

КАМ'ЯНЕЦЬ-ПОДІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ІВАНА ОГІЄНКА

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

**ЛАБОРАТОРНА РОБОТА №6**  
З ДИСЦИПЛІНИ «ОБ'ЄКТНО-ОРІЄНТОВАНЕ  
ПРОГРАМУВАННЯ»

Класи колекцій і протоколи ітерації

**Виконав:**

студент 1-го курсу

гр. KNms1-B21

Шевчук О. В.

**Прийняв:**

доцент,

Іванюк В. А.

Дата здачі «\_\_»\_\_\_\_\_2021 р

## ЗАВДАННЯ 1-4

### Код програми:

```
public interface ISignal
{
    int year { get; set; }
    string name { get; set; }
    int diameter { get; set; }
    int frequency { get; set; }
}

class Signal : ISignal, IComparable<Signal>
{
    public int year { get; set; }
    public string name { get; set; }
    public int diameter { get; set; }
    public int frequency { get; set; }

    public Signal(int year, string name, int diameter, int frequency)
    {
        this.year = year;
        this.name = name;
        this.diameter = diameter;
        this.frequency = frequency;
    }

    public string Info
    {
        get { return $"{year} {name}"; }
    }

    public int CompareTo(Signal other)
    {
        return string.Compare(other.Info, Info,
StringComparison.InvariantCultureIgnoreCase);
    }

    public override string ToString()
    {
        return String.Format("{0,20}|{1,20}|{2,20}|{3,20}", year, name, diameter,
frequency);
    }
}

class CollectionType<T> : IEnumerable<T> where T : Signal
{
    List<T> list = new List<T>();

    public CollectionType()
    {
        list = new List<T>();
    }

    public int Count
    {
        get { return list.Count; }
    }

    public T this[int index]
    {
        get
        {
            if (index < 0 || index >= Count)

```

```

        {
            throw new IndexOutOfRangeException();
        }
        return list[index];
    }
    set
    {
        if (index < 0 || index >= Count)
        {
            throw new IndexOutOfRangeException();
        }
        list[index] = value;
    }
}

public void Add(T signal)
{
    list.Add(signal);
}

public T Remove(T signal)
{
    var element = list.FirstOrDefault(h => h == signal);
    if (element != null)
    {
        list.Remove(element);
        return element;
    }
    throw new NullReferenceException();
}

public void Sort()
{
    list.Sort();
}

public T GetByName(string name)
{
    return
        list.FirstOrDefault(
            h => string.Compare(h.Info, name,
StringComparison.InvariantCultureIgnoreCase) == 0);
}

public IEnumerator<T> GetEnumerator()
{
    return list.GetEnumerator();
}

IEnumerator IEnumerable.GetEnumerator()
{
    return GetEnumerator();
}
}

class Program
{
    static void Main(string[] args)
    {
        Signal sgn1 = new Signal(1960, "Дрейк", 26, 1420);
        Signal sgn2 = new Signal(1970, "Троїцький", 14, 1875);
        Signal sgn3 = new Signal(1978, "Хоровіц", 300, 1665);
        Signal sgn4 = new Signal(1981, "Барановський", 180, 1774);
    }
}

