

ТЕХНИКА ФАНЛАРИ

6 ЖИЛД, 1 СОН

ТЕХНИЧЕСКИЕ НАУКИ

ТОМ 6, НОМЕР 1

TECHNICAL SCIENCES

VOLUME 6, ISSUE 1



ТЕХНИКА ФАНЛАРИ

ТЕХНИЧЕСКИЕ НАУКИ | TECHNICAL SCIENCES

№1 (2023) DOI <http://dx.doi.org/10.26739/2181-9696-2023-1>

Бош мухаррир:
Главный редактор:
Chief Editor:

Юсулбеков Нодирбек Рустамбекович
Техника-фанлари доктори, профессор

Бош мухаррир ўринбосари:
Заместитель главного редактора:
Deputy Chief Editor:

Игамбердиев Хусан Закирович
Техника-фанлари доктори, профессор

TAHRIRIY MASLAHAT KENGASHI | EDITORIAL BOARD | РЕДАКЦИОННЫЙ СОВЕТ

Мардонов Ботир - техника фанлари доктори, профессор, "Табий тоаларни дастлабки ишлаш технологияси" кафедра профессори.

Исматуллаев Патхулла Рахматович - Техника-фанлари доктори, профессор.

Рахмонов Анвар Тожибоевич - Техника-фанлари доктори, профессор

Хакимов Шеркул Шергозиевич - техника фанлари доктори, доцент, "Технологик машиналар ва жиҳозлар" кафедра доценти

Шин Илларион Георгиевич - техника фанлари доктори, доцент, "Машинашунослик ва сервис хизмати" кафедра профессори

Джураев Анвар - техника фанлари доктори, профессор, "Машинашунослик ва сервис хизмати" кафедра профессори

Хамраева Сановар Атоевна - техника фанлари доктори, профессор, Магистратура бўлими бошлиғи

Нигматова Фотима Усмановна - техника фанлари доктори, профессор, "Тикув буюмларини конструкциялаш ва технологияси" кафедра профессори

Ташпулатов Салих Шукурович - техника фанлари доктори, профессор, "Костюм дизайни" кафедра профессори

Набиева Ирода Абдусаматовна - техника фанлари доктори, профессор, "Кимёвий технология" кафедраси мудири

Худайбердиева Дильфуза Бахрамовна - техника фанлари доктори, профессор, "Кимёвий технология" кафедраси профессори

Бабаханова Халима Абишевна - техника фанлари доктори, доцент, "Матбаа ва қадоклаш жараёнлари технологияси" кафедраси профессори

Рафиков Адхам Салимович - профессор, "Кимё" кафедраси мудири

Ахмедов Жаҳонгир Адхамович - техника фанлари доктори, доцент, "Ипак ва йиғириш технологияси" кафедра доценти

Юлдашев Уришбой - Техника фанлари доктори

Усманкулов Алишер Қодирқулович - Техника фанлари доктори

Абдуназаров Жамшид Нурмухаматович - Техника фанлари номзоди

Почужевский Олег Дмитриевич - кандидат технических наук, доцент по кафедре "Подъемно-транспортные машины", работаю доцентом кафедры "Автомобильный транспорт" Криворожского национального университета (Украина, г. Кривой Рог).

Полвонов Омонжон Хусанбой ўғли - Ислом Каримов номидаги Тошкент давлат техника университети Кўкон филиали ассистенти.

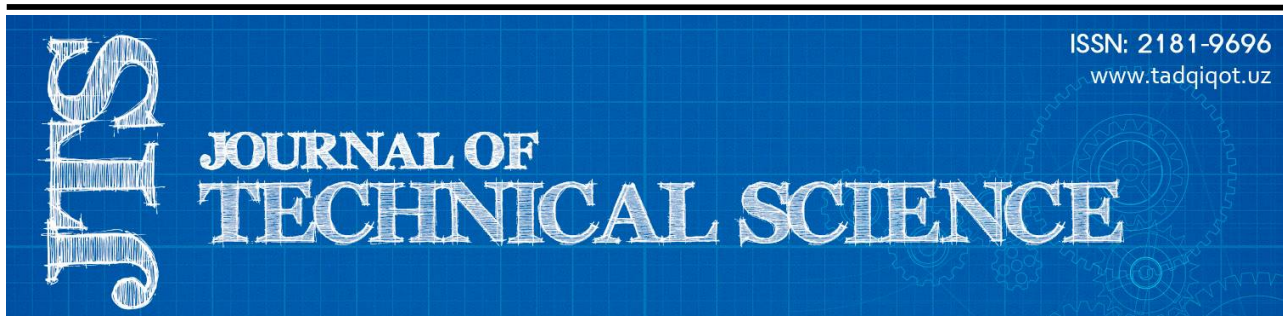
Тошпулатов Исломжон Адилжон ўғли - Ислом Каримов номидаги Тошкент давлат техника университети Кўкон филиали ассистенти

