

Decorator

...

Tim & Jael

Probleem

- We maken een game
- Als developer moeten we de mogelijkheid bieden om armor upgrades te geven.

<i>Armor</i>	<i>Armor</i>
- hasGem	- hasPurpleGem
+ ProtectMe()	- hasBlueGem
	- hasRedGem
	+ ProtectMe()

```
public class Armor
{
    private bool _hasBlueGem = false;
    private bool _hasPurpleGem = false;
    private bool _hasRedGem = false;

    public void SetBlueGem()
    {
        _hasBlueGem = true;
    }

    public void SetPurpleGem()
    {
        _hasPurpleGem = true;
    }

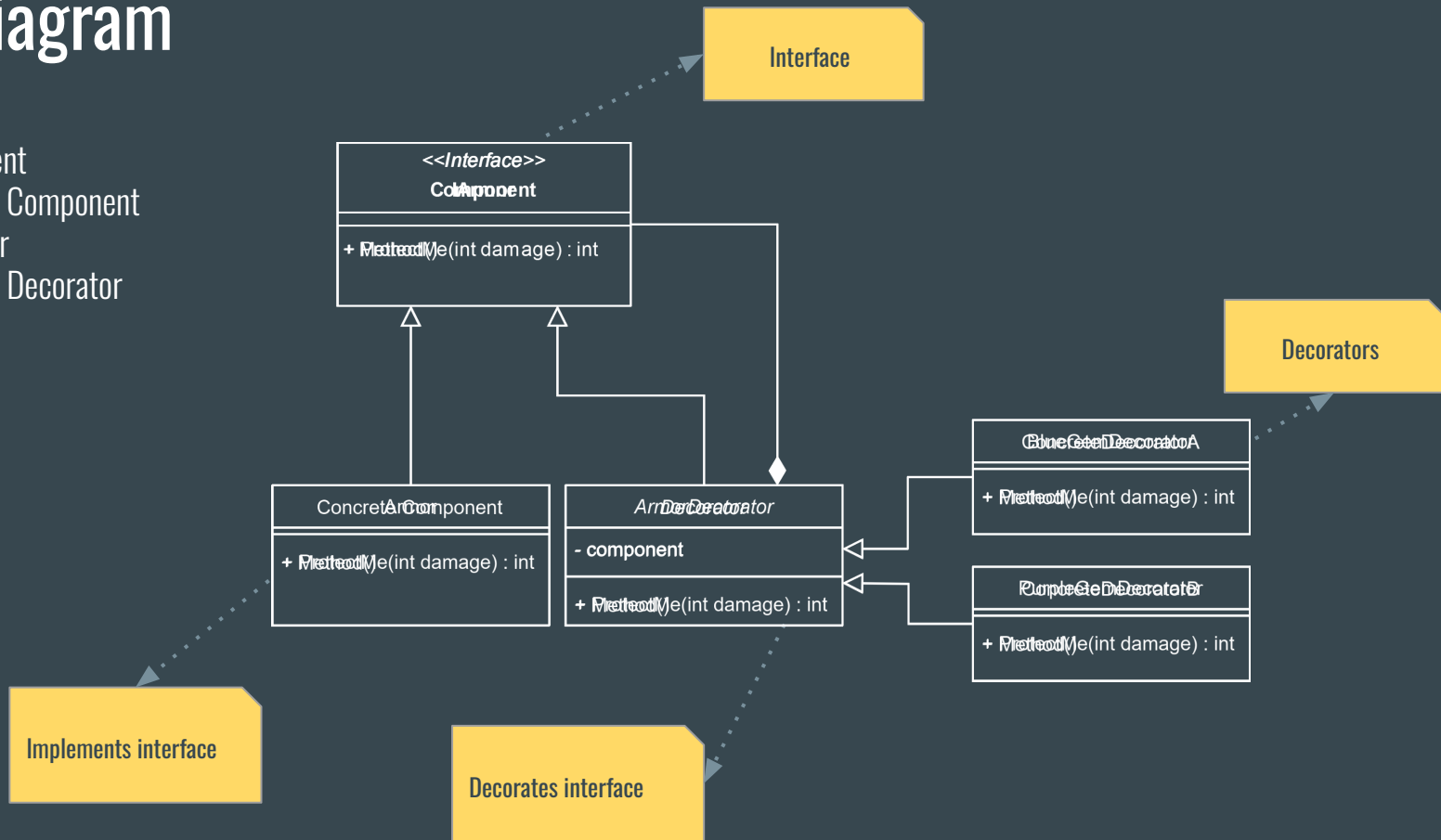
    public void SetRedGem()
    {
        _hasRedGem = true;
    }

    public int ProtectMe(int damage)
    {
        if (_hasBlueGem)
            damage -= 20;
        if (_hasPurpleGem)
            damage -= 20;
        if (_hasRedGem)
            damage -= 20;

        return damage - 10;
    }
}
```

Class diagram

- Component
- Concrete Component
- Decorator
- Concrete Decorator



Component - IArmor

```
public interface IArmor
{
    public int ProtectMe(int damage);
}
```