```

```

CollectionType<Signal> collection = new CollectionType<Signal>();
collection.Add(sgn1);
collection.Add(sgn2);
collection.Add(sgn3);
collection.Add(sgn4);
collection.Remove(sgn4);

Console.WriteLine(String.Format("{0,20}|{1,20}|{2,20}|{3,20}", "Pіk ",
"Науковий керівник ", "Діаметр антени (м) ", "Робоча частота (МГц)"));
foreach (Signal s in collection)
{
    Console.WriteLine(s.ToString());
}

Signal sgn5 = new Signal(1980, "Саржков", 45, 1256);
Signal sgn6 = new Signal(1983, "Лузкович", 155, 1733);
Signal sgn7 = new Signal(1988, "Прокопенко", 288, 1888);

CollectionType<Signal> collection2 = new CollectionType<Signal>();
collection.Add(sgn5);
collection.Add(sgn6);
collection.Add(sgn7);

var list = new List<CollectionType<Signal>>();
list.Add(collection);
list.Add(collection2);

Console.WriteLine("\n                OrderBy:");
var order = collection.OrderBy(h => h.diameter).ThenBy(h => h.year);
foreach (var signal in order)
{
    Console.WriteLine(signal);
}

Console.WriteLine("\n                where:");
var where = collection.Where(h => (h.diameter >= 100 && h.frequency > 1450)
|| h.Info.StartsWith("L"));
foreach (var signal in where)
{
    Console.WriteLine(signal.ToString());
}

Console.WriteLine("\n                Select:");
var select = collection.Select((h, i) => new { Index = i + 1, h.Info });
foreach (var s in select)
{
    Console.WriteLine(s);
}

Console.WriteLine("\n                Skip:");
var skip = collection.Skip(3);
foreach (var signal in skip)
{
    Console.WriteLine(signal);
}

Console.WriteLine("\n                Take:");
var take = collection.Take(3);
foreach (var signal in take)
{
    Console.WriteLine(signal);
}

Console.WriteLine("\n                Concat:");

```

```

var concat = collection.Concat(collection2);
foreach (var signal in concat)
{
    Console.WriteLine(signal);
}

Console.WriteLine("\n                First:");
var first = collection.First(h => h.Info.Length > 5);
Console.WriteLine(first);

Console.WriteLine("\n                Min: ");
var min = collection.Min(h => h.frequency);
Console.WriteLine(min);

Console.WriteLine("\n                Max: ");
var max = collection.Max(h => h.frequency);
Console.WriteLine(max);

Console.WriteLine("\nAll and Any:");
var allAny = list.First(c => c.All(h => h.diameter >= 14) && c.Any(h => h
is Signal)).Select(h => h.Info).OrderByDescending(s => s);
foreach (var str in allAny)
{
    Console.WriteLine(str);
}

Console.WriteLine("\nContains:");
var contains = list.Where(c => c.Contains(sgn1)).SelectMany(c =>
c.SelectMany(h => h.Info.Split(' '))).Distinct().OrderBy(s => s).ToList();
foreach (var str in contains)
{
    Console.WriteLine(str);
}
}
}

```

## Робота програми:

Рік	Науковий керівник	Діаметр антени (м)	Робоча частота (МГц)
1960	Дрейк	26	1420
1970	Троїцкий	14	1875
1978	Хоровіц	300	1665
OrderBy:			
1970	Троїцкий	14	1875
1960	Дрейк	26	1420
1980	Саржков	45	1256
1983	Лузкович	155	1733
1988	Прокопенко	288	1888
1978	Хоровіц	300	1665
where:			
1978	Хоровіц	300	1665
1983	Лузкович	155	1733
1988	Прокопенко	288	1888
Select:			
{ Index = 1, Info = 1960 Дрейк }			
{ Index = 2, Info = 1970 Троїцкий }			
{ Index = 3, Info = 1978 Хоровіц }			
{ Index = 4, Info = 1980 Саржков }			
{ Index = 5, Info = 1983 Лузкович }			

Skip:			
1980	Саржков	45	1256
1983	Лузкович	155	1733
1988	Прокопенко	288	1888
Take:			
1960	Дрейк	26	1420
1970	Троїцкий	14	1875
1978	Хоровіц	300	1665
Concat:			
1960	Дрейк	26	1420
1970	Троїцкий	14	1875
1978	Хоровіц	300	1665
1980	Саржков	45	1256
1983	Лузкович	155	1733
1988	Прокопенко	288	1888
First:			
1960	Дрейк	26	1420
Min:	1256		
Max:	1888		

All and Any:  
 1988 Прокопенко  
 1983 Лузкович  
 1980 Саржков  
 1978 Хоровіц  
 1970 Троїцкий  
 1960 Дрейк

