

Лабораториялық жұмыс 11. Parallel класының мүмкіндіктерін пайдалану.

Лабораториялық жұмыстың мақсаты: Студенттерде параллель орындалатын программалар құруда Parallel класының мүмкіндіктерін таңдап, пайдалану дағдыларын қалыптастыру.

Лабораториялық жұмысты орындау нәтижесінде студенттер келесі қабілеттерге ие болады:

- For(), ForEach(), Invoke() әдістерінің мүмкіндіктерін ажырату;
- For(), ForEach(), Invoke() әдістерін практикалық пайдалану.

Тапсырма: Төменде берілген мысалдармен танысып, нұсқа бойынша берілген есептерде For(), ForEach(), Invoke() әдістерінің қолданылу мүмкіндігін бағалап, барынша қолайлысын пайдаланыңыз.

Invoke() әдісінің көмегімен тапсырмаларды параллельдеу

```
using System.Threading;
using System.Threading.Tasks;
class DemoParallel {
// Тапсырма ретінде орындалатын әдіс.
static void MyMeth() {
Console.WriteLine("MyMeth іске қосылды");
for(int count = 0; count < 5; count++) {
Thread.Sleep(500);
Console.WriteLine("MyMeth әдісіндегі санауыш мәні: " + count );
}
Console.WriteLine("MyMeth аяқталды");
}
// Тапсырма ретінде орындалатын әдіс.
static void MyMeth2() {
Console.WriteLine("MyMeth2 іске қосылды");
for(int count = 0; count < 5; count++) {
Thread.Sleep(500);
Console.WriteLine("MyMeth2 әдісіндегі санауыш мәні: " + count );
}
Console.WriteLine("MyMeth2 аяқталды");
}
static void Main() {
Console.WriteLine("Негізгі ағын іске қосылды.");
// Екі атаулы әдісті параллель орындау.
Parallel.Invoke(MyMeth, MyMeth2);
Console.WriteLine("Негізгі ағын аяқталды.");
}
}
```

For() әдісін пайдалану

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

```

class DemoParallelFor {
static int[] data;
// Параллель орындалатын цикл тұлғасы ретінде қызмет ететін әдіс.
static void MyTrknsform(int i) {
data[i] = data[i] / 10;
if(data[i] < 10000) data[i] = 0;
if(data[i] > 10000 & data[i] < 20000) data[i] = 100;
if(data[i] > 20000 & data[i] < 30000) data[i] = 200;
if(data[i] > 30000) data[i] = 300;
}
static void Main() {
Console.WriteLine("Негізгі ағын іске қосылды.");
data = new int[100000000];
// Мәліметтерді әдеттегі for циклында инициалдау.
for(int i=0; i < data.Length; i++) data[i] = i;
// Циклді For() әдісімен параллель күйге келтіру.
Parallel.For(0, data.Length, MyTransform);
Console.WriteLine("Негізгі ағын аяқталды.");
}
}

```

ForEach() әдісін пайдалану

```

using System;
using System.Threading.Tasks;
class DemoParallelForWithLoopResult {
static int[] data;
// Параллель орындалатын цикл тұлғасы ретінде қызмет ететін әдіс.
static void DisplayData(int v, ParallelLoopState pls) {
// Теріс мән табылған жағдайда цикл жұмысын тоқтату.
if(v < 0) pls.Break();
Console.WriteLine("Мән: " + v);
}
static void Main() {
Console.WriteLine("Негізгі ағын іске қосылды.");
data = new int[100000000];
// Мәліметтерді инициалдау.
for(int i=0; i < data.Length; i++) data[i] = i;
// Теріс мәнді data жиымына ауыстыру,
data[100000] = -10;
// ForEach() әдісінің көмегімен параллель орындалатын циклді пайдалану
ParallelLoopResult loopResult = Parallel.ForEach(data, DisplayData);
// Циклдің аяқталғанын тексеру.
if(!loopResult.IsCompleted)
Console.WriteLine("\nЦикл мерзімінен бұрын аяқталды, себебі циклдің " +
loopResult.LowestBreakIteration + " нөмірлі қадамында теріс мән табылды\n");
Console.WriteLine("Негізгі ағын аяқталды.");
}
}

```

Тапсырмалар

1-нұсқа

Матрицаны векторға көбейтуді орындаңыз. Матрицаның әрбір жолы жеке ағында өңделуі тиіс.

2-нұсқа

Берілген аралыққа жататын барлық жай сандарды табыңыз. Тапсырманы орындау үшін классикалық Евклид алгоритмін қолданыңыз. Берілген аралықты кіші аралықтарға бөліп, әрбір ішкі аралықты жеке ағында өңдеңіз.