Page Maker | Верстка | Саҳифаловчи: Хуршид Мирзахмедов

Контакт редакций журналов. www.tadqiqot.uz
ООО Tadqiqot город Ташкент,
улица Амира Темура пр.1, дом-2.
Web: <http://www.tadqiqot.uz/>; Email: info@tadqiqot.uz
Тел: (+998-94) 404-0000


Editorial staff of the journals of www.tadqiqot.uz
Tadqiqot LLC the city of Tashkent,
Amir Temur Street pr.1, House 2.
Web: <http://www.tadqiqot.uz/>; Email: info@tadqiqot.uz
Phone: (+998-94) 404-0000

1. Hakimov Z. A. EFFECTS OF FUNCTIONAL CHANGES IN THE FOREBRAIN ON HUMAN FACIAL MUSCLE MOVEMENTS.....	4
2. Jovlieva Fayoz Ulashovna INCREASED SECURITY USING REMOTE SYSTEM MANAGEMENT REDESIGN.....	10
3. Mamatqulov Qosimjon Nasriddin o'g'li SOLIQ ORGANLARI BOSHQARUV TIZIMIDA E-AKTIV AXBOROT TIZMINING SINXRON VA ASINXRON DASTURALASH MODELLARI.....	14
4. Usmonov Jonibek Turdiqulovich, Mamatqulov Qosimjon Nasriddin o'g'li KORPORATIV INTEGRALLASAHGAN E-AKTIV AXBOROT TIZMIDA “CLICKHOUSE”DAN FOYDALANISH USTUNLIKLARI VA KAMCHILIKLARI.....	22
5. Джаматов Мустафа Хатамович, Мирзаева Малика Бахадировна IP XAVFSIZLIGINI BAHOLASH MEZONLARINING IKKI DARAJALI TIZIMI.....	27
6. To'rayev A. T., Turdiyev S. S., Bozorov A. A., Shamsiyeva O'. N. SHARTLI BOG'LIQSIZ TASODIFIY MIQDORLAR UCHUN MARKAZIY LIMIT TEOREMA.....	36
7. To'rayev A. T., Turdiyev S. S., Bozorov A. A., Shamsiyeva O'. N. VILKOKSON-MANN-UITEY STATISTIKASINING ASIMPTOTIK XOSSALARI.....	42



Mamatqulov Qosimjon Nasriddin o'g'li
“Yangi texnologiyalar” ilmiy-axborot markazi ilmiy-axborot
markazi Rouming operatori dasturiy ta’minoti bo’limi yetakchi mutaxasisi

SOLIQ ORGANLARI BOSHQARUV TIZIMIDA E-AKTIV AXBOROT TIZMINING SINXRON VA ASINXRON DASTURALASH MODELLARI

 <http://dx.doi.org/10.5281/zenodo.7993667>

ANNOTATSIYA

Maqolada axborot texnologiyalarining hayotimizning har bir sohada ommaviy ravishda joriy etilishi natijasida soliq organlarida ham zamon bilan hamnafas bo'lgan holatda soliq to'lovchi hamda soliq xizmati xodimlari o'rtasidagi shaffoflikni ta'minlovchi yangi E-Aktiv axborot tizimining imkoniyat va qulayliklar beradigan tizimning tez va samarali ishlash uchun maqbul dasturlash moqddellarini ko'rib chiqamiz.

Kalit so'zlar: Axborot modeli, axborotlashtirilgan tizim, soliq, sinxron dasturlash, asinxron dasturlash, async, await.

