Фишман Денис Борисович Группа:ПВ225

**Задание.**

1. Ознакомиться с материалом главы 16 - <https://metanit.com/sharp/tutorial/13.3.php>

2. Ознакомиться со статьей на MSDN: <https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/concepts/async/#asynchronous-exceptions>

3. Реализовать программу приготовления завтрака (https://learn.microsoft.com/ru-ru/dotnet/csharp/asynchronous-programming/) двумя способами:

1) Синхронно.

2) Асинхронно с использованием async/await/

4. Замерить время работы программы в обоих случаях. Вывести результаты работы и времени выполнения. Асинхронный 4 секунды, Синхронный 15 секунд

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DzAsincAwaitConsol\_21\_11\_2023

{

// These classes are intentionally empty for the purpose of this example. They are simply marker classes for the purpose of demonstration, contain no properties, and serve no other purpose.

internal class Bacon { }

internal class Coffee { }

internal class Egg { }

internal class Juice { }

internal class Toast { }

internal class ProgramAsinc

{

// для проверки раскоментировать Rmain, закоментировать ассинхронный Main

//static void Rmain(string[] args)

static async Task Main(string[] args)

{

Console.WriteLine("\nРаботает Асинхронный вариант\n");

Console.WriteLine($"Время: {DateTime.Now.ToLongTimeString()}\n");

Coffee cup = PourCoffee();

Console.WriteLine("coffee is ready");

var eggsTask = FryEggsAsync(2);

var baconTask = FryBaconAsync(3);

var toastTask = MakeToastWithButterAndJamAsync(2);

var breakfastTasks = new List<Task> { eggsTask, baconTask, toastTask };

while (breakfastTasks.Count > 0)

{

Console.WriteLine($"Время: {DateTime.Now.ToLongTimeString()}\n");

Task finishedTask = await Task.WhenAny(breakfastTasks);

if (finishedTask == eggsTask)

{

Console.WriteLine($"Время: {DateTime.Now.ToLongTimeString()}\n");

Console.WriteLine("eggs are ready");

}

else if (finishedTask == baconTask)

{

Console.WriteLine($"Время: {DateTime.Now.ToLongTimeString()}\n");

Console.WriteLine("bacon is ready");

}

else if (finishedTask == toastTask)

{

Console.WriteLine($"Время: {DateTime.Now.ToLongTimeString()}\n");

Console.WriteLine("toast is ready");

}

await finishedTask;

breakfastTasks.Remove(finishedTask);

}

Console.WriteLine($"Время: {DateTime.Now.ToLongTimeString()}\n");

Juice oj = PourOJ();

Console.WriteLine("oj is ready");

Console.WriteLine("Breakfast is ready!");

Console.WriteLine($"Время: {DateTime.Now.ToLongTimeString()}\n");

}

static async Task<Toast> MakeToastWithButterAndJamAsync(int number)

{

var toast = await ToastBreadAsync(number);

ApplyButter(toast);

ApplyJam(toast);

return toast;

}

private static Juice PourOJ()

{

Console.WriteLine("Pouring orange juice");

return new Juice();

}

private static void ApplyJam(Toast toast) =>

Console.WriteLine("Putting jam on the toast");

private static void ApplyButter(Toast toast) =>

Console.WriteLine("Putting butter on the toast");

private static async Task<Toast> ToastBreadAsync(int slices)

{

for (int slice = 0; slice < slices; slice++)

{

Console.WriteLine("Putting a slice of bread in the toaster");

}

Console.WriteLine("Start toasting...");

await Task.Delay(3000);

Console.WriteLine("Remove toast from toaster");

return new Toast();

}

private static async Task<Bacon> FryBaconAsync(int slices)

{

Console.WriteLine($"putting {slices} slices of bacon in the pan");

Console.WriteLine("cooking first side of bacon...");

await Task.Delay(3000);

for (int slice = 0; slice < slices; slice++)

