**Курс**: ПЛАТФОРМА MICROSOFT .NET ТА МОВА ПРОГРАМУВАННЯ C#

**Тема:** Взаємодія з файловою системою

**Модуль 12. Часть 2**

# Завдання 1:

Створіть додаток для роботи з колекцією рецептів. Рецепт містить наступні дані:

* назву рецепту;
* назву кухні, звідки родом рецепт (наприклад, італійська або японська);
* назву інгредієнтів;
* час приготування;
* покроковий опис процесу приготування. Додаток має давати можливість:
* додавати рецепти;
* видаляти рецепти;
* змінювати рецепти;
* шукати рецепти за різними характеристиками;
* зберігати рецепти у файл;
* завантажувати рецепти з файлу.

# Завдання 2:

Доповніть додаток з першого завдання можливістю генерувати звіти. Звіт може виводитися на екран або зберігатися у файл. Створіть такі звіти:

* за видом кухні;
* за часом приготування;
* за назвами інгредієнтів;
* за назвою рецепта.

# Завдання 3:

Доповніть додаток з першого завдання додатковими характеристиками:

* калорії інгредієнтів;
* тип страви: салат, перше, друге, закуска, десерт тощо. Створіть додаткові звіти:
* за сумою калорій;
* за типом страви;
* за комбінацією типів страв. Наприклад, звіт, який генерує комбінацію страв: закуска, салат, перше, друге, десерт тощо.

using System;

using System.IO;

using System.Text;

using System.Collections.Generic;

using System.Diagnostics;

namespace Dz

{

class Resipe

{

string name;

string kitchen;

string ingredients;

DateTime time\_to\_cook;

string description\_of\_cooking;

int kilocalories;

string type\_of\_dish;

public string Name

{

set { name = value; }

get { return name; }

}

public string Kitchen

{

set { kitchen = value; }

get { return kitchen; }

}

public string Ingredients

{

set { ingredients = value; }

get { return ingredients; }

}

public DateTime Time\_to\_cook

{

set { time\_to\_cook = value; }

get { return time\_to\_cook; }

}

public string Description\_of\_cooking

{

set { description\_of\_cooking = value; }

get { return description\_of\_cooking; }

}

public int Kilocalories

{

set { kilocalories = value; }

get { return kilocalories; }

}

public string Type\_of\_dish

{

set { type\_of\_dish = value; }

get { return type\_of\_dish; }

}

public Resipe(string name, string kitchen, string ingredients, DateTime time\_to\_cook, string description\_of\_cooking, int kilocalories, string type\_of\_dish)

{

this.name = name;

this.kitchen = kitchen;

this.ingredients = ingredients;

this.time\_to\_cook = time\_to\_cook;

this.description\_of\_cooking = description\_of\_cooking;

this.kilocalories = kilocalories;

this.type\_of\_dish = type\_of\_dish;

}

public override string ToString()

{

return name + "\nКухня: " + kitchen + "\nІнгрідієнти: " + ingredients + "\nЧас для приготування: " + time\_to\_cook.ToShortTimeString() + "\nОпис процесу приготування:\n" + description\_of\_cooking + "\nКалорійність: " + kilocalories + "ккал\nТип страви: " + type\_of\_dish;

}

}

class RecipeBook

{

List<Resipe> book;

public RecipeBook()

{

book = new List<Resipe>();

}

public void AddRecipe(Resipe resipe)

{

book.Add(resipe);

}

public void DeleteRecipe(string name)

{

int index = -1;

for(int i=0;i< book.Count;i++)

{

if (string.Compare(book[i].Name,name)==0)

{

index = i;

break;

}

}

if (index != -1)

{

book.RemoveAt(index);

}

else

Console.WriteLine("Не знайдено!");

}

public void PrintAll()

{

for(int i=0;i<book.Count;i++)

{

Console.WriteLine(book[i]);

Console.WriteLine("-----------------------------------------------------");

}

}

public void RedactRecipe(string name)

{

int index = -1;

for (int i = 0; i < book.Count; i++)

{

if (string.Compare(book[i].Name, name) == 0)

{

index = i;

break;

}

}

if (index != -1)

{

string tmp;

int h, m;

