

Лабораторна робота № 6

ПОНЯТТЯ ДЕЛЕГАТИВ. СТВОРЕННЯ ТА ВИКОРИСТАННЯ ДЕЛЕГАТИВ. ПРИНЦИПИ РОБОТИ ПОДІЙ. ОТРИМАННЯ ПОДІЙ.

Мета роботи – ознайомитися з поняттям делегатів та подій в C#. Навчитися використовувати делегати та події в програмних додатках.

Варіант 15

Завдання до лабораторної роботи

Завдання 1. Створіть набір методів для роботи з масивами. Використовуйте механізми делегатів.

Варіант 15. Метод для отримання усіх чисел в масиві, які є ступенями числа 2; Метод для отримання усіх чисел двозначних чисел в масиві.

```
using System;
```

```
public class Program
```

```
{
```

```
    public delegate bool CheckCondition(int number);
```

```
    public static void Main(string[] args)
```

```
    {
```

```
        int[] numbers = { 2, 3, 4, 8, 16, 17, 32, 64, 100 };
```

```
        int[] powersOfTwo = FilterArray(numbers, IsPowerOfTwo);
```

```
        Console.WriteLine("Числа, які є ступенями числа 2:");
```

```
        foreach (int num in powersOfTwo)
```

```
        {
```

```
            Console.WriteLine(num);
```

```
        }
```

```
int[] twoDigitNumbers = FilterArray(numbers, IsTwoDigitNumber);  
Console.WriteLine("\nДвозначні числа в масиві:");  
foreach (int num in twoDigitNumbers)  
{  
    Console.WriteLine(num);  
}  
}
```

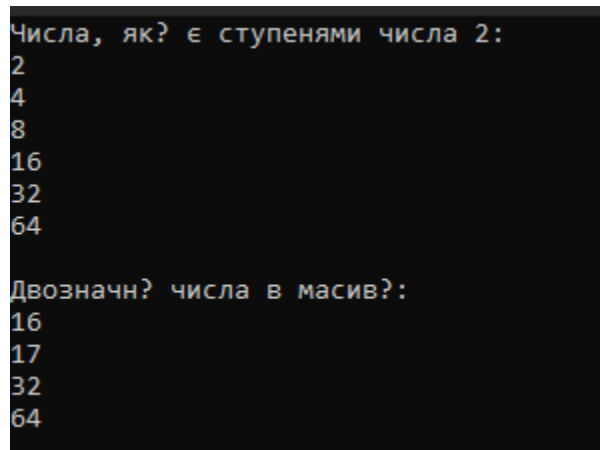
```
public static int[] FilterArray(int[] numbers, CheckCondition condition)  
{  
    int[] result = new int[numbers.Length];  
    int index = 0;  
    foreach (int num in numbers)  
    {  
        if (condition(num))  
        {  
            result[index++] = num;  
        }  
    }  
    Array.Resize(ref result, index);  
    return result;  
}
```

```
public static bool IsPowerOfTwo(int number)  
{  
    return (number != 0) && ((number & (number - 1)) == 0);  
}
```

```

public static bool IsTwoDigitNumber(int number)
{
    return number >= 10 && number <= 99;
}
}

```



```

Числа, як? є ступенями числа 2:
2
4
8
16
32
64

Двозначн? числа в масив?:
16
17
32
64

```

Рис. 1 – Результат виконання завдання 1

Завдання 2. Створіть анонімний делегат або лямбда-вираз для:
Варіант 15. перевірки, скільки днів у поточному місяці.

```

using System;

```

```

public class Program

```

```

{

```

```

    public static void Main(string[] args)

```

```

    {

```

```

        Func<int>    daysInCurrentMonth    =    ()    =>

```

```

        DateTime.DaysInMonth(DateTime.Now.Year, DateTime.Now.Month);

```

```

int days = daysInCurrentMonth();
Console.WriteLine($"У поточному місяці {days} днів.");
}
}

```

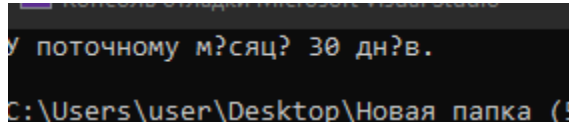


Рис. 2 – Результат виконання завдання 2

Завдання 3 (Непарні варіанти).

