На основі класу «список» реалізувати програму для ведення особистого бюджету витрат.

А саме передбачити такі можливості:

1. Додавання витрат
2. Видалення/редагування витрат
3. Пошук по дням
4. Зберігання витрат в файл
5. Зчитування з файлу

Вивести звітність:

1. Витрати помісячно
2. Витрати по категоріям (період запитувати в користувача).

Програма має бути написана в стилі ООП.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<Windows.h>

#include <iostream>

#include<conio.h>

using namespace std;

struct Date

{

int number;

int month;

int year;

bool operator==(const Date& o)

{

return number == o.number && month == o.month && year == o.year;

}

};

struct Costs

{

Date date;

char\* category;

char\* name;

float money;

Costs\* next;

Costs\* prev;

};

class List

{

private:

Costs\* head;

Costs\* tail;

int count;

void Add\_To\_Head(Date date\_, const char\* category\_, const char\* name\_, float money\_);

void Delete\_From\_Head();

void Delete\_From\_Tail();

public:

void Add\_To\_Tail(Date date\_, const char\* category\_, const char\* name\_, float money\_);

List();

List(const List& l);

~List();

void Delete\_All();

void Delete\_Cost(Date date\_, const char\* category\_, const char\* name\_);

void Redact(Date date\_, const char\* category\_, const char\* name\_);

void Look\_For\_By\_Day(Date date\_);

void Save\_To\_File(const char\* p);

void Read\_From\_File(const char\* p);

void Print\_All();

void Print\_By\_Month(int month, int year);

void Print\_By\_Category(const char\* category);

};

List::List()

{

head = tail = NULL;

count = 0;

}

List::List(const List& l)

{

this->head = this->tail = NULL;

Costs\* tmp = l.head;

this->count = l.count;

while (tmp != NULL)

{

this->Add\_To\_Tail(tmp->date, tmp->category, tmp->name, tmp->money);

tmp = tmp->next;

}

}

List::~List()

{

Delete\_All();

}

void List::Delete\_From\_Head()

{

Costs\* tmp = head;

head = head->next;

/\*head->next->prev = NULL;\*/

delete[] tmp->category;

delete[] tmp->name;

delete tmp;

count--;

}

void List::Delete\_From\_Tail()

{

Costs\* tmp = tail;

tail->prev->next = NULL;

delete[] tmp->category;

delete[] tmp->name;

delete tmp;

count--;

}

void List::Delete\_All()

{

while (head != NULL)

Delete\_From\_Head();

}

void List::Add\_To\_Head(Date date\_, const char\* category\_, const char\* name\_, float money\_)

{

Costs\* tmp = new Costs;

tmp->prev = NULL;

tmp->date.number = date\_.number;

tmp->date.month = date\_.month;

tmp->date.year = date\_.year;

tmp->category = new char[strlen(category\_) + 1];

tmp->name = new char[strlen(name\_) + 1];

strcpy(tmp->category, category\_);

strcpy(tmp->name, name\_);

tmp->money = money\_;

tmp->next = head;

if (head != NULL)

{

head->prev = tmp;

head = tmp;

}

else

{

head = tail = tmp;

}

count++;

}

void List::Add\_To\_Tail(Date date\_, const char\* category\_, const char\* name\_, float money\_)

{

Costs\* tmp = new Costs;

tmp->next = NULL;

tmp->date.number = date\_.number;

tmp->date.month = date\_.month;

tmp->date.year = date\_.year;

tmp->category = new char[strlen(category\_) + 1];

tmp->name = new char[strlen(name\_) + 1];

strcpy(tmp->category, category\_);

strcpy(tmp->name, name\_);

tmp->money = money\_;

tmp->prev = tail;

if (head != NULL)

{

tail->next = tmp;

tail = tmp;

}

else

{

head = tail = tmp;

}

count++;

}

void List::Delete\_Cost(Date date\_, const char\* category\_, const char\* name\_)

