

Wstęp:

W naszej firmie każdy pracownik po rozpoczęciu pracy zaznacza swoją obecność. Po zakończeniu pracy pracownik zaznacza koniec pracy.

Każdy pracownik ma swoją reprezentację w aplikacji w formie klasy. Np.

- pracownikowi o imieniu Roman odpowiada klasa RomanEmployee
- pracownikowi o imieniu Czesław odpowiada klasa CzeslawEmployee

Obecność pracownika jest zapamiętywana tylko w pamięci aplikacji do której jest wprowadzony obiekt klasy ImieEmployee. Klasa ImieEmployee jest napisana zgodnie ze wzorcem Singleton dla każdego pracownika osobno.

Opis działania aplikacji dla pracownika o imieniu Roman:

Aplikacja powinna pozwolić na utworzenie tylko jednego obiektu roman na podstawie klasy RomanEmployee. Sprawdzenie czy Roman jest w pracy polega na wywołaniu metody IsWorking() zwracającej True lub False na klasie (nie na obiekcie) RomanEmployee. Jeśli istnieje obiekt klasy RomanEmployee np.

```
Console.WriteLine($"Pracownik jest w pracy: { RomanEmployee.IsWorking()}");
```

```
//Zwraca 'Pracownik jest w pracy: True' lub 'Pracownik jest w pracy: False'.
```

Metoda Start() tworzy instancję klasy RomanEmployee

Metoda Stop() niszczy instancję klasy RomanEmployee

Zadanie:

Utwórz program konsolowy w którym utwórz klasę RomanEmployee. Klasa powinna spełniać warunki opisane wyżej. Zmień metodę Main na poniższą która sprawdza czy Roman jest w pracy.

```
static void Main(string[] args)
{
    Console.WriteLine("Roman rozpoczyna pracę");
    var roman = RomanEmployee.Start();
    Console.WriteLine($"Pracownik jest w pracy: {
RomanEmployee.IsWorking()}");
    Console.WriteLine("Roman kończy pracę");
    RomanEmployee.Stop();
    Console.WriteLine($"Pracownik jest w pracy: {
RomanEmployee.IsWorking()}");

    //Bonus
    //Console.WriteLine("Roman nie pracuje");
    //Console.WriteLine($"Pracownik pracuje już
{RomanEmployee.WorkingTime()}");
    //Console.WriteLine("Roman pracuje");
    //roman = RomanEmployee.Start();
    //Thread.Sleep(3000); // Czeka 3 sekundy.
    //Console.WriteLine($"Pracownik pracuje już
{RomanEmployee.WorkingTime()}");
    Console.Read();
}
```

Bonus:

Do klasy RomanEmployee dodaj metodę WorkingTime() która zwróci ilość czasu przepracowanego od momentu uruchomienia metody Start(). Jeśli pracownik jest w pracy to metoda WorkingTime() zwraca wartość czasu jaki jest przepracowany, jeśli pracownik nie pracuje zwrócona wartość to -1 sekund.