{

Console.WriteLine("flipping a slice of bacon");

}

Console.WriteLine("cooking the second side of bacon...");

await Task.Delay(3000);

Console.WriteLine("Put bacon on plate");

return new Bacon();

}

private static async Task<Egg> FryEggsAsync(int howMany)

{

Console.WriteLine("Warming the egg pan...");

await Task.Delay(3000);

Console.WriteLine($"cracking {howMany} eggs");

Console.WriteLine("cooking the eggs ...");

await Task.Delay(3000);

Console.WriteLine("Put eggs on plate");

return new Egg();

}

private static Coffee PourCoffee()

{

Console.WriteLine("Pouring coffee");

return new Coffee();

}

}

}

namespace DzAsincAwaitConsol\_21\_11\_2023

{

internal class Program

{

// These classes are intentionally empty for the purpose of this example. They are simply marker classes for the purpose of demonstration, contain no properties, and serve no other purpose.

internal class Bacon { }

internal class Coffee { }

internal class Egg { }

internal class Juice { }

internal class Toast { }

// для проверки раскоментировать синхронный вариант, закоментировать --- Pmain

//static void Main(string[] args)

static void Pmain(string[] args)

{

Console.WriteLine("\nРаботает синхронный вариант\n");

Console.WriteLine ($"Время: {DateTime.Now.ToLongTimeString()}\n");

Coffee cup = PourCoffee();

Console.WriteLine("coffee is ready");

Console.WriteLine($"Время: {DateTime.Now.ToLongTimeString()}\n");

Egg eggs = FryEggs(2);

Console.WriteLine("eggs are ready");

Console.WriteLine($"Время: {DateTime.Now.ToLongTimeString()} \n");

Bacon bacon = FryBacon(3);

Console.WriteLine("bacon is ready");

Console.WriteLine($"Время: {DateTime.Now.ToLongTimeString()} \n");

Toast toast = ToastBread(2);

ApplyButter(toast);

ApplyJam(toast);

Console.WriteLine("toast is ready");

Console.WriteLine($"Время: {DateTime.Now.ToLongTimeString()} \n");

Juice oj = PourOJ();

Console.WriteLine("oj is ready");

Console.WriteLine("Breakfast is ready!");

Console.WriteLine($"Время: {DateTime.Now.ToLongTimeString()} \n");

}

private static Juice PourOJ()

{

Console.WriteLine("Pouring orange juice");

return new Juice();

}

private static void ApplyJam(Toast toast) =>

Console.WriteLine("Putting jam on the toast");

private static void ApplyButter(Toast toast) =>

Console.WriteLine("Putting butter on the toast");

private static Toast ToastBread(int slices)

{

for (int slice = 0; slice < slices; slice++)

{

Console.WriteLine("Putting a slice of bread in the toaster");

}

Console.WriteLine("Start toasting...");

Task.Delay(3000).Wait();

Console.WriteLine("Remove toast from toaster");

return new Toast();

}

private static Bacon FryBacon(int slices)

{

Console.WriteLine($"putting {slices} slices of bacon in the pan");

Console.WriteLine("cooking first side of bacon...");

Task.Delay(3000).Wait();

for (int slice = 0; slice < slices; slice++)

{

Console.WriteLine("flipping a slice of bacon");

}

Console.WriteLine("cooking the second side of bacon...");

Task.Delay(3000).Wait();

Console.WriteLine("Put bacon on plate");

return new Bacon();

}

private static Egg FryEggs(int howMany)

{

Console.WriteLine("Warming the egg pan...");

Task.Delay(3000).Wait();

Console.WriteLine($"cracking {howMany} eggs");

Console.WriteLine("cooking the eggs ...");

Task.Delay(3000).Wait();

Console.WriteLine("Put eggs on plate");

return new Egg();

}

private static Coffee PourCoffee()

{

Console.WriteLine("Pouring coffee");

return new Coffee();

}

}

}