{

int k = 1;

bool b = false;

Costs\* del = head;

while (del != NULL)

{

if (del->date == date\_ && strcmp(del->category, category\_) == 0 && strcmp(del->name, name\_) == 0)

{

b = true;

break;

}

del = del->next;

k++;

}

if (b)

{

if (k == 1)

Delete\_From\_Head();

else if (k == count)

Delete\_From\_Tail();

else

{

del->prev->next = del->next;

del->next->prev = del->prev;

delete[] del->category;

delete[] del->name;

delete del;

}

count--;

}

else

cout << "Не знайдено!\n";

}

void List::Redact(Date date\_, const char\* category\_, const char\* name\_)

{

int a;

bool b = false;

Costs\* tmp = head;

while (tmp != NULL)

{

if (tmp->date == date\_ && strcmp(tmp->category, category\_) == 0 && strcmp(tmp->name, name\_) == 0)

{

b = true;

break;

}

tmp = tmp->next;

}

if (b)

{

cout << "Що хочете змінити?(число - 1, місяць - 2, рік - 3, ціна - 4, вихід - 0)->";

cin >> a;

while (a != 0)

{

switch (a)

{

case 1:

cout << "Введіть нове число ->";

cin >> tmp->date.number;

break;

case 2:

cout << "Введіть новий місяць ->";

cin >> tmp->date.month;

break;

case 3:

cout << "Введіть новий рік ->";

cin >> tmp->date.year;

break;

case 4:

cout << "Введіть нову ціну ->";

cin >> tmp->money;

break;

}

cout << "Продовжити змінювати?(число - 1, місяць - 2, рік - 3, ціна - 4, вихід - 0)->";

cin >> a;

}

}

else

cout << "Не знайдено!\n";

}

void List::Look\_For\_By\_Day(Date date\_)

{

bool b = false;

Costs\* tmp = head;

while (tmp != NULL)

{

if (tmp->date == date\_)

{

b = true;

cout << "-------------------------------------\n";

cout << tmp->category << endl;

cout << tmp->name << endl;

cout << tmp->money << "$" << endl;

cout << "-------------------------------------\n";

}

tmp = tmp->next;

}

if (!b)

cout << "Не знайдено!\n";

}

void List::Print\_By\_Month(int month, int year)

{

float suma = 0;

Costs\* tmp = head;

while (tmp != NULL)

{

if (tmp->date.month == month && tmp->date.year == year)

{

cout << "-------------------------------------\n";

cout << tmp->date.number << "." << tmp->date.month << "." << tmp->date.year << endl;

cout << tmp->category << endl;

cout << tmp->name << endl;

cout << tmp->money << "$" << endl;

cout << "-------------------------------------\n";

suma += tmp->money;

}

tmp = tmp->next;

}

if (suma != 0)

cout << "Витрачено за місяць - " << suma << "$\n";

else

cout << "Не знайдено!\n";

}

void List::Print\_All()

{

Costs\* tmp = head;

while (tmp != NULL)

{

cout << "-------------------------------------\n";

cout << tmp->date.number << "." << tmp->date.month << "." << tmp->date.year << endl;

cout << tmp->category << endl;

cout << tmp->name << endl;

cout << tmp->money << "$" << endl;

cout << "-------------------------------------\n";

tmp = tmp->next;

}

}

void List::Print\_By\_Category(const char\* category)

{

Costs\* tmp;

float suma;

int a,number, month, year;

cout << "Хочете побачити витрати за конкретне число - 1, конкретний місяць - 2, конкретний рік - 3, вихід - 0\n";

cout << "->";

cin >> a;

while (a != 0)

{

switch (a)

