Naprogramujte aplikaci, která obsluhuje člověka. Člověk má jméno a únavu, která je zpočátku 0. Může uběhnout určitou vzdálenost a také spát určitou dobu. Běháním se jeho únava zvyšuje (1 jednotka únavy na 1 km), spaním se snižuje (10 jednotek únavy na 1 hodinu). Navrhněte třídu tak, aby se únava nikdy nemohla dostat z rozmezí 0-20 jednotek.

Samozřejmě vám k tomu pomůže zapouzdření, únava určitě nebude veřejným atributem.

Program vyzkoušíte tak, že necháte člověk 3x uběhnout 10 km. Třetí uběhnutí by se nemělo povést. Když člověka necháte po druhém uběhnutí hodinu spát, zvládne i třetí běh.

Na úvod vypište instanci člověka. Definujte mu metodu ToString() tak, aby se vypsalo jeho jméno a věk.

Clovek karel = new Clovek("Karel Nový", 25);

Console.WriteLine(karel);

karel.Behej(10);

karel.Behej(10);

karel.Spi(1);

karel.Behej(10);

Console.ReadKey();

## Překrývání metody ToString()

Seznámíme se s metodu, kterou budeme používat i ve většině našich dalších objektů. Jedná se o metodu **ToString()**,kterou obsahuje každý objekt. Metoda je určena k tomu, aby vrátila tzv. textovou reprezentaci instance. Hodí se ve všech případech, kdy si instanci potřebujeme vypsat nebo s ní pracovat jako s textem. Tuto metodu mají např. i čísla. Již víme, že v C# funguje implicitní konverze, jakmile tedy budeme chtít do konzole vypsat číslo nebo kterýkoli jiný objekt, C# na něm zavolá metodu **ToString()** a vypíše její výstup. Pokud si děláme vlastní třídu, měli bychom zvážit, zda se nám takováto metoda nehodí.

/// <summary>

/// Vrátí textovou reprezentaci člověka

/// </summary>

/// <returns>textová reprezentace člověka</returns>

**public override string ToString()**

**{**

**return string.Format("{0} ({1})", jmeno, vek);**

**}**