Створити клас «Рюкзак». Характеристики рюкзака:

- Колір рюкзака;
- Марка і виробник;
- Вага рюкзака;
- Об'єм рюкзака;
- Вміст рюкзака (список об'єктів у кожного об'єкта, крім назви, потрібно враховувати зайнятий об'єм).

Створіть методи для заповнення характеристик. Створіть події для додавання об'єкту в рюкзак, видалення об'єкту з рюкзака, перегляду вмісту рюкзака, визначення вільного об'єму рюкзака. Якщо при спробі додавання об'єкту може бути перевищено обсяг рюкзака, потрібно генерувати виняток.

Код із файлу Item.cs:

```

public class Item
{
    public string Name { get; set; }
    public double Volume { get; set; }
}

```

```
}
```

Код із файлу ItemEventArgs.cs:

```
using System;

public class ItemEventArgs : EventArgs
{
    public Item Item { get; }

    public ItemEventArgs(Item item)
    {
        Item = item;
    }
}
```

Код із файлу Backpack.cs:

```
using System;
using System.Collections.Generic;

public class Backpack
{
    public event EventHandler<ItemEventArgs> ItemAdded;
    public event EventHandler<ItemEventArgs> ItemRemoved;

    private string color;
    private string brand;
```

```
private double weight;  
private double volume;  
private List<Item> contents;
```

```
public Backpack(string color, string brand, double weight, double volume)  
{  
    this.color = color;  
    this.brand = brand;  
    this.weight = weight;  
    this.volume = volume;  
    this.contents = new List<Item>();  
}
```

```
public void AddItem(Item item)  
{  
    if (GetFreeVolume() < item.Volume)  
    {  
        throw new InvalidOperationException("Рюкзак переполнен!");  
    }  
  
    contents.Add(item);  
    OnItemAdded(item);  
}
```

```
public void RemoveItem(Item item)  
{  
    contents.Remove(item);  
    OnItemRemoved(item);  
}
```

```
}
```

```
public void ViewContents()
```

```
{
```

```
    Console.WriteLine("Вміст рюкзака:");
```

```
    foreach (var item in contents)
```

```
    {
```

```
        Console.WriteLine($"{item.Name}, Об'єм: {item.Volume}");
```

```
    }
```

```
}
```

```
public double GetFreeVolume()
```

```
{
```

```
    double occupiedVolume = 0;
```

```
    foreach (var item in contents)
```

```
    {
```

```
        occupiedVolume += item.Volume;
```

```
    }
```

```
    return volume - occupiedVolume;
```

```
}
```

```
protected virtual void OnItemAdded(Item item)
```

```
{
```

```
    ItemAdded?.Invoke(this, new ItemEventArgs(item));
```

```
}
```

```
protected virtual void OnItemRemoved(Item item)
```

```
{
```

```
ItemRemoved?.Invoke(this, new ItemEventArgs(item));  
}  
}
```

Код із файлу Program.cs:

```
using System;  
  
class Program  
{  
    static void Main(string[] args)  
    {  
        Backpack backpack = new Backpack("Чорний", "Nike", 1.5, 20);  
  
        backpack.ItemAdded += Backpack_ItemAdded;  
        backpack.ItemRemoved += Backpack_ItemRemoved;  
  
        backpack.AddItem(new Item { Name = "Книга", Volume = 5 });  
        backpack.AddItem(new Item { Name = "Ланчбокс", Volume = 3 });  
        backpack.AddItem(new Item { Name = "Вода", Volume = 2 });  
  
        backpack.ViewContents();  
  
        backpack.RemoveItem(new Item { Name = "Ланчбокс", Volume = 3 });  
  
        backpack.ViewContents();  
    }  
}
```

