МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тихоокеанский государственный университет»

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

Операции над множествами

Лабораторная работа №2

по дисциплине «Алгоритмы дискретной математики»

Выполнил студент Пшеничный Д. О.

Факультет, группа ФКФН, ПО(аб)-81

Руководитель работы  **Резак Е.В.**

Хабаровск – 2020г.

**Исходная информация**: универсальное множество U, подмножества A, B, C

Множества задаются перечислением элементов.

**Требуется**: реализовать операции объединения, пересечения, дополнения, разности для множеств и выполнить вычисления для заданной функции, а также функцию проверки вхождения элемента в множество.

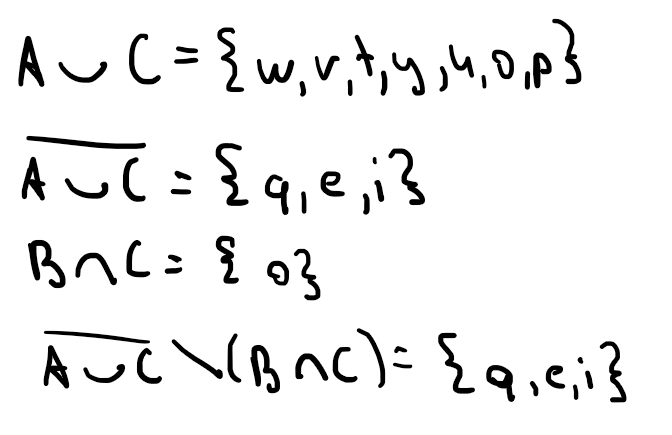
Вариант 16. 

Битовая шкала характеризует принадлежность элемента из U множеству А, поэтому ее также называют характеристическим вектором множества А с U.

**Ручной расчет**

U={q,w,e,r,t,y,u,i,o,p};

A={w,t,y,u}; B={q,w,i,o}; C={r,o,p};



**Листинг**

using System;

namespace Program

{

class Program

{

static void Main(string[] args)

{

Console.Write("Введите универсальное множество: ");

Array U = new Array(Console.ReadLine());

Console.Write("Введите множество A: ");

Array A = new Array(Console.ReadLine(), U);

Console.Write("Введите множество B: ");

Array B = new Array(Console.ReadLine(), U);

Console.Write("Введите множество C: ");

Array C = new Array(Console.ReadLine(), U);

Console.WriteLine("Универсальное множество: " + U.PrintArray() + "\nМножество A: " + A.PrintArray()

+ "\nМножество B: " + B.PrintArray() + "\nМножество C: " + C.PrintArray());

Array ApC = new Array().Addiction(A, C, U);

Console.WriteLine("Объединение А и С: " + ApC.PrintArray());

Array ApCd = new Array().Negation(ApC, U);

Console.WriteLine("Дополнение объединения А и С: " + ApCd.PrintArray());

Array BuC = new Array().Multiplication(B, C, U);

Console.WriteLine("Пересечение B и C: " + BuC.PrintArray());

Console.WriteLine("Разность дополнения объединения A и C и пересечения B и C: " + new Array().Substraction(ApCd, BuC, U).PrintArray());

}

}

}

class Array

{

List<char> list = new List<char>();

public Array(string \_str)

{

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

{

list.Add(\_str[i]);

}

}

public Array(string \_str, Array \_U)

{

for (int i = 0; i < \_U.list.Count; i++)

{

list.Add('-');

for(int j = 0; j < \_str.Length; j++)

{

if(\_U.list[i] == \_str[j])

{

list[i] = \_str[j];

break;

}

}

}

}

public Array()

{

}

public Array Addiction(Array array1, Array array2, Array \_U)

{

Array result = new Array();

for(int i = 0; i < array1.list.Count; i++)

{

if (array1.list[i] != '-' || array2.list[i] != '-')

result.list.Add(\_U.list[i]);

else

result.list.Add('-');

}

return result;

}

public Array Multiplication(Array array1, Array array2, Array U)

{

Array result = new Array();

for (int i = 0; i < array1.list.Count; i++)

{

if (array1.list[i] != '-' && array2.list[i] != '-')

result.list.Add(U.list[i]);

else

result.list.Add('-');

}

return result;

}

public Array Negation(Array array, Array U)

{

Array result = new Array();

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

{

if (array.list[i] == '-')

result.list.Add(U.list[i]);

else

result.list.Add('-');

}

return result;

}

public Array Substraction(Array array1, Array array2, Array U)

{

Array result = new Array();

for (int i = 0; i < array1.list.Count; i++)

{

if (array1.list[i] != '-')

{

if (array1.list[i] != array2.list[i])

result.list.Add(array1.list[i]);

else

result.list.Add('-');

}

else

result.list.Add('-');

}

return result;

}

public string PrintArray()

{

string res = "";

foreach(char i in list)

{

res += i;

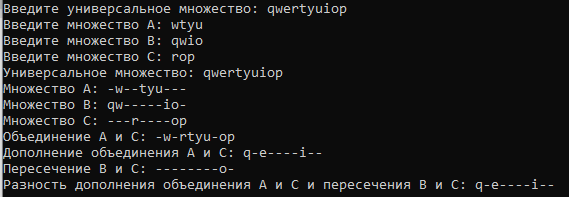
}

return res;

}

}

**Результат работы программы**

****