Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №3

за 1 семестр

По дисциплине: «СПП»

Выполнил:

Студент 3 курса

Группы ПО-5

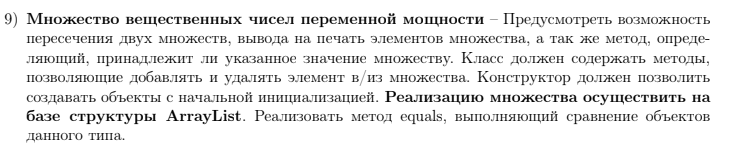
Лозейко П. А.

Проверил:

Крощенко А. А.

2021

Вариант 9



**Код программы:**

**main.cs**

using System;

namespace lab3.\_1.\_9

{

class main

{

static void Main(string[] args)

{

double[] arr1 = { 1.2, 1.1, 3.2, 4.3, 9.3 };

double[] arr2 = { 1.3, 1.1, 3.1, 4.3, 9.1 };

Set set1 = new Set(5, arr1);

Set set2 = new Set(5, arr2);

Set set3 = new Set();

set3.Crossing(set1.List, set2.List);

set3.Print();

set3.FindElement(1.2);

set1.AddElement(123.123, 2);

set1.Print();

Console.WriteLine(set1.Equals(set2));

Console.WriteLine(set2.ToString());

}

}

}

**set.cs**

using System;

using System.Collections.Generic;

using System.Linq;

namespace lab3.\_1.\_9

{

class Set

{

private List<double> list = new List<double>();

public List<double> List

{

get

{

return list;

}

set

{

list = value;

}

}

public void Print()

{

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

{

Console.Write($"{list[i]} ");

}

Console.WriteLine();

}

public void Crossing(List<double> lst1, List<double> lst2)

{

List<double> res = lst1.Union(lst2).ToList();

List.AddRange(res);

List.Sort();

}

public void FindElement(double element)

{

if(List.Exists(x => x == element))

{

Console.WriteLine("Элемент {0} найден", element);

}

else

Console.WriteLine("Элемент {0} не найден", element);

}

public void AddElement(double element, int position)

{

position--;

List.Insert(position, element);

}

public void DeleteElement(int position)

{

position--;

List.RemoveAt(position);

}

public override bool Equals(object obj)

{

if (obj == null)

return false;

Set set = obj as Set; // возвращает null если объект нельзя привести к типу Set

if ((set as Set) == null)

return false;

return set.List == List;

}

public override string ToString()

{

string str = "";

if (List == null)

return base.ToString();

else

{

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

{

str += List[i] + " ";

}

}

return str;

}

public Set()

{

}

public Set(int Size)

{

Random temp = new Random();

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

{

list.Add(temp.NextDouble() \* 20);

}

}

public Set(int Size, double[] arr)

{

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

{

list.Add(arr[i]);

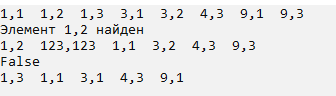
}

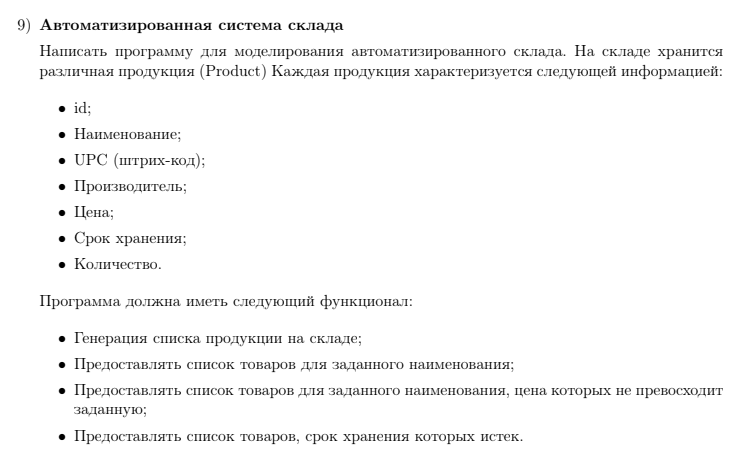
}

}

}

**Результат работы программы:**

****

****

**Код программы:**

**Program.sc**

using System;

using System.Linq;

using System.IO;

using System.Collections.Generic;

namespace lab3.\_2.\_9

{

class Program

{

static Product[] TakeDataFromFile(string[] args\_)