Concrete component - Armor



```
public interface IArmor
{
    public int ProtectMe(int damage);
}
```

```
public class Armor : IArmor
{
    private readonly int _addedProtection = 10;

    public int ProtectMe(int damage)
    {
        ...

        return damage - _addedProtection;
    }
}
```

Decorator - ArmorDecorator



```
public interface IArmor
{
    public int ProtectMe(int damage);
}
```

```
public abstract class ArmorDecorator : IArmor
{
    protected readonly IArmor _armor;

    public ArmorDecorator(IArmor armor)
    {
        _armor = armor;
    }

    public abstract int ProtectMe(int damage);
}
```

Concrete decorator - RedGemDecorator (etc.)



```
public abstract class ArmorDecorator : IArmor
{
    public class RedGemDecorator : ArmorDecorator
    {
        public readonly int _addedProtection = 8;

        public RedGemDecorator(IArmor armor) : base(armor) { }

        public override int ProtectMe(int damage)
        {
            ...

            return _armor.ProtectMe(damage - _addedProtection);
        }
    }
}
```

Example

```
public static void Main()
{
    var armor = new Armor();

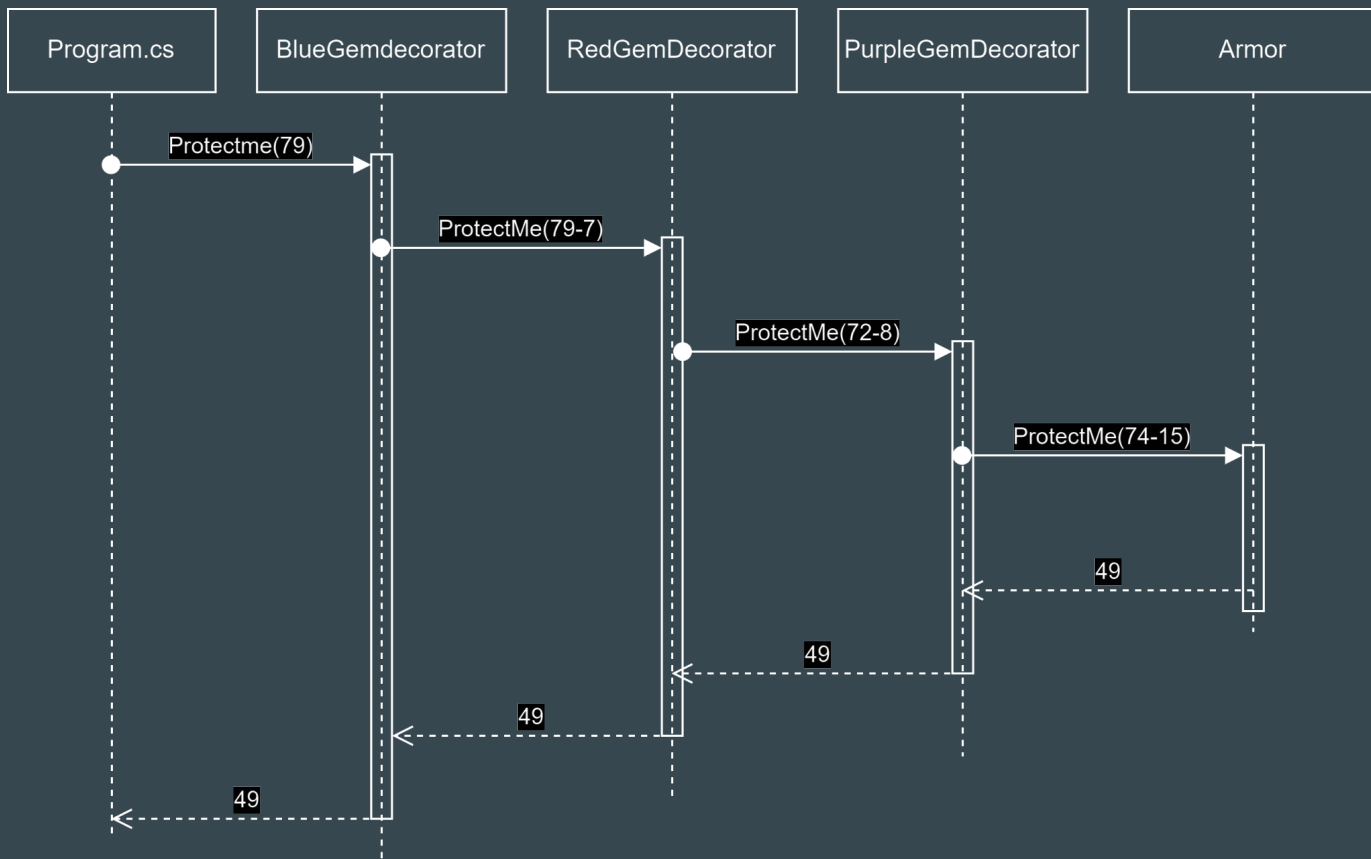
    var armorWithRedGem = new RedGemDecorator(armor);

    var armorWithBlueAndPurpleGem = new BlueGemDecorator(
        new PurpleGemDecorator(armor));

    int damage = 80;

    armor.ProtectMe(damage);
    armorWithRedGem.ProtectMe(damage);
    armorWithBlueAndPurpleGem.ProtectMe(damage);
}
```


Sequence diagram



Vragen?

...