ЛАБОРАТОРНАЯ РАБОТА № 7

**Составление программ с описанием множественного типа данных.**

Выполнил Юнусов Максим ИСП21.1А

Вариант№24.

Цели работы

1. Получение навыков в задании переменных множественного типа и выполнение простейших операций над ними.

2. Знакомство с задачами, в которых целесообразно использовать переменные множественного типа.

3. Получение навыков в организации ввода/вывода значений множественных типов.

4. Получение практических навыков в выполнении операций над множествами.

1. Практическое задание

Практическое задание включает в себя:

– знакомство с конечным и упорядоченным множеством символов. Нужно составить программу согласно своему варианту.

– это построение множеств и выполнение операций над множествами, составление программы согласно своему варианту.

Методические указания

– Определить алгоритм решения задачи, записать намеченный алгоритм в виде блок-схемы программы.

– Подобрать контрольный пример.

– После чего приступить к написанию программы.

Задание 1

Постановка задачи.

Дана непустая последовательность символов. Требуется построить и напечатать множество, элементами которого являются встречающиеся в последовательности символы согласно варианту.

**Варианты заданий**

8 Знаки арифметических операций и знаки препинания.

Задание 2

Постановка задачи. Придумать список некоторых объектов, например, названия товаров в магазинах; виды полезных ископаемых, которые добывают в различных странах; фамилии студентов, которые могут принимать участие в работе определенных кружков и т. п. Построить пять множеств (язык слов англ.), которые складываются из различных комбинаций элементов этого списка.

* Вывести на экран названия объектов, которые *находятся во всех множествах* *одновременно.*
* Определить, какие объекты *не входят ни в одно множество*.
* Выполнить дополнительно задание своего варианта, а именно, определить, какие объекты:

**Варианты заданий**

8 Входят в первое и третье множества, но не входят в пятое.

Задание 1

class Program

{

static void Main(string[] args)

{

string line = Console.ReadLine(); //исходная строка

HashSet<char> setLine = TranslateLineToSet(line); //множество символов исходной строки

PrintSet(setLine);

HashSet<char> setDigit = new HashSet<char> { '+', '=', '-','/','\*','%','^','~',',', '.', '-', '!', '?', ';', ':', ')', '(', '"',']','[','{','}' };// множество символов, которые нужно найти

PrintSet(setDigit);

HashSet<char> newSet = setLine; // копия исходного множества

newSet.IntersectWith(setDigit); //пересечение

PrintSet(newSet);

Console.ReadKey();

}

private static HashSet<char> TranslateLineToSet(string word)

{

HashSet<char> setWord = new HashSet<char>();

foreach (char c in word)

{

setWord.Add(c);

}

return setWord;

}

private static void PrintSet(HashSet<char> set)

{

foreach (char c in set)

{

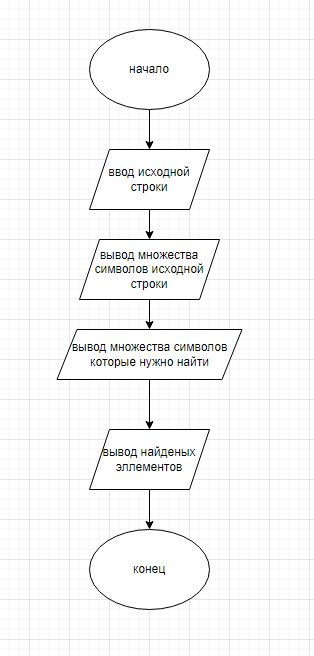
Console.Write($"{c} ");

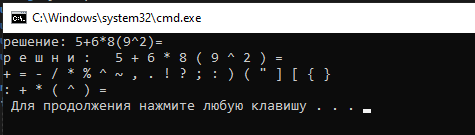
}

Console.WriteLine();

}

}





Задание 2

Console.WriteLine("Задание 2");

string[] allList = {"Strawberry", "Blueberry", "Raspberry", "Apples", "Pears",

"Apricots", "Plums", "Cherry", "Grapes"};

string[] list1 = { "Plums", "Raspberry", "Apples", "Pears" };

string[] list2 = { "Blueberry", "Apples", "Plums", "Apricots" };

string[] list3 = { "Raspberry", "Pears", "Apricots", "Plums" };

string[] list4 = { "Apples", "Plums", "Cherry", "Plums" };

string[] list5 = { "Pears", "Cherry", "Grapes", "Plums" };

PrintSet(allList);

PrintSet(list1);

PrintSet(list2);

PrintSet(list3);

PrintSet(list4);

PrintSet(list5);

Console.WriteLine("Входит во все множества:");

PrintSet(list1.Intersect(list2).Intersect(list3).Intersect(list4).Intersect(list5));

Console.WriteLine("НЕ входит ни в одно множества:");

PrintSet(allList.Except(list1).Except(list2).Except(list3).Except(list4).Except(list5));

Console.WriteLine("Входят в первое и третье множества, но не входят в пятое:");

PrintSet(allList.Intersect(list1).Intersect(list3).Except(list5));

}

private static void PrintSet(IEnumerable<string> set)

{

foreach (string c in set)

{

Console.Write($"{c} ");

}

Console.WriteLine();

}

