Cel

Celem zajęć jest zapoznanie się z pisaniem wielowątkowych aplikacji. W tym celu napiszemy program, który wykonuje dwa wątki.

Zadanie 1

Utwórz nowy projekt aplikacji konsolowej. W odpowiednim pliku dodaj bibliotekę obsługującą wielowątkowość

```
using System.Threading;
```

W statycznej metodzie **main** zadeklaruj dwa obiekty typu **Thread**. Nazwij je w sposób jednoznaczny.

```
Thread watek1;
Thread watek2;
```

By utworzyć obiekty watek1 i watek2 należy wcześniej napisać metody, jakie będą wykonywały. Mogą to być metody bez argumentów lub metody z argumentem typu object. Niezależnie od argumentów metody nie zwracają wyniku. Utwórz na początku puste metody o dowolnej nazwie, przyjmujące jeden argument typu object.

```
static void wykonanieWatku1(object par)
{
  argumentWatku parametry = par as argumentWatku;
}
static void wykonanieWatku2(object par)
{
  argumentWatku parametry = par as argumentWatku;
}
```

Zadanie 2

Utwórz klasę **argumentWatku**.. Do klasy argumentWatku dodaj publiczne pole typu int. Utwórz konstruktor, który przyjmuje jeden argument i ustawia pole klasy.

Utwórz dwa obiekty wątków. Jako argument przekaż statyczne metody utworzoną w poprzednim zadaniu. Zastanów się,w którym miejscu kodu utworzyć te wątki.

```
  watek1 = new Thread(new ParameterizedThreadStart(wypisywanieWatku1));
  watek2 = new Thread(new ParameterizedThreadStart(wypisywanieWatku2));
```

Uruchom oba wątki. Jako argument przekaż nowo utworzony obiekt typu argumentWatku.

```
  watek1.Start(new argumentWatku(  ));
  watek2. …
```

W głównej metodzie main poczekaj na zakończenie wykonywania obu wątków. Zastosuj do tego metodę **Join()**. Następnie wczytaj tekst, by zapobiec zamknięciu okna.

```
...
watek2.Join();
System.Console.ReadLine();
```

Zadanie 3

Zaimplementuj dwie metody statyczne zwracające wynik typu int. Jedna metoda rekurencyjna oblicza silnię z liczby n podanej jako argument. Druga metoda rekurencyjna oblicza n-ty wyraz ciągu Fibonacciego. Można również zaimplementować inne metody rekurencyjne.

Nowo utworzone metody wywołaj w metodach obsługujących wątki. Zastanów się w jaki sposób najłatwiej przekazać argumenty do metod obliczających silnię oraz wyraz ciągu Fibonacciego.

Zadanie 4

W celu określenia szybkości wykonywania się programu pisanego wielowątkowo w porównaniu do programu jednowątkowego warto użyć metody do obliczania wykonania się danego kodu programu. Poniżej sposób zastosowania takiego mechanizmu:

```
using System.Diagnostics;

Stopwatch sw = new Stopwatch();
sw.Start();
//kod poddany pomiarowi
sw.Stop();
Console.WriteLine(sw.ElapsedMilliseconds);
```

W metodzie main w pierwszej kolejności uruchom dwa wątki i przy ich pomocy wykonaj metody obsługujące wątki, a w nich metody obliczające silnię oraz wyraz ciągu Fibonacciego. Dokonaj pomiaru czasu wykonania się takiego kodu.

Następnie wykonaj statyczne metody obliczające silnię oraz wyraz ciągu Fibonacciego bez użycia wielowątkowości. Dokonaj także pomiaru czasu wykonania kodu.

Porównanie wykonaj dla argumetnów metod rekurencyjnych dobranych eksperymentalnie.

Czy jest różnica w czasie wykonania dwóch bloków kodów?