Console.WriteLine("Редагувати:\nНазву рецепта->1\nНазва кухні->2\nІнгрідієнти->3\nЧас на приготування->4\nОпис процесу приготування->5\nКалорійність->6\nТип страви->7\nВихід->0");

Console.Write("->");

int k = Convert.ToInt32(Console.ReadLine());

while(k!=0)

{

switch(k)

{

case 1:

Console.Write("Введіть нову назву рецепту->");

book[index].Name=Console.ReadLine();

break;

case 2:

Console.Write("Введіть нову назву кухні->");

book[index].Kitchen = Console.ReadLine();

break;

case 3:

Console.Write("Введіть інгрідієнти->");

book[index].Ingredients = Console.ReadLine();

break;

case 4:

Console.WriteLine("Введіть новий час на приготування:");

Console.Write("Години->");

h = Convert.ToInt32(Console.ReadLine());

Console.Write("Хвилини->");

m = Convert.ToInt32(Console.ReadLine());

book[index].Time\_to\_cook = new DateTime(1, 1, 1, h, m, 1);

break;

case 5:

Console.Write("Введіть новий опис приготування->");

book[index].Description\_of\_cooking = Console.ReadLine();

break;

case 6:

Console.Write("Введіть нову калорійність->");

book[index].Kilocalories = Convert.ToInt32(Console.ReadLine());

break;

case 7:

Console.Write("Введіть новий тип страви->");

book[index].Type\_of\_dish = Console.ReadLine();

break;

}

Console.Clear();

Console.WriteLine("Редагувати:\nНазву рецепта->1\nНазва кухні->2\nІнгрідієнти->3\nЧас на приготування->4\nОпис процесу приготування->5\nКалорійність->6\nТип страви->7\nВихід->0");

Console.Write("->");

k = Convert.ToInt32(Console.ReadLine());

}

}

else

Console.WriteLine("Не знайдено!");

}

public void SearchByName(string name)

{

bool b = false;

for(int i=0; i<book.Count;i++)

{

if(string.Compare(name,book[i].Name)==0)

{

b = true;

Console.WriteLine(book[i]);

Console.WriteLine("---------------------------------------");

}

}

if (!b)

Console.WriteLine("Не знайдено");

}

public void SearchByKitchen(string name)

{

bool b = false;

for (int i = 0; i < book.Count; i++)

{

if (string.Compare(name, book[i].Kitchen) == 0)

{

b = true;

Console.WriteLine(book[i]);

Console.WriteLine("---------------------------------------");

}

}

if (!b)

Console.WriteLine("Не знайдено");

}

public void SearchByTypeOfDish(string type)

{

bool b = false;

for (int i = 0; i < book.Count; i++)

{

if (string.Compare(type, book[i].Type\_of\_dish) == 0)

{

b = true;

Console.WriteLine(book[i]);

Console.WriteLine("---------------------------------------");

}

}

if (!b)

Console.WriteLine("Не знайдено");

}

public void Save()

{

FileStream fs = new FileStream("book.txt", FileMode.Create, FileAccess.Write, FileShare.Write);

StreamWriter streamWriter = new StreamWriter(fs, Encoding.UTF8);

streamWriter.WriteLine(book.Count.ToString());

for(int i=0;i<book.Count;i++)

{

streamWriter.WriteLine(book[i].Name);

streamWriter.WriteLine(book[i].Kitchen);

streamWriter.WriteLine(book[i].Ingredients);

streamWriter.WriteLine(book[i].Time\_to\_cook.Hour.ToString());

streamWriter.WriteLine(book[i].Time\_to\_cook.Minute.ToString());

streamWriter.WriteLine(book[i].Description\_of\_cooking);

streamWriter.WriteLine(book[i].Kilocalories.ToString());

streamWriter.WriteLine(book[i].Type\_of\_dish);

}

streamWriter.Dispose();

fs.Close();

}

public void Read()

{

FileStream fs = new FileStream("book.txt", FileMode.Open, FileAccess.Read, FileShare.Read);

StreamReader streamReader = new StreamReader(fs, Encoding.UTF8);

book = new List<Resipe>(Convert.ToInt32(streamReader.ReadLine()));

for(int i=0;i<book.Capacity;i++)

