# КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ТАРАСА ШЕВЧЕНКА



## ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

**Кафедра прикладних інформаційних систем**

**Звіт до лабораторної роботи №3**

# з курсу

**«Алгоритми і структури даних»**

*Студента 1 курсу*

*групи ПП-11 спеціальності 122 «Комп'ютерні науки» ОП «Прикладне програмування»*

%username%

*Викладач:*

д.е.н., к.т.н., проф. Плескач В.Л.

к.ф.-м.н., доц. Шолохов О.В.

## Київ – 202

**1.Назва роботи**

Робота зі списками

1. **Тема роботи**

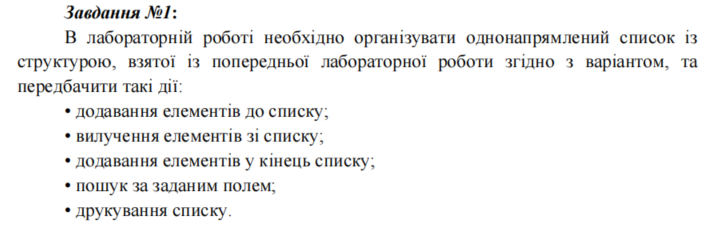
Робота зі списками

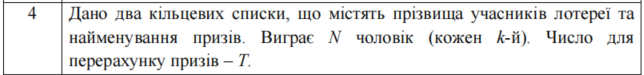
1. **Мета роботи**

Навчитися складати програми для роботи зі списками.

1. **Умова завдання**

(4 варіант)



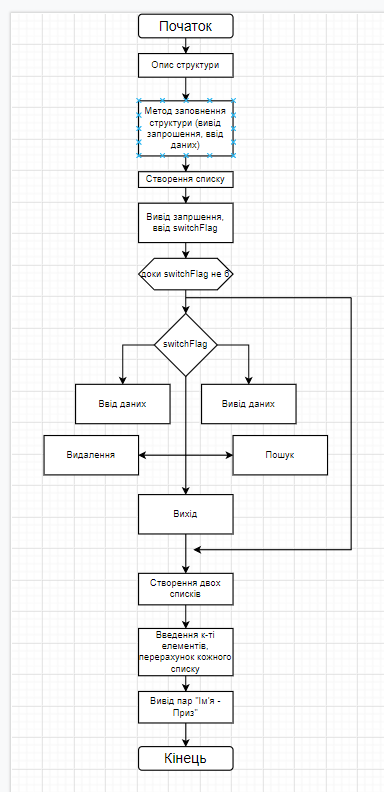


1. **Рішення**

Задача 1:

Описуємо структуру, виводимо запрошення, далі в залежності від вибору користувача застосовуємо методи (зазвичай це ввід даних, вивід списку, видалення та додавання елементів, пошук і вивід ще раз).

Блоксхема:



Код:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Лаба4 {

struct Aeroflot {

public string city;

public int num;

public string type;

}

class Program {

static Aeroflot structureFiller() {

Aeroflot currentStructure = new Aeroflot();

Console.WriteLine("Please enter the city:");

currentStructure.city = Console.ReadLine();

Console.WriteLine("Please enter the number:");

currentStructure.num = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Please enter the type:");

currentStructure.type = Console.ReadLine();

return currentStructure;

}

static void taskOne() {

LinkedList<Aeroflot> listOne = new LinkedList<Aeroflot>();

Console.WriteLine("PLS ENTER OPTN (1 to add, 2 to del, 3 to add to end, 4 to find, 5 to print, 6 to exit)");

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

while (switchFlag != 6) {

switch (switchFlag) {

case 1:

Console.WriteLine("How many elements do you want to add?");

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

Console.WriteLine("Begin entering the data:");

for (int q = 0; q < numOfElements; q++) {

listOne.AddFirst(structureFiller());

}

Console.WriteLine("What's next?");

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

break;

case 2:

Console.WriteLine("Removing...");

listOne.RemoveFirst();

Console.WriteLine("What's next?");

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

break;

case 3:

Console.WriteLine("Begin enering the data:");

listOne.AddLast(structureFiller());

Console.WriteLine("What's next?");

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

break;

case 4:

Console.WriteLine("Please enter the element you want to search for (1 for city, 2 for num, 3 for type):");

int switchFlag2 = Convert.ToInt32(Console.ReadLine());

switch (switchFlag2) {

case 1:

