Main

#include "Pharmacy.h"

#include "Recipe.h"

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

cout << "\t\t\t\t\t\tЛабораторна робота № 2" << endl;

unsigned choice;

bool stopWork = false;

Recipe recipe;

Pharmacy pharmacy;

while (stopWork == false)

{

cout << "1. Вихід" << endl;

cout << "2. Переглянути рецепт" << endl;

cout << "3. Редагувати рецепт" << endl;

cout << "4. Перевірити наявність препарату" << endl;

cin >> choice;

cout << endl;

switch (choice)

{

case 1: stopWork = true; break;

case 2: recipe.ShowInformation(); break;

case 3: recipe.EditRecipe(); break;

case 4: pharmacy.SearchMedicine(); break;

default: cout << "Помилка при виборі пункту меню." << endl; break;

}

}

system("pause");

return 0;

}

Recipe.cpp

#include "Recipe.h"

Recipe::Recipe()

{

list1.push\_back("Парацетамол");

list1.push\_back("Новірин");

list1.push\_back("Анальгін");

list1.push\_back("Стрепсілс");

list1.push\_back("Комбігрип");

}

void Recipe::ShowInformation()

{

std::cout << "Список потрібних ліків: " << std::endl;

int i = 1;

for (std::string medicines : list1)

{

std::cout << i <<" " << medicines << std::endl;

i++;

}

std::cout << std::endl;

}

std::vector<std::string> Recipe::changeRecipe()

{

std::cout << "Ви хочете редагувати весь рецепт, чи лиш частину?" << std::endl;

ShowInformation();

unsigned choice;

bool stopWork = false;

while (stopWork == false)

{

std::cout << "1. Вийти" << std::endl;

std::cout << "2. Редагувати весь рецепт" << std::endl;

std::cout << "3. Дописати ще 1 або кілька препаратів" << std::endl;

std::cout << "4. Видалити 1 чи кілька препаратів" << std::endl;

std::cin >> choice;

std::cout << std::endl;

switch (choice)

{

case 1: stopWork = true; break;

case 2: list1.clear(); addItem(); break;

case 3: addItem(); break;

case 4: deleteItem(); break;

default: std::cout << "Помилка при виборі пункту меню." << std::endl; break;

}

}

return std::vector<std::string>(list1);

}

std::vector<std::string> Recipe::deleteItem()

{

bool stopWork = false;

int n;

unsigned choice;

while (stopWork == false)

{

std::cout << "Який препарат ви хочете видалити з рецепту?" << std::endl;

ShowInformation();

std::cin >> n;

std::cout << std::endl;

while (n <= 0 || n > list1.size())

{

std::cout << "Ви ввели некоректне значення, спробуйте ще раз." << std::endl;

std::cin >> n;

std::cout << std::endl;

}

list1.erase(list1.begin()+(n - 1));

std::cout << "Елемент успішно видалено." << std::endl;

bool stopWork1 = false;

while (stopWork1 == false)

{

std::cout << "Бажаєте продовжити видалення?" << std::endl;

std::cout << "1. Так" << std::endl;

std::cout << "2. Ні" << std::endl;

std::cin >> choice;

std::cout << std::endl;

switch (choice)

{

case 1: stopWork1 = true; break;

case 2: stopWork = true; stopWork1 = true; break;

default: std::cout << "Помилка при виборі пункту меню." << std::endl; break;

}

}

}

return std::vector<std::string>(list1);

}

std::vector<std::string> Recipe::addItem()

{

std::cout << "Яку кількість препаратів ви хочете записати в рецепт?" << std::endl;

unsigned n;

std::cin >> n;

std::cout << std::endl;

std::cout << "Введіть назви препаратів:" << std::endl;

for (int i = 0; i < n; i++)

{

std::string name;

std::cin >> name;

list1.push\_back(name);

}

std::cout << std::endl;

std::cout << "Рецепт успішно відредаговано." << std::endl;

return std::vector<std::string>(list1);

}

std::vector<std::string> Recipe::EditRecipe()