{

string[] arr = ReadFile();

Product[] storage = new Product[Convert.ToInt32(args\_[0])];

char[] separators = new char[] { ' ', '\r', '\n' };

string[] subs;

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

{

storage[i] = new Product();

string str = arr[i];

subs = str.Split(separators, StringSplitOptions.RemoveEmptyEntries);

storage[i].ID = i + 1;

storage[i].Name = subs[0];

storage[i].UPC = subs[1];

storage[i].Producer = subs[2];

storage[i].Price = Convert.ToInt32(subs[3]);

storage[i].Date = DateTime.Parse(subs[4]);

storage[i].Quantity = Convert.ToInt32(subs[5]);

}

return storage;

}

static string[] ReadFile()

{

StreamReader read = new StreamReader("1.txt", System.Text.Encoding.Default);

var list = new List<string>();

while (!read.EndOfStream)

{

string line = read.ReadLine();

list.Add(line);

}

//Массив string[]

string[] arr = list.ToArray();

return arr;

}

static void GenerateStorage(Product[] storage)

{

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

{

storage[i] = new Product();

storage[i].ID = i + 1;

}

}

static void Main(string[] args)

{

Product[] Storage = new Product[Convert.ToInt32(args[0])];

Console.WriteLine("Сгенерировать склад (1) или Загрузить склад из файла (2): ");

switch(Console.ReadLine())

{

case "1":

GenerateStorage(Storage);

Console.WriteLine("Весь список склада:\n");

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

{

Storage[i].Print();

}

Console.WriteLine("\nСписок товаров, срок хранения которых истек:\n");

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

{

Storage[i].PrintDate();

}

Console.WriteLine("\nСписок товаров для заданного наименования:\n");

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

{

Storage[i].PrintName("Карандаш");

}

Console.WriteLine("\nСписок товаров для заданного наименования, цена которых не превосходит заданную:\n");

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

{

Storage[i].PrintNameAndLowerPrices("Карандаш", 5);

}

break;

case "2":

Storage = TakeDataFromFile(args);

Console.WriteLine("Весь список склада:\n");

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

{

Storage[i].Print();

}

Console.WriteLine("\nСписок товаров, срок хранения которых истек:\n");

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

{

Storage[i].PrintDate();

}

Console.WriteLine("\nСписок товаров для заданного наименования:\n");

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

{

Storage[i].PrintName("Карандаш");

}

Console.WriteLine("\nСписок товаров для заданного наименования, цена которых не превосходит заданную:\n");

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

{

Storage[i].PrintNameAndLowerPrices("Карандаш", 5);

}

break;

default:

Console.WriteLine("Введено неверное значение!");

break;

}

}

}

}

**Product.cs**

using System;

using System.Collections.Generic;

using System.Text;

namespace lab3.\_2.\_9

{

class Product

{

private int id;

private string name;

private string upc;

private string producer;

private int price;

private DateTime date;

private int quantity;

public int ID

{

get

{

return id;

}

set

{

id = value;

}

}

public string Name

{

get

{

return name;

}

set

{

name = value;

}

}

public string UPC

{

get

{

return upc;

}

set

{

upc = value;

}

}

public string Producer

{

get

{

return producer;

}

set

{

producer = value;

}

}

public int Price

{

get

{

return price;

}

set

{

price = value;

}

}

public DateTime Date

{

get

{

return date;

}

set

{

date = value;