Contains:  
 1960  
 1970  
 1978  
 1980  
 1983  
 1988  
 Дрейк  
 Лузкович  
 Прокопенко  
 Саржков  
 Троїцкий  
 Хоровіц

## ЗАВДАННЯ 5

### Код програми:

```
class CollectionType
{
    LinkedList<CollectionType> link = new LinkedList<CollectionType>();
    int year { get; set; }
    string name { get; set; }
    int diameter { get; set; }
    int frequency { get; set; }

    public CollectionType() {}

    public CollectionType(int year, string name, int diameter, int frequency)
    {
        this.year = year;
        this.name = name;
        this.diameter = diameter;
        this.frequency = frequency;
    }

    public void Add(CollectionType[] coll)
    {
```

```

        for (int i = 0; i < coll.Length; i++)
        {
            link.AddLast(coll[i]);
        }
    }

    public void Output()
    {
        Console.WriteLine(String.Format("{0,20}|{1,20}|{2,20}|{3,20}", "Рік ",
"Науковий керівник ", "Діаметр антени (м) ", "Робоча частота (МГц)"));
        foreach (CollectionType s in link)
        {
            Console.WriteLine(String.Format("{0,20}|{1,20}|{2,20}|{3,20}", s.year,
s.name, s.diameter, s.frequency));
        }
    }

    public void Select()
    {
        Console.WriteLine("\n                Запит 1:");
        var where = link.Where(h => (h.diameter >= 100 && h.frequency > 1450));
        Console.WriteLine(String.Format("{0,20}|{1,20}|{2,20}|{3,20}", "Рік ",
"Науковий керівник ", "Діаметр антени (м) ", "Робоча частота (МГц)"));
        foreach (var c in where)
        {
            Console.WriteLine(String.Format("{0,20}|{1,20}|{2,20}|{3,20}", c.year,
c.name, c.diameter, c.frequency));
        }
        Console.WriteLine("\n                Запит 2:");
        var min = link.Min(h => h.frequency);
        Console.WriteLine($"                {min}");
        Console.WriteLine("\n                Запит 3:");
        var max = link.Max(h => h.frequency);
        Console.WriteLine($"                {max}");
        Console.WriteLine("\n                Запит 4:");
        var count = link.Count();
        Console.WriteLine($"                {count}");
        Console.WriteLine("\n                Запит 5:");
        var order = link.OrderBy(h => h.diameter).ThenByDescending(h => h.year);
        Console.WriteLine(String.Format("{0,20}|{1,20}|{2,20}|{3,20}", "Рік ",
"Науковий керівник ", "Діаметр антени (м) ", "Робоча частота (МГц)"));
        foreach (var c in order)
        {
            Console.WriteLine(String.Format("{0,20}|{1,20}|{2,20}|{3,20}", c.year,
c.name, c.diameter, c.frequency));
        }
    }
}

class Program
{
    static void Main(string[] args)
    {
        CollectionType ct = new CollectionType();

        CollectionType[] collections = {
            new CollectionType(1960, "Дрейк", 26, 1420),
            new CollectionType(1970, "Троїцький", 14, 1875),
            new CollectionType(1978, "Хоровіц", 300, 1665)
        };

        ct.Add(collections);
        ct.Output();
        ct.Select();
    }
}

```

}  
}

**Робота програми:**

Рік	Науковий керівник	Діаметр антени (м)	Робоча частота (МГц)
1960	Дрейк	26	1420
1970	Троїцкий	14	1875
1978	Хоровіц	300	1665
Запит 1:			
Рік	Науковий керівник	Діаметр антени (м)	Робоча частота (МГц)
1978	Хоровіц	300	1665
Запит 2:			
1420			
Запит 3:			
1875			
Запит 4:			
3			
Запит 5:			
Рік	Науковий керівник	Діаметр антени (м)	Робоча частота (МГц)
1970	Троїцкий	14	1875
1960	Дрейк	26	1420
1978	Хоровіц	300	1665