{

ShowInformation();

bool stopWork = false;

unsigned choice;

while (stopWork == false)

{

std::cout << "Виберіть, що саме ви хочете зробити:" << std::endl;

std::cout << "1. Вийти" << std::endl;

std::cout << "2. Перейти до редагування рецепту" << std::endl;

std::cin >> choice;

std::cout << std::endl;

switch (choice)

{

case 1: stopWork = true; break;

case 2: changeRecipe(); break;

default: std::cout << "Помилка при виборі пункту меню." << std::endl; break;

}

}

return std::vector<std::string>(list1);

}

Recipe.h

#pragma once

#include <iostream>

#include <string>

#include <vector>

#include <Windows.h>

class Recipe

{

std::vector<std::string> list1;

std::vector<std::string> deleteItem();

std::vector<std::string> changeRecipe();

std::vector<std::string> addItem();

public:

Recipe();

void ShowInformation();

std::vector<std::string> EditRecipe();

};

Medicine.cpp

#include "Medicine.h"

Medicine::Medicine(std::string TheNameOfTheDrug, std::string instruction, double value)

{

this->TheNameOfTheDrug = TheNameOfTheDrug;

this->instruction = instruction;

this->value = value;

}

std::string Medicine::GetName()

{

return TheNameOfTheDrug;

}

void Medicine::SetName(std::string TheNameOfTheDrug)

{

this->TheNameOfTheDrug = TheNameOfTheDrug;

}

std::string Medicine::GetInstruction()

{

return instruction;

}

void Medicine::SetInstruction(std::string instruction)

{

this->instruction = instruction;

}

double Medicine::GetValue()

{

return value;

}

void Medicine::SetValue(double value)

{

this->value = value;

}

void Medicine::ShowInformation()

{

std::cout << "Назва препарату: " << GetName() << std::endl;

std::cout << "Інструкція: " << GetInstruction() << std::endl;

std::cout << "Вартість: " << GetValue() << std::endl;

}

Medicine.h

#pragma once

#include <iostream>

#include <string>

class Medicine

{

std::string TheNameOfTheDrug;

std::string instruction;

double value;

public:

Medicine(std::string TheNameOfTheDrug, std::string instruction, double value);

std::string GetName();

void SetName(std::string TheNameOfTheDrug);

std::string GetInstruction();

void SetInstruction(std::string instruction);

double GetValue();

void SetValue(double value);

void ShowInformation();

};

Pharmacy.cpp

#include "Pharmacy.h"

Pharmacy::Pharmacy()

{

list2.push\_back(Medicine("Парацетамол", "Приймати по 1 таблетці 3 рази на добу", 30.2));

list2.push\_back(Medicine("Стрепсілс", "Приймати по 1 таблетці кожних 3-4 години", 90));

list2.push\_back(Medicine("Пульмолор", "Приймати по 1 таблетці 4 рази на добу", 66.6));

list2.push\_back(Medicine("Пульмобріз", "Приймати по 1 таблетці 4 рази на добу", 124.8));

list2.push\_back(Medicine("Нурофен", "Приймати по 1 таблетці 3 рази на добу", 134.3));

}

void Pharmacy::ShowInformation()

{

for (auto& drug : list2)

{

drug.Medicine::ShowInformation();

std::cout << std::endl;

}

}

void Pharmacy::SearchMedicine()

{

std::cout << "Введіть назву препарату, який шукаєте: " << std::endl;

std::string name;

std::cin >> name;

bool found = false;

double value1;

std::cout << std::endl;

for (auto drug : list2)

{

if (drug.GetName() == name)

{

found = true;

value1 = drug.GetValue();

break;

}

}

if (found)

{

std::cout << "Препарат '" << name << "' в наявності. Ціна: " << value1<< std::endl;

}

else

{

std::cout << "Препарат '" << name << "' не в наявності." << std::endl;

}

}

Pharmacy.h

#pragma once

#include "Medicine.h"

#include "Recipe.h"

#include <list>

class Pharmacy

{

std::list<Medicine> list2;

public:

Pharmacy();

void ShowInformation();

void SearchMedicine();

};