}

}

public int Quantity

{

get

{

return quantity;

}

set

{

quantity = value;

}

}

public Product()

{

Random temp = new Random();

string[] name\_ = new string[] { "Ручка", "Карандаш", "Тетрадь", "Ластик", "Скребка", "Бумага", "Степлер", "Ножницы" };

string[] producer\_ = new string[] { "Bogue\_Dream", "Cerciocastro", "Fournachem", "Rittenhouse", "Grant\_Design", "Fine\_Lotus", "River\_And\_Neck", "Creative\_Space" };

Name = name\_[temp.Next(0, 8)];

UPC = temp.Next(100000000, 999999999).ToString();

Producer = producer\_[temp.Next(0, 8)];

Price = temp.Next(1, 10);

Date = RandomDay();

Quantity = temp.Next(0, 1001);

}

DateTime RandomDay()

{

Random temp = new Random();

DateTime start = new DateTime(2021, 1, 1);

DateTime end = new DateTime(2025, 1, 1);

int range = (end - start).Days;

return start.AddDays(temp.Next(range));

}

public void Print()

{

Console.WriteLine("ID #{0}\t Наименование: {1}\t UPC: {2}\t Производитель: {3}\t Цена: {4}\t Срок хранения: {5:d}\t Количество: {6}", ID, Name, UPC, Producer, Price, Date, Quantity);

}

public void PrintName(string name)

{

if(name == Name)

{

Print();

}

}

public void PrintNameAndLowerPrices(string name, int price)

{

if(price > Price)

{

PrintName(name);

}

}

public void PrintDate()

{

var periodNow = DateTime.Now;

if (Date < periodNow)

{

Print();

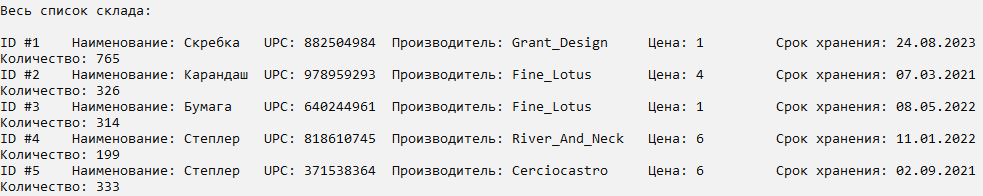
}

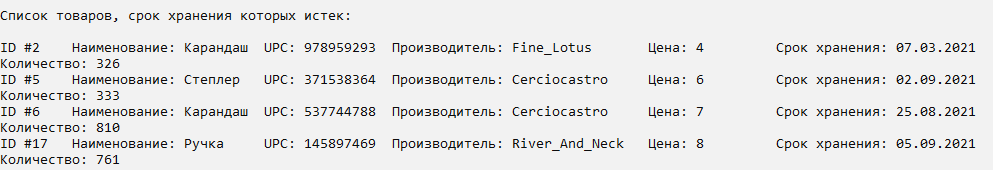
}

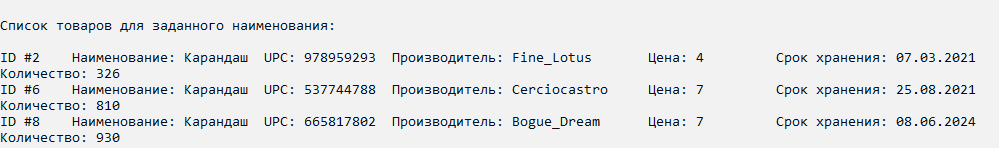
}

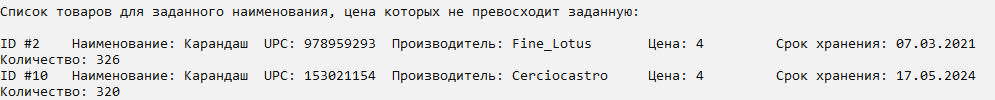
}

**Результат работы программы:**

****

****

****

****