Маматкулов Қосимжон Насриддин ўғли
Ведущий специалист отдела программного
обеспечения операторов роуминга

Научно-информационного центра «Новые технологии»

СИНХРОННАЯ И АСИНХРОННАЯ ПРОГРАММНЫЕ МОДЕЛИ ЭЛЕКТРОННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ В СИСТЕМЕ УПРАВЛЕНИЯ НАЛОГОВЫМИ ОРГАНАМИ

АННОТАЦИЯ

В результате массового внедрения информационных технологий во все сферы нашей жизни в статье описывается новая информационная система E-Active, которая обеспечивает прозрачность между налогоплательщиком и работниками налоговой службы в состоянии, соответствующем времени, а также система, предоставляющая возможности и удобства, рассмотрим оптимальные модели программирования для быстрой и эффективной работы.

Ключевые слова: информационная модель, информационная система, налог, синхронное программирование, асинхронное программирование, асинхронность, ожидание.

Mamatqulov Qosimjon Nasriddin o'g'li
Leading Specialist of the Department of Software
for Roaming Operators of the Research and
Information Center "New Technologies"

SYNCHRONOUS AND ASYNCHRONOUS PROGRAM MODELS OF ELECTRONIC INFORMATION SYSTEM IN TAX AUTHORITIES MANAGEMENT SYSTEM

ABSTRACT

As a result of the mass introduction of information technology in every sphere of our life, the article describes the new E-Active information system, which ensures transparency between the tax payer and the tax service employees in a state that is in tune with the times, as well as the system that provides opportunities and conveniences. We will consider optimal programming models for fast and efficient work.

Key words: Information model, information system, tax, synchronous programming, asynchronous programming, async, await.

Kirish

Hozirgi kunda butun dunyoda raqamli iqtisodiyotga o'tish jarayoni ketmoqda. Shu bilan bir qatorda bizning mamlakatimizda ham bu borada ko'pgina ishlar olib borilmoqda. Mamlakatimiz rivojida hissa qo'shayotgan tadbirkorlar hozirgi kunda o'z tadbirkorlik subyektlarini avtomatlashtirib borishmoqda. Shu bilan bir qatorda o'z hisob kitoblarini soliq xizmati xodimlaridan yashirib mamlakatimiz iqtisodiyotiga zarar keltirayotganlar ham uchrab turibti. Shu boisdan hozirgi kunda mamlakatimizda soliq sohasida ham axborot texnologiyalarini tadbir qilish ishlari amalga oshirilib kelinmoqda. Shunday dasturlardan biri sifatida e-aktiv (e-ombor) axborot tizimini keltirib o'tishimiz mumkin. Bu borada O'zbekiston Respublikasi Vazirlar mahkamasining 23.12.2021 yildagi 595-sonli "Soliq to'lovchilarni hisobga olishni yanada takomillashtirish va qo'shilgan qiymat solig'ining o'rnini qoplash tartibini soddalashtirish chora-tadbirlari to'g'risida" qarorining 3-band "b" kichik bandida "xo'jalik yurituvchi subyektlar tomonidan asosiy vositalar, nomoddiy aktivlar va tovar-moddiy resurslarni hisobga olish (sotib olish (qurish), ishga tushirish, qayta baholash, tasarruf etish) hamda o'z ustav fondini (ustav kapitalini) shakllantirish va kamaytirish bilan bog'liq barcha operatsiyalarni davlat soliq organlarining "E-aktiv" avtomatlashtirilgan axborot tizimida aks ettirish" deb E-Aktiv axborot tizimidan foydalanish tartibi ko'rsatilgan. Bu axborot tizimi orqali ko'pgina yashirin iqtisodiyotni bartaraf etishga erishish mumkin. Hozirda E-Aktiv(E-Ombor) axborot tizimining ikkinchi versiyasi ishlab chiqilmoqda. Ushbu versiyasi avvalgi versiyasidan anchagina tez ishlashi va qulayliklari bilan birinchi versiyasidan farq qiladi. E-aktiv (e-ombor) axborot tizimining tez va samarali ishlashiga asinxron dasturlashni qo'llash orqali erishishimiz mumkin.

Dasturlashda sinxron va asinxron dasturlash modellari muhim ahamiyatga ega. Sinxron dasturlashda vazifalar tartib bilan amalga oshiriladi – keying vazifaga o'tishdan oldin oldingi vazifani bajarilishini kutadi. Asinxron dasturlashda vazifalar har qanday tartibda yoki hatto bir vaqtning o'zida bajarilishi mumkin.

