Задача про банкомат «для своїх»

//-------------------------------- Note.h -------------

#pragma once

#include <string>

using namespace std;

class Note

{

//1.Описуємо поля

int noteValue\_; //Номінал купюри

int notesCount\_; //Кількість купюр

//Для перевірки чи є допустимим введений номінал

const int ACCIBBLE\_NOTES\_COUNT = 5; //Кількість допустимих номіналів

const int NOTES\_VALUES[]{10,50,100,200,500}; //Масив допустимих номіналів

bool isNoteAccessible(int value); //Визначення того, чи є вказане значення допустимим номіналом

public:

//2.Методи доступу

int get\_noteValue();

void set\_noteValue(int value);

int get\_noteCount();

void set\_noteCount(int value);

//3.Конструктри

Note(int noteValue, int count);

Note(int noteValue);

Note(const Note & obj);

Note();

//4.Інші методи

void input();

int getTotalSum();

string toString();

};

//--------------------------- Note.cpp --------------------------

#include "stdafx.h"

#include "Note.h"

#include <iostream>

using namespace std;

bool Note::isNoteAccessible(int value)

{

for (int i = 0; i < ACCIBBLE\_NOTES\_COUNT; i++)

{

if (value == NOTES\_VALUES[i])

return true;

}

return false;

}

int Note::get\_noteValue()

{

return noteValue\_;

}

void Note::set\_noteValue(int value)

{

if (isNoteAccessible( value))

{

noteValue\_ = value;

}

else

{

throw "err";

}

}

int Note::get\_noteCount()

{

return notesCount\_;

}

void Note::set\_noteCount(int value)

{

if (value>=0)

{

notesCount\_ = value;

}

else

{

throw "err";

}

}

Note::Note(int noteValue, int count)

{

set\_noteCount(count);

set\_noteValue(noteValue);

}

Note::Note(int noteValue):Note(noteValue,0)

{

}

Note::Note(const Note & obj):Note(obj.noteValue\_,obj.notesCount\_)

{

}

Note::Note():Note(NOTES\_VALUES[0],0)

{

}

void Note::input()

{

printf("Note Value =");

int value;

cin >> value;

set\_noteValue(value);

printf("Notes Count=");

int value;

cin >> value;

set\_noteCount(value);

}

int Note::getTotalSum()

{

return notesCount\_\*noteValue\_;

}

string Note::toString()

{

return to\_string(noteValue\_)+"-"+to\_string(notesCount\_);

}

//----------------------------------- Bankomat.h ------------------------------

#pragma once

#include "Note.h"

class Bankomat

{

int notesTypeCount\_; //кількість видів банкнот

Note\* \* notes; //масив банкнот

const int CLIENTS\_COUNT = 6; //Кількість своїх клієнтів

const int CLIENTS\_ID[]{ 233,123,678,235,127,679 }; //ідентифікатори своїх клієнтів

public:

int get\_notesTypeCount();

void set\_notesTypeCount(int value);

Bankomat(int banknotsTypeCount, int firstBanknotValue,...); //Bankomat(5, 10,7, 50,67, 100,230, 200,90 )

Bankomat(int banknotsTypeCount, Note\*\* notes);

~Bankomat();

};

//-------------------------------------- Bankomat.cpp -----------------------

#include "stdafx.h"

#include "Bankomat.h"

int Bankomat::get\_notesTypeCount()

{

return notesTypeCount\_;

}

void Bankomat::set\_notesTypeCount(int value)

{

if (value>0)

{

notesTypeCount\_ = value;

notes = new Note\*[notesTypeCount\_]; //Створюємо масив покажчиків на купюри

}

else

{

throw "err";

}

}

Bankomat::Bankomat(int banknotsTypeCount, int firstBanknotValue, ...)

{

set\_notesTypeCount( banknotsTypeCount);

int \* data = &firstBanknotValue;

for (int i = 0; i < banknotsTypeCount; i++) //Створюємо банкноти і записуємо адреси у масив

{

notes[i] = new Note(data[2\*i], data[2 \* i+1]);

}

}

Bankomat::Bankomat(int banknotsTypeCount, Note \*\* notes)

{

set\_notesTypeCount(banknotsTypeCount);

for (size\_t i = 0; i < banknotsTypeCount; i++)

{

this->notes[i] = new Note(\*(notes[i]));

}

}

Bankomat::~Bankomat()

{

for (int i = 0; i < notesTypeCount\_; i++) //Видаляємо банкноти, адреси яких є у масиві

{

delete notes[i];

}

delete[] notes; //Видаляємо масив покажчиків

}

//-------------------------------------------- main --------------------------

// ConsoleApplication41.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include "Bankomat.h"

int main()

{

Note\*\* notes = new Note\*[3];

notes[0] = new Note(50,27);

notes[1] = new Note(100, 89);

notes[2] = new Note(200, 90);

Bankomat\* bankomat = new Bankomat(3, notes);

//-------------------------------------------------------

Bankomat\* bankomat = new Bankomat(3, 50,27,100,89,200,90);

return 0;

}

//========================================================

Допомога для Triad. Спочатку потрібно розробити клас Number , який представляє число з нижньою і верхньою межею.

#pragma once

class Number

{

//1.Поля

int value\_;

int minValue\_;

int maxValue\_;

void set\_Maxvalue(int value);

void set\_Minvalue(int value);

public:

//2.Методи доступу

void set\_value(int value);

int get\_value();

int get\_Maxvalue();

int get\_Minvalue();

//3. Конструктори

Number(int value, int minValue, int maxValue);

Number(int value, int maxValue);

Number(int value);

Number();

//4.Інші методи

bool increase(); //збільшення на 1. Повертає true, якщо не вийшли за межі і false в іншому випадку

bool decrease(); //збільшення на 1. Повертає true, якщо не вийшли за межі і false в іншому випадку

};