



## Zadanie 1

- Przepisz klasę `CriticalSectionUnsafe` aby uniknąć “resource race”
  - Użyj `lock`
  - Użyj klasy `Interlocked`

## Zadanie 2

- Użyj klasy Task do implementacji sytuacji gdzie zdanie "Hello {Name}!!!" jest wyświetlane na konsoli co sekundę
- Kiedy jakiś przycisk na klawiaturze jest wciśnięty, przestań wypisywać "Hello {Name}!!!", wypisz raz "Bye" i czekaj na dalszy input od użytkownika
- Użyj CancellationToken.



## Zadanie 3

- Napisz algorytm wykonujący równoległe następujące zadanie
- Każde zadanie sprawdza czy liczba, którą otrzyma w parametrze metody jest liczbą pierwszą (Task<bool>)
- Górna granica wyszukiwania może być ustalana losowo między 10 a 300
- Zlicz wszystkie powodzenia próby ustalenia liczby pierwszej, a wartość zwróć przez zadanie.
- Zlicz sumę dla rezultatów zakończonych powodzeniem i wypisz tę sumę.
- Użyj statycznej metody z klasy Task do synchronizacji. (WaitAll lub WhenAll)



## Zadanie 4

- Napisz prostą metodę która rzuca wyjątkiem.
- Złap i obsłuż wyjątek wypisując go do konsoli. W tym celu użyj:
  - Klasy Thread
  - Klasy Task.Wait()
  - async/await
  - BeginInvoke/EndInvoke na delegatach
  - Klasy Parallel



## Zadanie 5

- Zaimplementuj rozwiązanie problem pięciu filozofów
- Użyj semaphore w rozwiązaniu
- Użyj CancellationToken do zatrzymania procesu



## Zadanie 6 (Winforms lub WPF)

- Odczytaj plik
  - Wypisz jego zawartość na TextBox
  - Użyj składni async/await
  - Użyj metody `ReadStream.ReadToEndAsync`
- 
- Button, OpenFileDialog, TextBox



## Zadanie 7

- Zaimplementuj zadanie znajdowania liczby liczb pierwszych w aplikacji okienkowej (WPF, lub WinForms)
- Kiedy użytkownik naciśnie przycisk “Start” odpowiedni process jest wykonywany
- Przy okazji wyliczania kolejnych liczb pierwszych:
  - odświeżany jest Progressbar dla każdej liczby
  - Zwiększany jest counter w kontrolce dla każdej znalezionej liczby



## Zadanie 8

- Używając klasy Parallel pobierz listę plików z zadanego folderu do folderu C:\Temp\
- Wypisz na konsolę wszystkie pobrane nazwy plików wraz z ich wielkością (w KB)
- Obsłuż ewentualny wyjątek metody z klasy Parallel, wypisz wszystkie błędy





## Zadanie 9

- Używając PLINQ wypisz na konsolę listę plików z folderu C:\Temp\
- Wypisz na konsolę liczbę plików oraz ich łączną wielkość (w KB)



## Zadanie 10

- Zaimplementuj zadanie z wypisywaniem sumy znalezionych liczb pierwszych z użyciem PLINQ



## Zadanie 11

- Zaimplementuj Opcję wyszukiwania z użyciem TextBox wykorzystując Reactive Extensions (Rx) (FromEventPattern, Where, Throttle)
- Zaczniij szukać kiedy użytkownik prowadzi przynajmniej 3 znaki
- Nie zaczynaj wyszukiwania dopóki użytkownik pisze
  - odstęp pomiędzy kolejnymi kliknięciami na klawiaturze jest  $< 500$  ms



## Zadanie 12

- Użyj Reactive Extension do obsługi zdarzeń w aplikacji desktopowej
- Obsłuż zdarzenie poruszania kursorem myszki nad kontrolką
  - Na etykiecie wypisz aktualne koordynaty pozycji kursora



## Zadanie 13

- Zmodyfikuj klasę SimpleDictionarySample, tak aby była wątkowo bezpieczna



## Zadanie 14

- Zaimplementuj rozwiązanie problemu producenta/konsumenta
  - Utwórz jednego producenta i jednego konsumenta
  - Produktami mogą być porcje tekstu
  - Co w przypadku kiedy jest więcej producentów / konsumentów



## Zadanie 15

- Przeanalizuj testy jednostkowe metod asynchronicznych