Asinxron dasturlash

Asinxron dasturlash ko'p tarmoqli model bo'lib, u tarmoq va aloqa uchun eng mos keladigan dasturlash modelidir. Asinxron - bu bloklanmaydigan arxitektura, ya'ni u bir yoki bir nechta operatsiyalar bajarilayotganda keyingi bajarilishi kerak bo'lgan jarayonni bloklaydi.

Asinxron dasturlashda bir nechta tegishli operatsiyalar boshqa vazifalar tugashini kutmasdan bir vaqtning o'zida ishlashi mumkin. Asinxron aloqa paytida tomonlar xabarlarini qabul qilgandan so'ng darhol javob berish o'rniga, qulay yoki mumkin bo'lganda qabul qiladi va qayta ishlaydi Async va await kalit so'zlaridan foydalaniladi.

Sinxron dasturlash

Sinxron blokirovka qiluvchi arxitektura sifatida tanilgan va reaktiv tizimlarni dasturlash uchun ideal hisoblanadi. Bir ipli model sifatida u qat'iy ketma-ketlik to'plamiga amal qiladi, ya'ni operatsiyalar bir vaqtning o'zida mukammal tartibda amalga oshiriladi. Bitta operatsiya bajarilayotganda, keyingi jarayon bloklanadi. Birinchi vazifaning bajarilishi keyingi vazifani ishga tushiradi va hokazo.

Sinxron dasturlash qanday ishlashini ko'rsatish uchun telefon suhbatini misol qilishimiz mumkin. Bir kishi gapirganda, ikkinchisi tinglaydi. Birinchi kishi tugatsa, ikkinchisi darhol javob berishga intiladi.

Sinxron va asinxron dasturlashni Microsoft kompaniyasi nonushta tayyorlash misoli orqali tushuntiradi. Nonushta tayyorlashda 7 ta jarayon bor deb olamiz. Bular quyidagilar:

1. Kofe damlash;
2. Tovani qizdirish;
3. Tuxumlarni qovurish;
4. Kalbasa qovurish;
5. Aparat yordamida nonlarni qizartirish;
6. Qizargan nonlarga murabbo surtish;
7. Sharbat quyish;



1-ram. Sinxron va asinxron dasturlash ketma-ketligi

1-rasmda yuqorida keltirilgan jarayonlarning ketma-ketligi ko'rsatilgan. Sinxron dasturlashda ham huddi shunday ketma-ketlikda jarayonlar amalga oshiriladi. Avvaliga kofe damlanadi, so'ngra tova qizdiriladi. Tova qizigandan so'ng tuxumni qovurib olinadi. Tuxumni olgandan so'ng esa kalbasa qovuriladi. Undan so'ng nonlar aparat yordamida qizartiriladi hamda murabbo surtiladi. So'ngra sharbat quyiladi. Shu bilan nonushtamiz tayyor bo'ladi. Bu hayotimizda har kuni ko'radigan harayonimizdir. Ammo bu jarayonlarni boshlanishidan yortib to oxirigacha 30 daqiqa sarflaysiz. Bundan tashqari barcha jarayonlar tugaganidan so'ng nonushta qilishni boshlaydigan bo'lsangiz nonushtangiz sovub qolishi mumkin. Dastulash ham huddi shunga o'xshaydi ya'ni bitta jarayonni boshlab oxirigacha tugatishgacha ancha vaqt ketib qolishi mumkin. Bu esa foydalanuvchiga noqulaylik keltirib chiqaradi. Bundan tashqari agar dasturdan ko'p foydalanuvchi foydalanadigan bo'lsa dasturga tushayotgan og'irlik ortib dastur umuman og'ir ishlaydigan axborot tizimiga aylanib qolishi mumkin. Yuqoridagi yozilganlarni qanday qilib kodlarda ifodalash mumkin:

```
using System;
using System.Threading.Tasks;
namespace AsyncBreakfast
{
    internal class Bacon { }
    internal class Coffee { }
    internal class Egg { }
    internal class Juice { }
    internal class Toast { }
    class Program
    {
        static void Main(string[] args)
        {
            Coffee cup = PourCoffee();
            Console.WriteLine("coffee is ready");
            Egg eggs = FryEggs(2);
            Console.WriteLine("eggs are ready");
            Bacon bacon = FryBacon(3);
            Console.WriteLine("bacon is ready");
            Toast toast = ToastBread(2);
            ApplyButter(toast);
            ApplyJam(toast);
            Console.WriteLine("toast is ready");
            Juice oj = PourOJ();
            Console.WriteLine("oj is ready");
            Console.WriteLine("Breakfast is ready!");
        }
        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();
        }
    }
}
```