{

case 1:  
 cout << "Введіть число->";  
 cin >> number;  
 cout << "Введіть місяць->";  
 cin >> month;  
 cout << "Введіть рік->";  
 cin >> year;  
 tmp = head;  
 suma = 0;  
 while (tmp != NULL)  
 {  
 if (strcmp(category, tmp->category) == 0 && tmp->date.number == number && tmp->date.month == month && tmp->date.year == year)  
 {  
 cout << "-------------------------------------\n";  
 cout << tmp->name << endl;

cout << tmp->money << "$" << endl;

cout << "-------------------------------------\n";  
 suma += tmp->money;  
 }  
 tmp = tmp->next;  
 }  
 if (suma != 0)

cout << "Витрачено за місяць - " << suma << "$\n";

else

cout << "Не знайдено!\n";  
  
 break;

case 2:

cout << "Введіть місяць->";  
 cin >> month;  
 cout << "Введіть рік->";  
 cin >> year;;  
 tmp = head;  
 suma = 0;

while (tmp != NULL)  
 {  
 if (strcmp(category, tmp->category) == 0 && tmp->date.month == month && tmp->date.year == year)  
 {  
 cout << "-------------------------------------\n";  
 cout << tmp->name << endl;

cout << tmp->money << "$" << endl;

cout << "-------------------------------------\n";  
 suma += tmp->money;  
 }  
 tmp = tmp->next;  
 }  
 if (suma != 0)

cout << "Витрачено за місяць - " << suma << "$\n";

else

cout << "Не знайдено!\n";  
 break;

case 3:

cout << "Введіть рік->";  
 cin >> year;  
 tmp = head;  
 suma = 0;

while (tmp != NULL)  
 {  
 if (strcmp(category, tmp->category) == 0 && tmp->date.year == year)  
 {  
 cout << "-------------------------------------\n";  
 cout << tmp->name << endl;

cout << tmp->money << "$" << endl;

cout << "-------------------------------------\n";  
 suma += tmp->money;  
 }  
 tmp = tmp->next;  
 }  
 if (suma != 0)

cout << "Витрачено за місяць - " << suma << "$\n";

else

cout << "Не знайдено!\n";  
 break;

}

cout << "Хочете побачити витрати за конкретне число - 1, конкретний місяць - 2, конкретний рік - 3, вихід - 0\n";

cout << "->";

cin >> a;

}

}

//шлях вказувати повний!Наприклад E:\\TEST\\aa.txt

void List::Save\_To\_File(const char\* p)

{

Costs\* tmp = head;

FILE\* f = fopen(p, "wb");

while (tmp != NULL)

{

fwrite(&tmp->date, sizeof(Date), 1, f);

int size = strlen(tmp->category) + 1;

fwrite(&size, sizeof(int), 1, f);

fwrite(tmp->category, sizeof(char), strlen(tmp->category) + 1, f);

size = strlen(tmp->name) + 1;

fwrite(&size, sizeof(int), 1, f);

fwrite(tmp->name, sizeof(char), strlen(tmp->name) + 1, f);

fwrite(&tmp->money, sizeof(float), 1, f);

tmp = tmp->next;

}

fclose(f);

}

//шлях вказувати повний!Наприклад E:\\TEST\\aa.txt

void List::Read\_From\_File(const char\* p)

{

FILE\* f = fopen(p, "rb");

int size;

while (!feof(f))

{

Costs\* tmp = new Costs;

int readed\_bytes = fread(&tmp->date, sizeof(Date), 1, f);

if (readed\_bytes == 0)

break;

fread(&size, sizeof(int), 1, f);

tmp->category = new char[size + 1];

fread(tmp->category, sizeof(char), size, f);

fread(&size, sizeof(int), 1, f);

tmp->name = new char[size + 1];

fread(tmp->name, sizeof(char), size, f);

fread(&tmp->money, sizeof(float), 1, f);

Add\_To\_Tail(tmp->date, tmp->category, tmp->name, tmp->money);

delete tmp;

}

fclose(f);

}

char\* String\_for\_anything()

{

char\* tmp = NULL;

int k = 0;

do

{

char s;

s = \_getch();

cout << s;

if (s == 13)

break;

tmp = (char\*)realloc(tmp, sizeof(char) \* (k + 1));

tmp[k] = s;

k++;

