

Kamp Arena (C#)

Det sker ret ofte at man som programmør render ind i mange forskellige udfordringer. Én af disse udfordringer er, at man ikke må ændre på eksisterende kode. Hvis man foretager ændringer kan noget kode muligvis ikke eksekveres eller også kan det ske at der skabes for mange bindinger.



I denne opgave skal du udarbejde en lille "kampsimulation" mellem forskellige typer af spilkarakter.

En ridder har et "attack range" på 1-6, hvor 6 er det højeste. Ridderen har også et forsvar på 30 point og hver gang ridderen angribes trækkes den attack range fra, som han bliver angrebet med. Nogle gange (Der er ca. 15% chance) bliver ridderen rigtig bange, når **han har** forsvaret sig og så løber han sin vej og taber kampen.

1. Implementer en ridder der gør brug af interfacet "Fighter" – du **må ikke** ændre på interfacet.

```
public interface IFighter
{
    //Denne property skal angive hvor mange forsvarspoint den
    //pågældende fighter har tilbage
    int DefenseLeft { get; }

    //Denne metode kaldes når fighteren skal forsvare sig
    void Defend(int attack);

    //return true hvis fighteren stikker af
    bool HasEscaped();

    //Denne metode kaldes når fighteren skal angribe
    //og returnere en int værdi med hvor meget han /hun angriber
    int Attack();
}
```

2. Opret 2 riddere og simuler en kamp mellem disse 2. Du kan bare gøre det fra main-metoden. Indsæt metoden "fight" og benyt denne til at simulere dit spil. **Du må ikke ændre i metoden eller signaturen!**

Du downloade metoden her

https://drive.google.com/open?id=1Gxl37lhgmNkwU94mwPEbCTC_C7O7L1uF

Nu da du har afprøvet simuleringen med 2 riddere, er det tid til lidt hjernegymnastik! Du skal nu tilføje en ny spilkarakter.

Spillet mangler nemlig en "Wizard". Implementeringen af en Wizard er allerede udviklet af en 3. part og du skal nu inkorporere denne "Wizard" i dit setup. Du kan download klassen "Wizard" her

https://drive.google.com/open?id=1RIZTzY8ih3_aZUATolk3IFn7LvasZuUt

Du må ikke ændre i Wizard koden!

3. Nu skal du kalde metoden "fight" igen, denne gang med en Knight og en Wizard. Hvordan vil du gøre det, når du ikke må ændre hverken i "fight" eller "Wizard"? Find en løsning på dette ved hjælp af et adapter pattern.

```
public static IFighter Fight(IFighter f1, IFighter f2)
{
    while ((!f1.HasEscaped() && !f2.HasEscaped()) && ((f1.DefenseLeft > 0) && (f2.DefenseLeft > 0)))
    {
        // Første fighter henter attack
        int attack = f1.Attack();
        // Anden fighter skal forsvare sig
        f2.Defend(attack);
        // Anden fighter henter attack
        attack = f2.Attack();
        // Første fighter skal forsvare sig
        f1.Defend(attack);
    }

    IFighter winner = null;

    // kampen er afsluttet
    if ((f1.DefenseLeft > 0) && (!f1.HasEscaped()))
        winner = f1;

    if ((f2.DefenseLeft > 0) && (!f2.HasEscaped()))
        winner = f2;

    // Hvis der returneres null, så har kampen været jævnbyrdig
    return winner;
}
```

Hvis du har mere tid....

Prøv også at implementere en drage. Du kan downloade "Dragen" her

https://drive.google.com/open?id=1kg0x6RsaAMgRAeshYKAWk_0g-MYF7PF5

En drage **angriber** hver anden gang med sin hale og hver anden gang spyr den ild.