{

book.Add(new Resipe(streamReader.ReadLine(), streamReader.ReadLine(), streamReader.ReadLine(), new DateTime(1, 1, 1, Convert.ToInt32(streamReader.ReadLine()), Convert.ToInt32(streamReader.ReadLine()), 1), streamReader.ReadLine(), Convert.ToInt32(streamReader.ReadLine()), streamReader.ReadLine()));

}

streamReader.Dispose();

fs.Close();

}

bool IsNeIs(List<string> l,string txt)

{

for(int i=0;i<l.Count;i++)

{

if(string.Compare(l[i],txt)==0)

{

return true;

}

}

return false;

}

public void CreateReportByKitchen()

{

List<string> l=new List<string>();

for(int i=0;i<book.Count;i++)

{

if(!IsNeIs(l,book[i].Kitchen))

{

l.Add(book[i].Kitchen);

}

}

Console.WriteLine(DateTime.Now);

Console.WriteLine("Кухні:");

for (int i=0;i<l.Count;i++)

{

Console.WriteLine(l[i]);

}

Console.WriteLine("------------------------");

Console.Write("Зберегти в файл ->1\nВихід ->0\n->");

int k = Convert.ToInt32(Console.ReadLine());

if (k == 1)

{

FileStream fs = new FileStream("Report.txt", FileMode.Append, FileAccess.Write, FileShare.Write);

StreamWriter streamWriter = new StreamWriter(fs, Encoding.UTF8);

streamWriter.WriteLine(DateTime.Now);

streamWriter.WriteLine("Кухні:");

for (int i = 0; i < l.Count; i++)

{

streamWriter.WriteLine(l[i]);

}

streamWriter.WriteLine("------------------------");

streamWriter.Dispose();

fs.Close();

}

}

public void CreateReportByTimeForCooking()

{

string txt = "";

Console.WriteLine(DateTime.Now);

Console.WriteLine("Часи для приготування:");

for(int i=0;i<book.Count;i++)

{

txt += book[i].Name + " ->" + book[i].Time\_to\_cook.ToShortTimeString() + "\n";

}

txt += "------------------------";

Console.WriteLine(txt);

Console.Write("Зберегти в файл ->1\nВихід ->0\n->");

int k = Convert.ToInt32(Console.ReadLine());

if (k == 1)

{

FileStream fs = new FileStream("Report.txt", FileMode.Append, FileAccess.Write, FileShare.Write);

StreamWriter streamWriter = new StreamWriter(fs, Encoding.UTF8);

streamWriter.WriteLine(DateTime.Now);

streamWriter.WriteLine("Часи для приготування:");

streamWriter.WriteLine(txt);

streamWriter.Dispose();

fs.Close();

}

}

public void CreateReportByIngridients()

{

string txt = "";

Console.WriteLine(DateTime.Now);

Console.WriteLine("Інгрідієнти для рецептів:");

for (int i = 0; i < book.Count; i++)

{

txt += book[i].Name + " ->" + book[i].Ingredients + "\n";

}

txt += "------------------------";

Console.WriteLine(txt);

Console.Write("Зберегти в файл ->1\nВихід ->0\n->");

int k = Convert.ToInt32(Console.ReadLine());

if (k == 1)

{

FileStream fs = new FileStream("Report.txt", FileMode.Append, FileAccess.Write, FileShare.Write);

StreamWriter streamWriter = new StreamWriter(fs, Encoding.UTF8);

streamWriter.WriteLine(DateTime.Now);

streamWriter.WriteLine("Інгрідієнти для рецептів:");

streamWriter.WriteLine(txt);

streamWriter.Dispose();

fs.Close();

}

}

public void CreateReportByName()

{

string txt = "";

Console.WriteLine(DateTime.Now);

Console.WriteLine("Доступні рецепти:");

for (int i = 0; i < book.Count; i++)

{

txt += book[i].Name + "\n";

}

txt += "------------------------";

Console.WriteLine(txt);

Console.Write("Зберегти в файл ->1\nВихід ->0\n->");

int k = Convert.ToInt32(Console.ReadLine());

if (k == 1)

{

FileStream fs = new FileStream("Report.txt", FileMode.Append, FileAccess.Write, FileShare.Write);

