class Knight

{

    private int \_health; //tuto private hodnotu potřebuji pro custom ověření - viz public int Health

    public static int KnightCount { get; private set; } //příklad fungování static

    public string Name { get; } //if nenastavím public bude defaultně private, dám pouze {get;} abych nemohla změnit

    public int Strength { get; } //nemám zde {set;} tzn. že mohu nastavit jen v konstruktoru => zabráním nežádoucí změně

    // public int Health;

    public int Health

    {

        get

        {

            return \_health;

        }

        private set //(int value)

        {

            if (value < 0)

            {

                \_health = 0;

            }

            else

            {

                \_health = value;

            }

        }

    }

    // //analogie s metodou (if bych místo (int health) si to pojmenovala (int value)):

    // public void AddHealth(int value) //můj setter - nutný když změním Health na private (a nevyužiji {set;})

    // {

    //     if (\_health < 10)

    //     {

    //         this.\_health += value;

    //     }

    // }

//konstruktor souhrnný včetně defaultního nastavení\*\*

    public Knight(string name, int strength = 20, int health = 100)

    //public Knight(string name = "Null", int strength=20, int health=100) by vzal i zadání bez jména

    {

        this.Name = name;

        this.Strength = strength;

        this.Health = health;

        Knight.KnightCount += 1;

    }

}