Marker som gennemført

Åbnet: fredag den 23. februar 2024, 13:00

Sofie har lavet noget rod, hun har forsøgt at lave et interface og en række spilkarakter. Hun overholder dog ikke interface segregation princippet, så du er nødt til at hjælpe med at få styr på det hele.

Aflever din kode enten via link til visual studio online eller som zippet filer her på Praxis.

```
public interface ICharacter
{
  void ThrowMagicMisile();
  void Heal();
  void Die();
  void ThrowFrostNova();
  void RaiseShield();
  void Fight();
  void Teleport(int x, int y);
  void Bash();
  void Cleave();
  void Slash();
  void ShieldGlare();
}
```

?

```
public class Wizard : ICharacter
{
   public void Bash()
   {
      throw new NotImplementedException();
   }

   public void Cleave()
   {
     throw new NotImplementedException();
   }

   public void Die()
   {
     Console.WriteLine("I'm dying");
   }

   public void Fight()
   {
     Console.WriteLine("I'm fighting");
   }

   public void Heal()
   {
     Console.WriteLine("I'm healing");
   }

   public void RaiseShield()
   {
      throw new NotImplementedException();
   }

   public void ShieldGlare()
   {
      throw new NotImplementedException();
   }

   public void Slash()
   {
      throw new NotImplementedException();
   }

   public void Teleport(int x, int y)
   {
      Console.WriteLine("I'm teleporting to "+ x+ " "+y);
   }

   public void ThrowFrostNova()
   {
      Console.WriteLine("I'm throwing my frost nova");
   }

   public void ThrowMagicMisile()
   {
      Console.WriteLine("I'm throwing a magic misile");
   }
}
```

```
public class Babarian : ICharacter
{
    public void Bash()
    {
      Console.WriteLine("I'm bashing someone");
    }

    public void Cleave()
    {
      Console.WriteLine("I'm cleaving someone");
    }

    public void Die()
    {
      Console.WriteLine("I'm dying");
    }

    public void Fight()
    {
      Console.WriteLine("I'm fighting");
    }

    public void Heal()
    {
      Console.WriteLine("I'm healing");
    }

    public void RaiseShield()
    {
      throw new NotImplementedException();
    }

    public void ShieldGlare()
    {
       throw new NotImplementedException();
    }

    public void Slash()
    {
        Console.WriteLine("I'm slashing someone");
    }

    public void Teleport(int x, int y)
     {
        throw new NotImplementedException();
    }

    public void ThrowFrostNova()
    {
        throw new NotImplementedException();
    }

    public void ThrowMagicMisile()
    {
        throw new NotImplementedException();
    }
 }
```

```
public class Knight : ICharacter
{
    public void Bash()
    {
       Console.WriteLine("I'm bashing someone");
    }

    public void Cleave()
    {
       Console.WriteLine("I'm cleaving someone");
    }

    public void Die()
    {
       Console.WriteLine("I'm dying");
    }

    public void Fight()
    {
        Console.WriteLine("I'm fighting");
    }

    public void Heal()
    {
       Console.WriteLine("I'm healing");
    }

    public void RaiseShield()
    {
       Console.WriteLine("I'm raising my shield");
    }

    public void ShieldGlare()
    {
       Console.WriteLine("I'm throwing shield glare");
    }

    public void Slash()
    {
       Console.WriteLine("I'm slashing someone");
    }

    public void Teleport(int x, int y)
    {
        throw new NotImplementedException();
    }

    public void ThrowFrostNova()
    {
        throw new NotImplementedException();
    }

    public void ThrowMagicMisile()
    {
        throw new NotImplementedException();
    }
}
```

```
public class Witch: ICharacter
{
    public void Bash()
    {
       throw new NotImplementedException();
    }

    public void Cleave()
    {
       throw new NotImplementedException();
    }

    public void Die()
    {
       Console.WriteLine("I'm dying");
    }

    public void Fight()
    {
       Console.WriteLine("I'm fighting");
    }

    public void Heal()
    {
        Console.WriteLine("I'm healing");
    }

    public void RaiseShield()
    {
        Console.WriteLine("I'm raising my shield");
    }

    public void ShieldGlare()
    {
        Console.WriteLine("I'm throwing shield glare");
    }

    public void Slash()
    {
        throw new NotImplementedException();
    }

    public void Teleport(int x, int y)
    {
        Console.WriteLine("I'm teleporting to "+ x+ " "+y);
    }

    public void ThrowFrostNova()
    {
        throw new NotImplementedException();
    }

    public void ThrowMagicMisile()
    {
        throw new NotImplementedException();
    }
 }
```

Besvar opgaven

Afleveringsstatus

Antal forsøg på besvarelse	Dette er besvarelsesforsøg 1.
Afleveringsstatus	No submissions have been made yet

Vurderingsstatus	Ikke bedømt

◄ Se videoen : The I in SOLID

Spring til...

Dependency Inversion ►