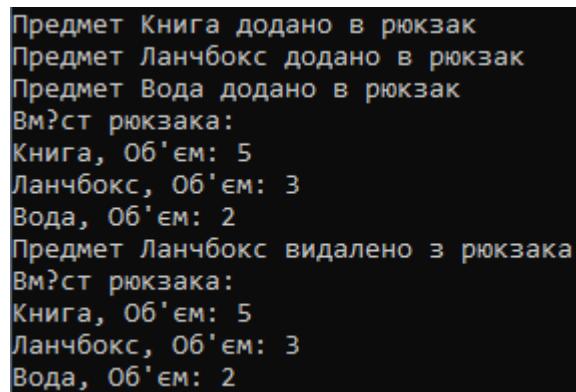


```

private static void Backpack_ItemAdded(object sender, ItemEventArgs e)
{
    Console.WriteLine($"Предмет {e.Item.Name} додано в рюкзак");
}

private static void Backpack_ItemRemoved(object sender, ItemEventArgs e)
{
    Console.WriteLine($"Предмет {e.Item.Name} видалено з рюкзака");
}
}

```



```

Предмет Книга додано в рюкзак
Предмет Ланчбокс додано в рюкзак
Предмет Вода додано в рюкзак
Вміст рюкзака:
Книга, Об'єм: 5
Ланчбокс, Об'єм: 3
Вода, Об'єм: 2
Предмет Ланчбокс видалено з рюкзака
Вміст рюкзака:
Книга, Об'єм: 5
Ланчбокс, Об'єм: 3
Вода, Об'єм: 2

```

Рис. 3 – Результат виконання завдання 3

Контрольні запитання

1. Що таке делегат в C#?

В C# делегат - це тип, який представляє посилання на метод. Він дозволяє передавати метод як параметр із одного місця коду в інше, що дозволяє викликати методи динамічно.

2. Нащадками яких класів є делегати?

В C# делегати є нащадками класу System.Delegate.

3. Синтаксис оголошення делегатів C#.

`delegate <returnType> <DelegateName>(`

4. Які типи методів можна присвоювати делегатам?

Делегати можуть посилатися на методи, які відповідають їх сигнатурі. Це можуть бути методи зі збігаючою сигнатурою та поверненням значенням, що вказується в делегаті.

5. Що означає термін – багатоадресні делегати?

Багатоадресні делегати (multicast delegates) - це делегати, які можуть утримувати посилання на більше ніж один метод. Коли багатоадресний делегат викликається, всі методи, що утримуються в ньому, викликаються послідовно.

6. Які операції допустимі для багатоадресних делегатів?

Для багатоадресних делегатів допустимі операції додавання (+=) і видалення (-=) методів зі списку методів, які утримуються в делегаті.

7. Що таке анонімні методи?

Анонімний метод - це безіменний блок коду, який може бути визначений та використаний в тому ж контексті, де його створено. Вони часто використовуються як аргументи для методів, таких як `delegate` або підписки на події.

8. Що таке події в C#?

Події в C# - це механізм, який дозволяє об'єктам класу сповіщати інші об'єкти про те, що відбулася певна подія.

9. Поясніть значення термінів видавець та передплатник відносно подій?

В контексті подій, видавець - це об'єкт, який сповіщає про виникнення події. Підписник (або передплатник) - це об'єкт, який підписується на подію і очікує сповіщення від видавця, коли подія відбувається.

10. Синтаксис оголошення подій.

`public event EventHandler MyEvent;`

11. Які основні властивості подій?

- `add` - додає делегат до списку делегатів, які відповідають за подію.

- remove - видаляє делегат зі списку делегатів, які відповідають за подію.
- invoke - викликає всі делегати в списку делегатів, які відповідають за подію.