МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №3

По дисциплине «Современные платформы программирования»

Специальность ПО-8

Выполнил:

Серко А.С.

студент группы ПО-8

Проверил:

ст. преп. кафедры ИИТ,

«\_\_»\_\_\_\_\_\_\_\_\_\_\_2024 г.

**Цель работы:** научиться создавать и использовать классы в программах на языке программирования С#

**Вариант 19**

**Задание 1. Множество целых чисел переменной мощности – Предусмотреть возможность пересечения двух множеств, вывода на печать элементов множества, а также метод, определяющий, принадлежит ли указанное значение множеству. Класс должен содержать методы, позволяющие добавлять и удалять элемент в/из множества. Конструктор должен позволить создавать объекты с начальной инициализацией. Реализацию множества осуществить на базе структуры ArrayList. Реализовать метод equals, выполняющий сравнение объектов данного**

**типа.**

Выполнение

**Код программы**

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using static System.Runtime.InteropServices.JavaScript.JSType;

namespace Lab3

{

internal class MySet

{

private List<float> \_list;

public string name;

public List<float> List

{

set { \_list = value; }

get { return \_list; }

}

public MySet(string name, bool initialize=false)

{

this.name = name;

\_list = new List<float>();

if (initialize)

{

\_list.Add(1.0f);

\_list.Add(2.0f);

\_list.Add(3.0f);

}

}

public void AddToList(float value)

{

\_list.Add(value);

}

public void PrintValues()

{

Console.WriteLine($"{this.name} elements:");

foreach (var item in \_list)

{

Console.WriteLine(item);

}

}

public void Remove(float number)

{

\_list.Remove(number);

Console.WriteLine($"{number} was removed from {this.name}");

}

public bool Contains(float number) {

if (!\_list.Contains(number))

{

return false;

}

return true;

}

public void Intersection(MySet obj)

{

Console.WriteLine($"What {this.name}'s elements intersect with {obj.name}?");

List<float> intersection = new List<float>();

foreach (var item in \_list)

{

if (obj.Contains(item))

{

intersection.Add(item);

}

}

if (intersection.Count == 0)

{

Console.WriteLine("No common items");

}

else

{

foreach (var item in intersection)

{

Console.WriteLine(item);

}

}

}

public bool Equals(MySet obj)

{

Console.WriteLine($"Does {this.name} equal {obj.name}?");

if (\_list.Count != obj.\_list.Count) return false;

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

{

if (\_list[i] != obj.\_list[i])

{

return false;

}

}

return true;

}

}

}

using Lab3;

MySet set1 = new MySet("Set1");

MySet set2 = new MySet("Set2",true);

set1.AddToList(3.2f);

set1.AddToList(1.2f);

set1.AddToList(3.8f);

set1.AddToList(4.2f);

set1.AddToList(5.6f);

set1.PrintValues();

set2.AddToList(4.2f);

set2.AddToList(3.8f);

set2.AddToList(3.9f);

set2.PrintValues();

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

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

set1.Intersection(set2);

Console.WriteLine($"Does {set1.name} contain 1.2?");

Console.WriteLine(set1.Contains(1.2f));

Console.WriteLine($"Does {set2.name} contain 1.0?");

Console.WriteLine(set2.Contains(1.0f));

set2.Remove(1.0f);

Console.WriteLine($"Does {set2.name} contain 1.0?");

Console.WriteLine(set2.Contains(1.0f));

**Спецификация ввода**

Данные задаются программно  
MySet set1 = new MySet("Set1");

MySet set2 = new MySet("Set2",true);

set1.AddToList(3.2f);

set1.AddToList(1.2f);

set1.AddToList(3.8f);

set1.AddToList(4.2f);

set1.AddToList(5.6f);

set1.PrintValues();

set2.AddToList(4.2f);

set2.AddToList(3.8f);

set2.AddToList(3.9f);

**Спецификация вывода**

Set1 elements:

<эл-ты 1 множества>

Set2 elements:

<эл-ты 2 множества>

Does Set1 equal Set1?

True или False

Does Set1 equal Set2?

True или False

What Set1's elements intersect with Set2?

<эл-ты пересечения 2-х множеств>

Does Set1 contain 1.2?

True или False

Does Set2 contain 1.0?

True или False

<число (элемент) was removed from <название множества>

Does Set2 contain 1.0?

True или False

**Пример**

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

**Задание 2. Написать программу для моделирования автоматизированного склада. На складе хранится различная продукция (Product) Каждая родукция характеризуется следующей информацией:**

• id;

• Наименование;

• UPC (штрих-код);

• Производитель;

• Цена;

• Срок хранения;

• Количество.

Программа должна иметь следующий функционал:

• Генерация списка продукции на складе;

• Предоставлять список товаров для заданного наименования;

• Предоставлять список товаров для заданного наименования, цена которых не превосходит

заданную;

• Предоставлять список товаров, срок хранения которых истек**.**

Выполнение

**Код программы**

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Task2

{

internal class Product

{

public int Id { get; set; }

public string Name { get; set; }

public string UPC { get; set; }

public string Manufacturer { get; set; }

public double Price { get; set; }

public DateTime ExpiryDate { get; set; }

public int Quantity { get; set; }

}

internal class Store

{

private List<Product> products = new List<Product>();

public Store()

{

products = new List<Product>();

}

public void AddProduct(Product product)

{

products.Add(product);

}

public void GenerateProductsFromDataFile(string filename)

{

StreamReader file = new StreamReader(filename);

string line;

while ((line = file.ReadLine()) != null)

{

string[] data = line.Split(";");

Product product = new Product

{

Id = Convert.ToInt32(data[0]),

Name = data[1],

UPC = data[2],

Manufacturer = data[3],

Price = Convert.ToDouble(data[4]),

ExpiryDate = Convert.ToDateTime(data[5]),

Quantity = Convert.ToInt32(data[6])

};

products.Add(product);

}

file.Close();

}

public List<Product> GetProductsByName(string name)

{

Console.WriteLine($"Products with name {name}");

return products.Where(p => p.Name == name).ToList();

}

public void PrintProducts(List<Product> productList)

{

foreach (var item in productList)

{

Console.WriteLine($"Product Data:{item.Id} {item.UPC} {item.Name} {item.Manufacturer} {item.Price} {item.Quantity} {item.ExpiryDate}");

}

}

public List<Product> GetProductsByNameAndPrice(string name, double maxPrice)

{

Console.WriteLine($"Products with name {name} and price<{maxPrice}");

return products.Where(p => p.Name == name && p.Price <= maxPrice).ToList();

}

public List<Product> GetExpiredProducts()

{

Console.WriteLine("Expired Products");

return products.Where(p => p.ExpiryDate < DateTime.Today).ToList();

}

}

}

**Спецификация ввода**

**products.txt**

Id;name;upc;manufacturer;price;expiryDate;quantity

**Пример**

1;Apple;123456789012;Apple Inc.;1,50;29/01/2024;100

2;Banana;234567890123;Chiquita Brands International;0,75;28/01/2024;150

3;Milk;345678901234;Dairy Farmers of America;2,00;04/02/2024;50

4;Bread;456789012345;Wonder Bread Company;1,25;02/02/2024;75

5;Eggs;567890123456;Happy Egg Co.;3,00;19/02/2024;30

6;Apple;123456789017;Tasty Farm.;2,80;18/01/2024;40

**Спецификация вывода**

Product Data:<id> <upc> <name> <manufacturer> <price> <quantity> <expiryDate>

**Пример**

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**Вывод**

Научились создавать и использовать классы в программах на языке программирования С#.