

Практична 5.

Завдання 1 (знову)

```
using System;

using System.Collections.Generic;


// Interface for Spells

interface ISpell

{

    void Cast(Mage caster, Mage target);

}


// Delegate for mage attack and defense events

delegate void MageEventHandler(string message);


// Abstract class Mage

abstract class Mage

{

    public string Name { get; private set; }

    public int Level { get; private set; }

    public int Health { get; private set; }

    public List<ISpell> Spells { get; private set; }
```

```
public event MageEventHandler OnAttack;  
public event MageEventHandler OnDefend;
```

```
protected Mage(string name, int level)  
{  
    Name = name;  
    Level = level;  
    Health = 100;  
    Spells = new List<ISpell>();  
}
```

```
public abstract void Attack(Mage target);  
public abstract void Defend(int damage);
```

```
public void AdjustHealth(int amount)  
{  
    Health += amount;  
    if (Health < 0)  
        Health = 0;  
}
```

```
protected virtual void RaiseAttackEvent(string message)
{
    OnAttack?.Invoke(message);
}
```

```
protected virtual void RaiseDefendEvent(string message)
{
    OnDefend?.Invoke(message);
}
```

```
public bool IsAlive()
{
    return Health > 0;
}
}
```

```
// Fire Mage class
```

```
class FireMage : Mage
{
    public FireMage(string name, int level) : base(name, level)
    {
        Spells.Add(new Fireball());
    }
}
```

```
    Spells.Add(new FlameShield());  
}
```

```
public override void Attack(Mage target)  
{  
    RaiseAttackEvent($"{Name} attacks {target.Name} with a fire spell!");  
    Spells[0].Cast(this, target); // Using Fireball as attack  
}
```

```
public override void Defend(int damage)  
{  
    RaiseDefendEvent($"{Name} defends with a flame shield!");  
    Spells[1].Cast(this, this); // Using FlameShield as defense  
    AdjustHealth(-damage);  
    Console.WriteLine($"{Name} has {Health} health left.");  
}  
}
```

```
// Water Mage class
```

```
class WaterMage : Mage
```

```
{  
    public WaterMage(string name, int level) : base(name, level)
```

```

{
    Spells.Add(new WaterBlast());
    Spells.Add(new WaterShield());
}

public override void Attack(Mage target)
{
    RaiseAttackEvent($"{Name} attacks {target.Name} with a water spell!");
    Spells[0].Cast(this, target); // Using WaterBlast as attack
}

public override void Defend(int damage)
{
    RaiseDefendEvent($"{Name} defends with a water shield!");
    Spells[1].Cast(this, this); // Using WaterShield as defense
    AdjustHealth(-damage);
    Console.WriteLine($"{Name} has {Health} health left.");
}
}

// Spells
class Fireball : ISpell

```

```
{  
    public void Cast(Mage caster, Mage target)  
    {  
        int damage = 20;  
        Console.WriteLine($"Fireball hits {target.Name} for {damage} damage!");  
        target.AdjustHealth(-damage);  
    }  
}
```

```
class FlameShield : ISpell
```

```
{  
    public void Cast(Mage caster, Mage target)  
    {  
        int reduction = 10;  
        Console.WriteLine($"Flame shield reduces damage by {reduction}.");  
        caster.AdjustHealth(reduction);  
    }  
}
```

```
class WaterBlast : ISpell
```

```
{  
    public void Cast(Mage caster, Mage target)
```

```
{  
    int damage = 15;  
    Console.WriteLine($"WaterBlast hits {target.Name} for {damage} damage!");  
    target.AdjustHealth(-damage);  
}  
}
```

```
class WaterShield : ISpell
```

```
{  
    public void Cast(Mage caster, Mage target)  
    {  
        int reduction = 15;  
        Console.WriteLine($"Water shield reduces damage by {reduction}.");  
        caster.AdjustHealth(reduction);  
    }  
}
```

```
// Main game class
```

```
class Game
```

```
{  
    public void Start()  
    {
```

```
Console.WriteLine("Welcome to the Battle of Mages!");
```

```
Mage mage1 = ChooseMage("Player 1");
```

```
Mage mage2 = ChooseMage("Player 2");
```

```
mage1.OnAttack += DisplayEvent;
```

```
mage1.OnDefend += DisplayEvent;
```

```
mage2.OnAttack += DisplayEvent;
```

```
mage2.OnDefend += DisplayEvent;
```

```
Battle(mage1, mage2);
```

```
}
```

```
private Mage ChooseMage(string player)
```

```
{
```

```
    Console.WriteLine($"{player}, choose your mage (1. Fire, 2. Water): ");
```

```
    int choice = int.Parse(Console.ReadLine());
```

```
    Console.WriteLine("Enter your mage's name: ");
```

```
    string name = Console.ReadLine();
```

```
    return choice switch
```



```
{  
    1 => new FireMage(name, 1),  
    2 => new WaterMage(name, 1),  
    _ => throw new InvalidOperationException("Invalid choice")  
};  
}
```

```
private void Battle(Mage mage1, Mage mage2)
```

```
{  
    while (mage1.IsAlive() && mage2.IsAlive())  
    {  
        mage1.Attack(mage2);  
        if (mage2.IsAlive())  
        {  
            mage2.Attack(mage1);  
        }  
    }  
}
```

```
if (mage1.IsAlive())  
{  
    Console.WriteLine($"{mage1.Name} wins!");  
}
```

```
        else
        {
            Console.WriteLine($"{mage2.Name} wins!");
        }
    }

    private void DisplayEvent(string message)
    {
        Console.WriteLine(message);
    }
}

// Main Program
class Program
{
    static void Main()
    {
        Game game = new Game();
        game.Start();
    }
}
```

```

- === Code Execution Successful ===

```