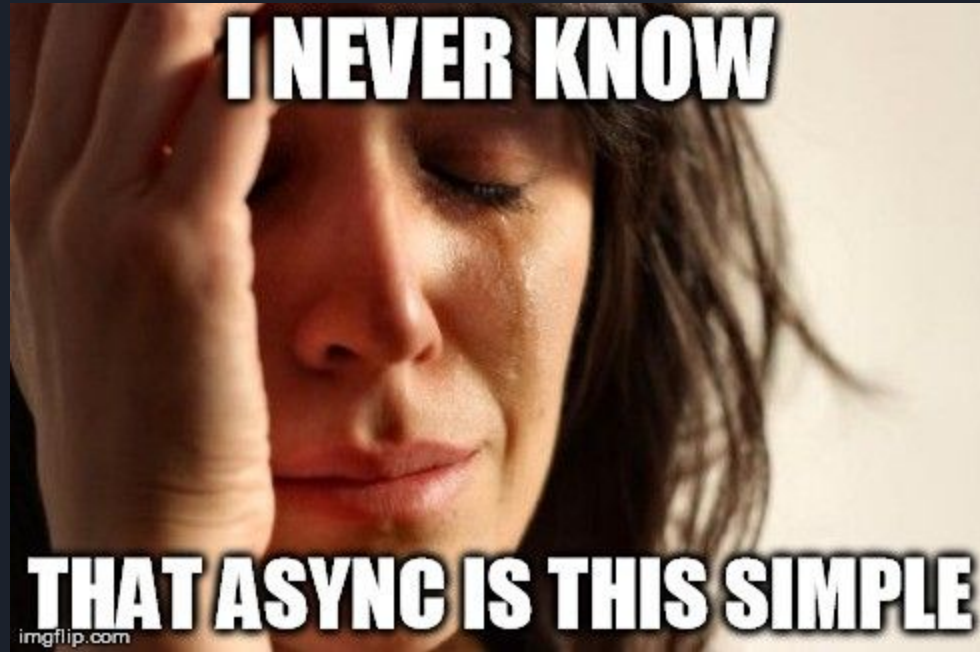


Използване на класа Task - 2 част





Връщане на стойност от Task

- За да се върне стойност от Task той трябва да се създаде с шаблонна нотация:
 - Task<тип> обект
- Резултатът се получава чрез: обект.Result
- Ако задачата не е готова, а достъпим .Result се изчаква.



Продължение на задача

- Може да закачите продължение на задача, което работи с резултата от предишната чрез `ContinueWith` метода:
- В метода чрез ламбда израз може да се укаже продължаващата задача
- Може да се укаже продължение при успех, провал или отмяна.



Пример - продължение на задача

```
Task<int> task = Task.Run(() =>  
  
    {  
  
        return 42;  
  
    }).ContinueWith((i) =>  
  
    {  
  
        return i.Result * 2;  
  
    });  
  
Console.WriteLine(task.Result); // Displays 84
```



Чакане на всички задачи

- Понякога се налага да изчакате приключването на всички задачи, които сте стартирали
- Използва се:
 - `Task.WaitAll(tasks);` или `Task.WaitAll(task1, task2, ...)`
- `Task.WaitAll` блокира нишката до приключването на всички активни задачи, които са посочени като параметър

Благодаря за вниманието!
Автор: Петър Р. Петров,
ПГЕЕ „К. Фотинов“, гр. Бургас

