

## Лабораториялық жұмыс 12. Асинхронды функцияларды құру.

**Лабораториялық жұмыстың мақсаты:** Студенттерде асинхронды функцияларда асинхронды шаблондарды пайдалану дағдыларын қалыптастыру.

Лабораториялық жұмысты орындау нәтижесінде студенттер келесі қабілеттерге ие болады:

- Асинхронды шаблондарды ажырату.
- Асинхронды шаблондарды пайдалану.

**Тапсырма:** Төменде берілген мысалдармен танысып, нұсқа бойынша берілген есептерде асинхронды шаблондардың бірін жүзеге асырыңыз.

### Cancellation шаблонны

```
class CancellationToken
{
public bool IsCancellationRequested { get; private set; }
public void Cancel() { IsCancellationRequested = true; }
public void ThrowIfCancellationRequested()
{
if (IsCancellationRequested)
throw new OperationCanceledException();
}
}
Жұмысын тоқтатуға болатын асинхронды әдіс:
async Task Foo (CancellationToken cancellationToken)
{
for (int i = 0; i < 10; i++)
{
Console.WriteLine (i);
await Task.Delay (1000);
cancellationToken.ThrowIfCancellationRequested();
}
}
```

### Тапсырма комбинаторлары

#### WhenAny

```
Task<int> winningTask = await Task.WhenAny (Delay1(), Delay2(), Delay3());
Console.WriteLine ("Done");
Console.WriteLine (winningTask.Result); // 1
```

#### WhenAll

```
await Task.WhenAll (Delay1(), Delay2(), Delay3());
Task<int> task1 = Task.Run (() => 1);
Task<int> task2 = Task.Run (() => 2);
int[] results = await Task.WhenAll (task1, task2); // { 1, 2 }
```

```
async Task<int> GetTotalSize (string[] uris)
{
    IEnumerable<Task<byte[]>> downloadTasks = uris.Select (uri =>
new WebClient().DownloadDataTaskAsync (uri));
byte[][] contents = await Task.WhenAll (downloadTasks);
return contents.Sum (c => c.Length);
}
}
```

## 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.