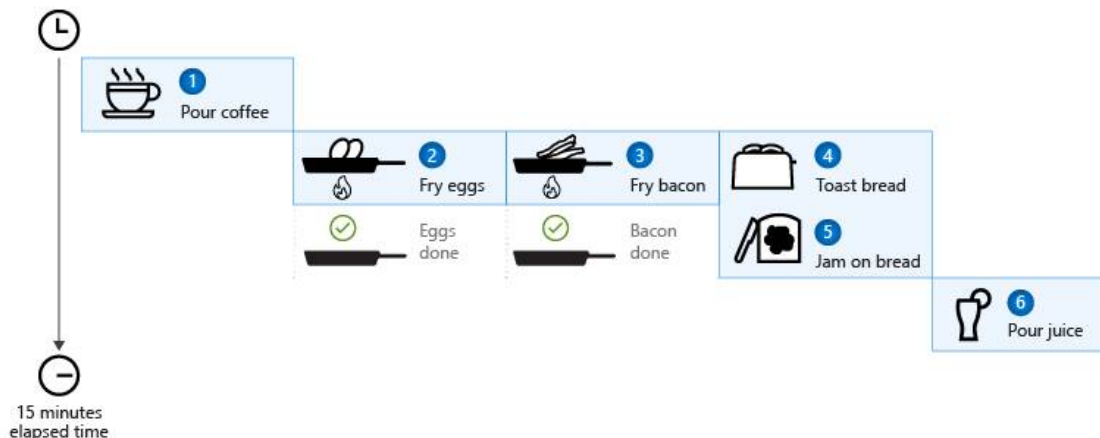
```

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();
}
}
}
}

```

Ushbu kod yuqorida tushuntirilgan jarayonning kod qismi hisoblanadi.

Endi buni qanday qilib optimallashtirishimiz mumkin ekanligi haqida bosh qotirishimiz kerak. Bunda ham hayotiy misol bilan tushuntiraman. Agar oshpaz tajribali bo‘ladigan bo‘lsa yuqoridagi jarayonni tezlashtirishi mumkin. Qanday deysizmi? Javobi oddiy ba’zi bir jarayonlarni parallel amalga oshirgan holatda albatta. Birinchi navbatta kofe damlab olamiz. Undan so‘ng esa bir biriga bog‘liq bo‘lmagan ishlarni bir vaqtning o‘zida amalga oshiramiz. Ya’ni tuxum va kalbasani qovurish uchun bitta emas 2 ta tovadan foydalanamiz. Bundan tashqari aparatga nonni qizartirish uchun qo‘yib qo‘yamiz. Shunda bir vaqtning o‘zida 3 ta jarayonni bajarayotgan bo‘lamiz. Ushbu jarayonlar bir biriga bog‘liq emas hamda mustaqil bajarsa bo‘ladigan jarayonlardir. Tuxum pishirishini kutib undan keyin tovaga kalbasa pishirishga ketadigan vaqtni anchasini tejab qolishga erishdik. Undan keying jarayonimiz esa qizargan nonlarge murabbo surkash. Bu jarayon o‘zidan oldingi jarayonga bog‘liq hisoblanadi ya’ni avval nonni qizarib chiqishini kutishimizga to‘g‘ri keladi. Jarayonning oxirida esa sharbat quyish bilan barcha jarayon yakunladi.



2-rasm. Asinxorn dasturlash jarayoning qadamlari va bajarilish ketma-ketligi

Yuqoridagi jarayonlar 2-rasmda ko'rsatilgan. Rasmdan ko'rinib turibdiki biz asinxron bajarish orqali 15 daqiqada nonushta tayyorlashga ulgurdik. Ya'ni 15 daqiqa tejaldi. Endi buni kodda qanday ko'rinishini ko'ramiz.

