

Лабораториялық жұмыс 9. Көпәғындық немесе көптапсырмалы программада аластамаларды өңдеу.

Лабораториялық жұмыстың мақсаты: Студенттерде ағындармен немесе тапсырмалармен жұмыс барысында аластамаларды өңдеу дағдыларын қалыптастыру.

Лабораториялық жұмысты орындау нәтижесінде студенттер келесі қабілеттерге ие болады:

- Ағынның және/немесе тапсырманың жұмысындағы аластамалық жағдайды анықтау;
- Аластамалық жағдайдың ағындар/тапсырмалар жиынының жұмысына әсерін басқару.

Тапсырма: 2-ші және 7-ші лабораториялық жұмыстардағы программалар құрамындағы аластамалық жағдайларды айқындаңыз, орындалған тапсырмаларға аластамаларды өңдеу блоктарының жұмысын қосыңыз.

Аластамаларды өңдеу мысалдары

Мысал 1. Тапсырманың орындалуын болдырмау кезінде аластама өңдеу

```
using System;
using System.Threading;
using System.Threading.Tasks;
class DemoCancelTask {
// Тапсырма ретінде орындалатын әдіс
static void MyTask(Object ct) {
CancellationTok cancelTok = (CancellationTok) ct;
// Тапсырманы іске қосудың алдында оның жойылмағанын тексеру
cancelTok.ThrowIfCancellationRequested();
Console.WriteLine("MyTask() іске қосылды");
for(int count = 0; count < 10; count++) {
// Тапсырманың жойылғанын бақылау үшін сұрау тәсілі қолданылады
if(cancelTok.IsCancellationRequested) {
Console.WriteLine("Тапсырманы жоюға сұраныс түсті");
cancelTok.ThrowIfCancellationRequested();
}
Thread.Sleep(500);
Console.WriteLine("MyTask() әдісіндегі санауыш мәні: " + count );
}
Console.WriteLine("MyTask аяқталды");
}
static void Main() {
Console.WriteLine("Негізгі ағын іске қосылды");
// Тапсырманы жою белгілерінің көзі объектісін құру
CancellationTokSource cancelTokSrc = new CancellationTokSource();
// Тапсырманы іске қосу, тапсырмаға және делегатқа жою белгісін беру
Task tsk = Task.Factory.StartNew(MyTask, cancelTokSrc.Token, cancelTokSrc.Token);
// Тапсырманың күші жойылғанға дейін орындалуына мүмкіндік беру
Thread.Sleep(2000);
try {
```

```

// Тапсырманың орындалуынан бас тарту
cancelTokSrc.Cancel();
// tsk тапсырмасы аяқталғанға дейін Main() әдісінің орындалуын кідірту
tsk.Wait();
} catch (AggregateException exc) {
if(tsk.IsCanceled)
Console.WriteLine("\ntsk тапсырмасының орындалуынан бас тарту\n");
Console.WriteLine(exc);
} finally {
tsk.Dispose();
cancelTokSrc.Dispose();
}
Console.WriteLine("Негізгі ағын аяқталды.");
}
}

```

Мысал 2. Ағыннан жұмысын тоқтату кезіндегі аластама

```

using System;
using System.Threading;
class MyThread {
public Thread Thrd;
public MyThread(string name) {
Thrd = new Thread(this.Run);
Thrd.Name = name;
Thrd.Start();
}
// АҒЫНҒА кіру нүктесі.
void Run() {
try {
Console.WriteLine(Thrd.Name + " басталды.");
for (int i = 1; i <= 1000; i++) {
Console.Write(i + " ");
if((i%10)==0) {
Console.WriteLine();
Thread.Sleep(250);
}
}
Console.WriteLine(Thrd.Name + " қалыпты аяқталды.");
} catch(ThreadAbortException exc) {
Console.WriteLine("Ағын тоқтатылды, аяқтау коды " + exc.ExceptionState);
} }
}
class UseAltAbort {
static void Main() {
MyThread mt1 = new MyThread("Ағын 1");
Thread.Sleep(1000); // туынды ағынның жұмысына рұқсат беру
Console.WriteLine("Ағынды тоқтату.");
mt1.Thrd.Abort (100);
mt1.Thrd.Join(); // ағын тоқтатылуын күту
Console.WriteLine("Негізгі ағын тоқтатылды.");
}}

```

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.