3-нұсқа

Аю мен аралардың өзара байланысу процесін бейнелейтін программа құрыңыз. N араның әрқайсысы бал жинауға қатысады: бір рет бал жинауға шыққанда барлығы бірдей мөлшерде ($M1$) бал жинайды және оған кездейсоқ уақыт жұмсайды. Аю X уақыт ішінде $M2$ мөлшерінде балмен қоректенеді және келесі X уақытта азықсыз тіршілік ете алады. Әрбір араның жұмысын жеке ағында жүзеге асырыңыз.

4-нұсқа

Шарлардың қозғалысын бейнелейтін программа құрыңыз. N шар берілген. Олардың тік және көлденең координаталары кездейсоқ шамаларға өзгереді. Егер шар берілген аймақ шекарасынан төмен түссе, жоғалып кетеді. Әрбір шардың координатасының өзгеруін жеке ағында жүзеге асырыңыз.

5-нұсқа

Топтардың қарсыласуын бейнелейтін программа құрыңыз. Ойыншылардың N тобы бар. Әрбір топтың ойыншылар саны кездейсоқ шамаға артады және қарсылас топтың кездейсоқ ойыншылар санын жояды. Әрбір топтар жұбының өзара қарсыластығы жеке ағында жүзеге асырылуы тиіс.

6-нұсқа

Бақылау суммасы. Өлшемдері әртүрлі N файл берілген. Әрбір файл үшін бақылау суммасын (файлдың барлық символдары кодтарының қосындысын) анықтау керек. Әрбір файл жеке ағында өңделуі тиіс.

7-нұсқа

Кедергілі жүгірісті бейнелейтін программа құрыңыз. Жүгіруге арналған трассаның матрица түріндегі шартты картасы жасалады. Матрицаның ені жүгірушілер санына тең, ал биіктігі бекітілген, кездейсоқ ұяшықтарға орналастырылған кездейсоқ кедергілер саны бейнеленеді. Жүгірушілер трасса бойымен жылжи отырып, кедергіге кезіккен жағдайда нақты анықталған уақытқа бөгеледі. Мәреге жеткен жүгірушілер өздерінің нөмірлерін хабарлайды. Әрбір жүгірушінің жұмысын жеке ағында жүзеге асырыңыз.

8-нұсқа

Қойлар мен қасқыр ойынын бейнелейтін программа құрыңыз. Бірнеше қой мен қасқырдың қозғалыстарын бейнелейтін программа құру қажет. Қасқыр мен қойдың координаталары сәйкес келген жағдайда, қой жоғалады. Егер екі қойдың координаталары сәйкес келсе, жаңа қой пайда болады. Қасқыр мен қойлар кездейсоқ қозғалады. Әрбір қойдың қозғалысын жеке ағында жүзеге асырыңыз.

9-нұсқа

$Y=23*x^2-33$ функциясының мәндерін $x=0.01$ қадамымен есептеуді орындаңыз. Есептелген мәндер x мәндерімен қатар жиымға жазылып отыруы тиіс. Жиымға жазылған x және y мәндерін экранға шығару керек. Мәндерді есептеу және жиымға жазу бір ағында, мәндерді жиымнан оқып, экранға шығару жеке ағында орындалуы тиіс.

10-нұсқа

Мәліметтер жиымын сұрыптау және сұрыптау күйін экранда бейнелеу. Бірінші ағында жиымды өсу реті бойынша, екінші ағында кему реті бойынша сұрыптау орындалуы тиіс. Әрбір элементтің орны ауысқанда экранда жиымның ағымдағы күйі бейнеленеді.

11-нұсқа

0 мен 9 аралығынан кездейсоқ сандарды генерациялайтын 3 ағын құрыңыз. Белгілі бір батырманы басқанда генерациялау тоқтатылып, генерацияланған сандар тізбектерінен келесідей ішкі тізбектерді іздеу қажет: қатар орналасқан үш бірдей сан, қатар орналасқан екі бірдей сан. Әрбір ағындық тізбек үшін осы ішкі тізбектер санын анықтаңыз.

12-нұсқа

Берілген аралықтан Фибоначчи сандарын іздейтін ағынды және жай сандарды іздейтін ағынды құру қажет. Сандар тізбектері екі жеке файлға жазылады, экранға сандар тізбектері және олардың мөлшері шығарылады.

13-нұсқа

Берілген файл құрамынан берілген тіркесті іздеу программасын құрыңыз. Әрбір файлдың қатары жеке ағымда өңделуі тиіс.

14-нұсқа

Файлға кездейсоқ мәліметтер жазып, оларды оқып, экранға шығару программасын құрыңыз. Файлға мәліметтерді жазу мен оларды оқып, экранға шығару екі жеке ағында орындалуы тиіс.