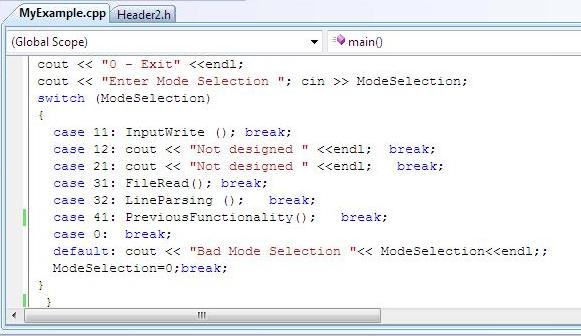
**Лабораторна робота №12 Розгляд проблемних питань**

1. Завдання впорядкувати функціональність програми шляхом використання оператора вибору, визначивши групи та коди операцій передбачає не тільки виведення меню та визначення коду обрання позиції цього меню, але і реструктурування програми. В ЛР№12 наведено приклад меню. Реалізацію цього меню доцільно робити шляхом використання оператору вибору **case**, локалізуючи функціональність в функціях (див. скреншот нижче, розглядаючи його як приклад). В подальшому при реалізації функціональності похідних класів навантаження переноситься на функції, які можуть підлягти подальшій декомпозиції. Такий підхід відповідає вимогам структурного програмування, оскільки модулі будуть достатньо компактними і прозорими.



1. Для розбору виведеного до файлу рядка за складовими пропонується використовувати не запропонований в лекції "ООП Тема 13 Поліморфізм та віртуальні функції" клас String, а методику, яка полягає в тому, що визначається початкова позиція атрибуту, який відповідає значенню члену класу (це 0 – для першого атрибуту та позиція, наступна за знайденим розділювачем ";"), а потім знаходиться найближчий розділювач ";". Відповідно, значення атрибуту між цими двома позиціями. Використовувати для розбору рядка клас String недоцільно, оскільки там використовуються символьні змінні, для опрацьовування яких потрібно створювати більш громіздкий текст ніж при використанні рядкових змінних (string). Нижче приводиться приклад розбору рядка з використанням рядкових змінних. Якщо виникли проблеми з виконанням п.2 ЛП№12, можете скористатися ідеями, наведеними в тексті нижче.

**#include <iostream>**

**#include <string>**

**#include <Windows.h>**

**using namespace std;**

**int main()**

**{ system("color F0");**

**std::string MyString("New;0;2005;+380442233445;1;#");**

**std::string Delimiter(";");**

**std::string CurrentAttribute;**

**int StrungSize;**

**int DelimiterPos=0;**

**int BegPos=0;**

**int i=1;**

**StrungSize=MyString.length();**

**std::cout << "str = " << MyString <<"("<<StrungSize<<")\n";**

**while (DelimiterPos < StrungSize-2 )**

**{DelimiterPos=MyString.find(Delimiter,BegPos);**

**CurrentAttribute=MyString.substr(BegPos,DelimiterPos-BegPos);**

**std::cout<< i << " substr = " << CurrentAttribute <<'\n';**

**i++;**

**BegPos=DelimiterPos+1;**

**}**

**system("pause");**

**return 0;**

**}**