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Threading.Tasks;
```

```
namespace AsyncBreakfast
```

```
{
```

```
    internal class Bacon { }
```

```
    internal class Coffee { }
```

```
    internal class Egg { }
```

```
    internal class Juice { }
```

```
    internal class Toast { }
```

```
class Program
```

```
{
```

```
    static async Task Main(string[] args)
```

```
    {
```

```
        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)
```

```
        {
```

```
            Task finishedTask = await Task.WhenAny(breakfastTasks);
```

```
            if (finishedTask == eggsTask)
```

```
            {
```

```
                Console.WriteLine("eggs are ready");
```

```
            }
```

```
            else if (finishedTask == baconTask)
```

```
            {
```

```
                Console.WriteLine("bacon is ready");
```

```
            }
```

```
            else if (finishedTask == toastTask)
```

```
            {
```

```
                Console.WriteLine("toast is ready");
```

```
            }
```

```
            await finishedTask;
```

```
            breakfastTasks.Remove(finishedTask);
```

```
        }
```

```
        Juice oj = PourOJ();
```

```
        Console.WriteLine("oj is ready");
```

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

```
    }
```

```
    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();
    }
}
```

Yuqoridagi koddan ko‘rishimiz mumkinki asinxron dasturlashdan o‘z joyida foydalana olish dasturiy maxsulotning effektivligiga, tez ishlashiga ancha yordam beradi. Yuqoridagi kodni tahlil qilib savol tug‘ilishi mumkin “Dasturiy maxsulotning barcha joyi asinxron emasku. Unda nega asinxron deb atalmoqda?”. Asinxron dasturlash bo‘lishi uchun barcha kod asinxron bo‘lib ishlashi shart emas. Sababi barcha baribir bir jarayon ikkinchi jarayonga bo‘g‘liq bo‘lib qolgan joyi bo‘ladi. Shu sababli ham hamma joyda asinxron kodlashdan foydalanish juda qiyin masaladir.

Xulosa

Xulosa qilib shuni aytish mumkinki e-aktiv axborot tizimida ham asinxron dasturlashdan foydalanish axborot tizimidan foydalanish effektivligini oshirishga va foydalanuvchiga qulayliklar olib keladi. Albatta barcha jarayonni ham asinxron tarzda ishlaydigan qilishning iloji bo‘lmasligi mumkin. Ammo backenddan backendga murojatlarni va bir xil vazifa bajaruvchi holatlarni ya’ni bir nechta ma’lumotlar omboriga yoziladigan joylarda qo‘llash ancha qo‘l keladi.

Adabiyotlar ro‘yxati

1. <https://lex.uz/uz/docs/-5650010> O‘zbekiston Respublikasi Vazirlar Maxkamasining 22.09.2021 yildagi 595-sonli qarori, 3-band.
2. “Axborot-kommunikatsiya texnologiyalari izohli lug‘ati” BMTTDning O‘zbekistondagi vakolatxonasi, Toshkent 2010.
3. <https://avtovsamare.ru/uz/prognozirovanie-prodazh-na-mesyac-metodom-algoritm-prognozirovaniya-ob-ma-prodazh/>
4. <https://zobiki.ru/uz/poleznoe/raschet-prognoza-prognozirovanie-obema-prodazh-kak-opredelit-optimalnyi/>
5. Explained: Asynchronous vs. Synchronous Programming (mendix.com)
6. Asynchronous programming in C# | Microsoft Learn

ТЕХНИКА ФАНЛАРИ

6 ЖИЛД, 1 СОН

ТЕХНИЧЕСКИЕ НАУКИ
ТОМ 6, НОМЕР 1

TECHNICAL SCIENCES
VOLUME 6, ISSUE 1

Контакт редакций журналов. www.tadqiqot.uz
ООО Tadqiqot город Ташкент,
улица Амира Темура пр.1, дом-2.
Web: <http://www.tadqiqot.uz/>; Email: info@tadqiqot.uz
Тел: (+998-94) 404-0000

Editorial staff of the journals of www.tadqiqot.uz
Tadqiqot LLC The city of Tashkent,
Amir Temur Street pr.1, House 2.
Web: <http://www.tadqiqot.uz/>; Email: info@tadqiqot.uz
Phone: (+998-94) 404-0000