} while (true);

tmp = (char\*)realloc(tmp, sizeof(char) \* (k + 1));

tmp[k] = '\0';

cout << endl;

return tmp;

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int a;

List lst;

Date b;

char\* category=NULL;

char\* name=NULL;

float money;

cout << "Ведення особистого бюджету витрат\n";

cout << "Додати витрату - 1, видалити витрату - 2, редагувати витрату - 3, пошук по дням - 4, вивести витрати за місяць - 5, вивести витрати по категоріям - 6, вивести всі витрати - 7, зчитати з файлу витрати - 8, зберегти витрати в файл - 9, вихід - 0\n";

cin >> a;

while (a != 0)

{

switch (a)

{

case 1:  
 if (category != NULL)  
 delete[] category;  
 if (name != NULL)  
 delete[] name;  
 cout << "Введіть число ->";  
 cin >> b.number;  
 cout << "Введіть місяць ->";  
 cin >> b.month;  
 cout << "Введіть рік ->";  
 cin >> b.year;  
 cout << "Введіть категорію:\n";  
 category = String\_for\_anything();  
 cout << "Введіть назву:\n";  
 name= String\_for\_anything();  
 cout << "Введіть суму витрати->";  
 cin >> money;  
 lst.Add\_To\_Tail(b,category,name, money);  
 break;

case 2:

if (category != NULL)  
 delete[] category;  
 if (name != NULL)  
 delete[] name;  
 cout << "Введіть число ->";  
 cin >> b.number;  
 cout << "Введіть місяць ->";  
 cin >> b.month;  
 cout << "Введіть рік ->";  
 cin >> b.year;  
 cout << "Введіть категорію:\n";  
 category = String\_for\_anything();  
 cout << "Введіть назву:\n";  
 name = String\_for\_anything();

lst.Delete\_Cost(b, category, name);

break;

case 3:

if (category != NULL)  
 delete[] category;  
 if (name != NULL)  
 delete[] name;  
 cout << "Введіть число ->";  
 cin >> b.number;  
 cout << "Введіть місяць ->";  
 cin >> b.month;  
 cout << "Введіть рік ->";  
 cin >> b.year;  
 cout << "Введіть категорію:\n";  
 category = String\_for\_anything();  
 cout << "Введіть назву:\n";  
 name = String\_for\_anything();

lst.Redact(b, category, name);

break;

case 4:

cout << "Введіть число ->";  
 cin >> b.number;  
 cout << "Введіть місяць ->";  
 cin >> b.month;  
 cout << "Введіть рік ->";  
 cin >> b.year;

lst.Look\_For\_By\_Day(b);

break;

case 5:

cout << "Введіть місяць ->";  
 cin >> b.month;  
 cout << "Введіть рік ->";  
 cin >> b.year;

lst.Print\_By\_Month(b.month, b.year);

break;

case 6:

if (category != NULL)  
 delete[] category;

cout << "Введіть категорію:\n";  
 category = String\_for\_anything();

lst.Print\_By\_Category(category);

break;

case 7:

lst.Print\_All();

break;

case 8:

if (category != NULL)  
 delete[] category;

cout << "Введіть шлях файлу:\n";

category = String\_for\_anything();

lst.Read\_From\_File(category);

break;

case 9:

if (category != NULL)  
 delete[] category;

cout << "Введіть шлях файлу:\n";

category = String\_for\_anything();

lst.Save\_To\_File(category);

break;

}

cout << "Додати витрату - 1, видалити витрату - 2, редагувати витрату - 3, пошук по дням - 4, вивести витрати за місяць - 5, вивести витрати по категоріям - 6, вивести всі витрати - 7, зчитати з файлу витрати - 8, зберегти витрати в файл - 9, вихід - 0\n";

cin >> a;

}

if (category != NULL)  
 delete[] category;  
 if (name != NULL)  
 delete[] name;

return 0;

}