}

for(x=1;x<=y\*2+6;x++){

Input.Myfile << "\*";

}

Input.Myfile << "\n";

}

}

void InvertedTriangle(string file,int i){

Input.Myfile.open(file, ios::app);

for (y=0;y<=i-1;y++){

for(x=y;x<i;x++){

Input.Myfile << "\*";

}

Input.Myfile << "\n";

}

}

void Triangle(string file,int i){

Input.Myfile.open(file, ios::app);

for (int y=1;y<=i;y++){

for(int x=0;x<y;x++){

Input.Myfile << "\*";

}

Input.Myfile << "\n";

}

}

void Diamond(string file,int i){

Input.Myfile.open(file, ios::app);

for (y=1;y<=i;y++){

for(x=1;x+y<=i;x++){

Input.Myfile << " ";

}

for(x=1;x<=y\*2-1;x++){

Input.Myfile << "\*";

}

Input.Myfile << endl;

}

for (y=1;y<=i;y++){

for(x=1;x<=y;x++){

Input.Myfile << " ";

}

for(x=y\*2-1;x<=(i-1)\*2-1;x++){

Input.Myfile << "\*";

}

Input.Myfile << endl;

}

}

};

#endif

main.cpp

#include <iostream>

#include "menu.hpp"

using namespace std;

int main(){

menu menus;

int choice;

char again;

do{

cout << "============================================" << endl

<< "| OLD WORDPAD PROGRAM |" << endl

<< "| by : ikhwan fikri nur afra |" << endl

<< "| |" << endl

<< "| Menu |" << endl

<< "| 1. New Document |" << endl

<< "| 2. Open Document |" << endl

<< "| 3. Edit Document |" << endl

<< "| 4. Find Words |" << endl

<< "| 5. Copy Text |" << endl

<< "| 6. Shape Generator |" << endl

<< "| |" << endl

<< "|please input the number(1-7) |" << endl

<< "============================================" << endl;

cin >> choice;

switch (choice) {

case 1: menus.NewDocument();break;

case 2: menus.OpenDocument();break;

case 3: menus.EditDocument();break;

case 4: menus.FindWords();break;

case 5: menus.CopyText();break;

case 6: menus.ShapeGenerator();break;

default:

cout <<"Invalid Choice"<<endl;break;

}

cout << "Do you want to go back to main menu (y/n) ? \a";

cin >> again;

cout << endl;

} while(again!='n');

return 0;

}