StreamWriter streamWriter = new StreamWriter(fs, Encoding.UTF8);

streamWriter.WriteLine(DateTime.Now);

streamWriter.WriteLine("Доступні рецепти:");

streamWriter.WriteLine(txt);

streamWriter.Dispose();

fs.Close();

}

}

public void CreateReportByCalories()

{

int suma = 0;

Console.WriteLine(DateTime.Now);

Console.WriteLine("Доступні рецепти:");

for (int i = 0; i < book.Count; i++)

{

suma += book[i].Kilocalories;

}

Console.WriteLine("Сума кілокалорій за всіма рецептами: " + suma + "ккал");

Console.Write("Зберегти в файл ->1\nВихід ->0\n->");

int k = Convert.ToInt32(Console.ReadLine());

if (k == 1)

{

FileStream fs = new FileStream("Report.txt", FileMode.Append, FileAccess.Write, FileShare.Write);

StreamWriter streamWriter = new StreamWriter(fs, Encoding.UTF8);

streamWriter.WriteLine(DateTime.Now);

streamWriter.WriteLine("Сума кілокалорій за всіма рецептами: " + suma.ToString() + "ккал");

streamWriter.WriteLine("------------------------");

streamWriter.Dispose();

fs.Close();

}

}

public void CreateReportByTypeOfDishes()

{

List<string> l = new List<string>();

for (int i = 0; i < book.Count; i++)

{

if (!IsNeIs(l, book[i].Type\_of\_dish))

{

l.Add(book[i].Type\_of\_dish);

}

}

Console.WriteLine(DateTime.Now);

Console.WriteLine("Види страв:");

for (int i = 0; i < l.Count; i++)

{

Console.WriteLine(l[i]);

}

Console.WriteLine("------------------------");

Console.Write("Зберегти в файл ->1\nВихід ->0\n->");

int k = Convert.ToInt32(Console.ReadLine());

if (k == 1)

{

FileStream fs = new FileStream("Report.txt", FileMode.Append, FileAccess.Write, FileShare.Write);

StreamWriter streamWriter = new StreamWriter(fs, Encoding.UTF8);

streamWriter.WriteLine(DateTime.Now);

streamWriter.WriteLine("Види страв:");

for (int i = 0; i < l.Count; i++)

{

streamWriter.WriteLine(l[i]);

}

streamWriter.WriteLine("------------------------");

streamWriter.Dispose();

fs.Close();

}

}

public void CreateReportByCombination(string types)

{

string txt = "";

bool b = false;

int k = 0;

var tmp = types.Split(" ".ToCharArray());

for(int i=0;i<tmp.Length;i++)

{

for(int j=0;j<book.Count;j++)

{

if(string.Compare(tmp[i],book[j].Type\_of\_dish)==0)

{

txt += book[j].Name+"\n";

b = true;

k++;

break;

}

}

if (!b)

break;

else

b = false;

}

if (k == tmp.Length)

{

Console.WriteLine(DateTime.Now);

Console.WriteLine("Комбінація з таких страв: " + types);

Console.Write(txt);

Console.WriteLine("------------------------");

Console.Write("Зберегти в файл ->1\nВихід ->0\n->");

int m = Convert.ToInt32(Console.ReadLine());

if (m == 1)

{

FileStream fs = new FileStream("Report.txt", FileMode.Append, FileAccess.Write, FileShare.Write);

StreamWriter streamWriter = new StreamWriter(fs, Encoding.UTF8);

streamWriter.WriteLine(DateTime.Now);

streamWriter.WriteLine("Комбінація з таких страв: " + types);

streamWriter.Write(txt);

streamWriter.WriteLine("------------------------");

streamWriter.Dispose();

fs.Close();

}

}

else

Console.WriteLine("Немає таких видів страв");

}

}

public class Example

{

public static void Main()

{

System.Console.OutputEncoding = System.Text.Encoding.UTF8;

string name;

string kitchen;

string ingredients;

string description\_of\_cooking;

int kilocalories;

string type\_of\_dish;

int h, m;

Resipe resipe = new Resipe("Олів'є", "Україна", "ковбаса, картопля, огірки, яйця, морква", new DateTime(1, 1, 1, 3, 30, 1), "Чистимо овочі, відварюємо, нарізаємо кубиками", 339, "Салат");