Console.WriteLine("Enter the city:");

string desiredCity = Console.ReadLine();

foreach (Aeroflot currentStructure in listOne) {

if (currentStructure.city == desiredCity) {

Console.WriteLine("Match! It's the #" + currentStructure.num + "flight of " + currentStructure.type + " type to " + currentStructure.city + " !");

}

}

break;

case 2:

Console.WriteLine("Enter the number:");

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

foreach (Aeroflot currentStructure in listOne) {

if (currentStructure.num == desiredNum) {

Console.WriteLine("Match! It's the #" + currentStructure.num + "flight of " + currentStructure.type + " type to " + currentStructure.city + " !");

}

}

break;

case 3:

Console.WriteLine("Enter the type:");

string desiredType = Console.ReadLine();

foreach (Aeroflot currentStructure in listOne) {

if (currentStructure.city == desiredType) {

Console.WriteLine("Match! It's the #" + currentStructure.num + " flight of " + currentStructure.type + " type to " + currentStructure.city + " !");

}

}

break;

}

Console.WriteLine("What's next?");

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

break;

case 5:

Console.WriteLine("Here goes:");

foreach (Aeroflot currentStructure in listOne) {

Console.WriteLine("Here's the city:");

Console.WriteLine(currentStructure.city);

Console.WriteLine("Here's the number:");

Console.WriteLine(currentStructure.num);

Console.WriteLine("Here's the type:");

Console.WriteLine(currentStructure.type);

}

Console.WriteLine("What's next?");

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

break;

}

}

}

static void taskTwo() {

string[] namesRaw = { "Pink", "Blonde", "Orange", "Blue", "White", "Brown" };

string[] prizesRaw = { "Candy", "Money", "Bullet", "Plush", "Ring", "Token" };

LinkedList<string> names = new LinkedList<string>(namesRaw);

LinkedList<string> prizes = new LinkedList<string>(prizesRaw);

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

for (int q = 0; q < numOfWinners; q++) {

Console.WriteLine(namesRaw[q] + " won " + prizesRaw[q]);

}

}

static void Main(string[] args) {

taskOne();

taskTwo();

}

}

}

Результат роботи програми:

[screenshot]

[screenshot]

[screenshot]

[screenshot]

1. **Контрольні запитання**
2. Поняття структури даних

Структура даних - спосіб організації даних у пам’яті.

1. Рівні подання структур даних

Модель подання структури даних описує логіку взаємозв’язків між даними. Існують такі моделі структур даних: **ієрархічна**, де дані становлять множину,частково упорядковану так, що існує лише один елемент цієї множини, який не має попереднього, а всі інші елементи мають лише один попередній; **мережна,** де дані становлять множину, частково впорядковану так, що принаймні для деяких елементів множини існує більше, ніж один попередній елемент; **реляційна**, де дані є сукупністю взаємопов’язаних таблиць; **об’єктно-орієнтована**, де дані є сукупністю взаємопов’язаних об’єктів.

1. Основні типи структур даних

Алгебричний тип даних узагальнений

Асоціативний масив

Клас

Тип-добуток (запис-кортеж-структура)

Масив

Множина

Об'єднання

Об'єкт, метаоб'єкт

1. Класифікація структур даних у програмах користувача й у пам'яті комп'ютера

Представлення даних у програмах залежить від мови програмування, але найчастіше використовують наступні типи даних: скаляри, одновимірні масиви та записи. Часто використовують структури даних типу множин, зв’язаних списків та таблиць. У пам’яті комп’ютера дані можуть бути впорядкованими/невпорядкованими, однорідними/неоднорідними, містити/не містити ключі. Неупорядковані представлені множинами, впорядковані - чергою, стеком, списком або масивом.

1. Приклади структур даних

Масив, зв’язаний список, запис (власне структура)

1. Порівняти різні схеми зберігання даних

Кожен спосіб зберігання даних використовується для специфікованої задачі (наприклад, для зберігання номеру аеропорту доцільно використати одну змінну, а для бази даних про рейси - масив структур), а отже, порівняти їх без контексту неможливо.

1. Навести приклади прямого доступу до даних.

У С# можна навести приклад запису інформації відразу до змінної без використання аксесорів класу чи методу set.

1. **Висновки**

В результаті виконання даної лабораторної роботи я оволодів практичними навичками використання структур і масивів структур, навчився складати програми для виконання операцій з полями структур, навчився використовувати динамічне виділення пам’яті. Реалізував ручну ініціалізацію масиву, його сортування від меншого до більшого та вивід на екран. Вважаю дану лабораторну роботу виконаною в повному обсязі.