Resipe resipe2 = new Resipe("Борщ", "Китай", "ковбаса, картопля, огірки, яйця, морква", new DateTime(1, 1, 1, 3, 30, 1), "Чистимо овочі, відварюємо, нарізаємо кубиками", 339, "Борщ");

RecipeBook recipeBook = new RecipeBook();

Console.WriteLine("Додати рецепт->1\nВидалити рецепт->2\nЗберегти книгу рецептів->3\nЗчитати книгу рецептів з файлу->4\nЗнайти рецепт за іменем->5\nЗнайти рецепт за кухнею->6\nРедагувати рецепт->7\nПоказати всі рецепти->8\nСтворити звіт за:\nЗа видом кухні->9\nЗа часом приготування->10\nЗа назвами інгрідієнтів->11\nЗа назвою рецепта->12\nЗа сумою калорій->13\nЗа видом страви->14\nЗа комбінацією типом страв->15\nВихід->0");

Console.Write("->");

int k = Convert.ToInt32(Console.ReadLine());

while(k!=0)

{

switch(k)

{

case 1:

Console.Write("Введіть назву рецепта->");

name = Console.ReadLine();

Console.Write("Введіть назву кухні->");

kitchen = Console.ReadLine();

Console.Write("Введіть назву інгрідієнтів->");

ingredients = Console.ReadLine();

Console.Write("Введіть опис процесу приготування->");

description\_of\_cooking = Console.ReadLine();

Console.Write("Введіть калорійність->");

kilocalories = Convert.ToInt32(Console.ReadLine());

Console.Write("Введіть вид страви->");

type\_of\_dish = Console.ReadLine();

Console.WriteLine("Введіть час на приготування:");

Console.Write("Години->");

h = Convert.ToInt32(Console.ReadLine());

Console.Write("Хвилини->");

m = Convert.ToInt32(Console.ReadLine());

recipeBook.AddRecipe(new Resipe(name, kitchen, ingredients, new DateTime(1, 1, 1, h, m, 1), description\_of\_cooking, kilocalories, type\_of\_dish));

Console.Clear();

Console.WriteLine("Успішно додано");

break;

case 2:

Console.Write("Введіть назву рецепта->");

name = Console.ReadLine();

recipeBook.DeleteRecipe(name);

break;

case 3:

recipeBook.Save();

Console.WriteLine("Успішно збежено");

break;

case 4:

recipeBook.Read();

Console.WriteLine("Успішно зчитано");

break;

case 5:

Console.Write("Введіть назву рецепта->");

name = Console.ReadLine();

recipeBook.SearchByName(name);

break;

case 6:

Console.Write("Введіть назву кухні->");

kitchen = Console.ReadLine();

recipeBook.SearchByKitchen(kitchen);

break;

case 7:

Console.Write("Введіть назву рецепта->");

name = Console.ReadLine();

recipeBook.RedactRecipe(name);

break;

case 8:

recipeBook.PrintAll();

break;

case 9:

recipeBook.CreateReportByKitchen();

break;

case 10:

recipeBook.CreateReportByTimeForCooking();

break;

case 11:

recipeBook.CreateReportByIngridients();

break;

case 12:

recipeBook.CreateReportByName();

break;

case 13:

recipeBook.CreateReportByCalories();

break;

case 14:

recipeBook.CreateReportByTypeOfDishes();

break;

case 15:

Console.Write("Введіть види страв через пробіл->");

type\_of\_dish = Console.ReadLine();

recipeBook.CreateReportByCombination(type\_of\_dish);

break;

}

Console.WriteLine("Додати рецепт->1\nВидалити рецепт->2\nЗберегти книгу рецептів->3\nЗчитати книгу рецептів з файлу->4\nЗнайти рецепт за іменем->5\nЗнайти рецепт за кухнею->6\nРедагувати рецепт->7\nПоказати всі рецепти->8\nСтворити звіт за:\nЗа видом кухні->9\nЗа часом приготування->10\nЗа назвами інгрідієнтів->11\nЗа назвою рецепта->12\nЗа сумою калорій->13\nЗа видом страви->14\nЗа комбінацією типом страв->15\nВихід->0");

Console.Write("->");

k = Convert.ToInt32(Console.ReadLine());

Console.Clear();

}

}

}

}