



BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

İŞLETİM SİSTEMLERİ

THE DINING SAVAGES

PROBLEM

30.12.2020

Tuğçe ARAS 18253509

Feyza EdaTÜRK 17253058

Yuşah Yağız ÖZTÜRK 17253074

Yağmur Selin ŞENTÜRK 18253052

## Problem Tanımı:

Vahşiler kabilesi , M porsiyon büyüklükte bir tencereden ortak akşam yemeklerini yiyorlar. Tencere boş olmadığı sürece vahşi yemek yiyebiliyor. Ancak tencere boşsa aşçı uyandırılıyor ve vahşiler tencerenin dolmasını bekliyor.

## Main Sınıfı:

```
package dining_savages_semaphore;
import java.util.concurrent.Semaphore;
public class Main {
    public static void main(String[] args) {

        int m = 3; // porsiyon adedimiz
        Semaphore eat = new Semaphore(0);
        Semaphore mutex = new Semaphore(1);
        Semaphore yeniden_doldurma = new Semaphore(0);

        Tencere tencere = new Tencere(m);
        Cook cook = new Cook(yeniden_doldurma, eat, tencere);

        // 6 adet vahşimiz bulunmakta.
        Savage s0 = new Savage(0, mutex, eat, yeniden_doldurma, tencere);
        Savage s1 = new Savage(1, mutex, eat, yeniden_doldurma, tencere);
        Savage s2 = new Savage(2, mutex, eat, yeniden_doldurma, tencere);
        Savage s3 = new Savage(3, mutex, eat, yeniden_doldurma, tencere);
        Savage s4 = new Savage(4, mutex, eat, yeniden_doldurma, tencere);
        Savage s5 = new Savage(5, mutex, eat, yeniden_doldurma, tencere);
```

```
        cook.start();
        s0.start();
        s1.start();
        s2.start();
        s3.start();
        s4.start();
        s5.start();
    }
}
```

## Cook Sınıfı:

```
package dining_savages_semaphore; import
java.util.concurrent.Semaphore;
```

```
public class Cook extends Thread {
private Semaphore yeniden_doldurma, eat; private Tencere
tencere;
```

```
public Cook(Semaphore yeniden_doldurma, Semaphore eat, Tencere tencere){
this.yeniden_doldurma = yeniden_doldurma;
this.eat = eat; this.tencere = tencere;
}
```

```
// Tencerede yemek kalmazsa aşçının uyandırılıp yemek yapması gereken kısım public void
yeniden_doldurma(int m) {
System.out.println("Aşçı uyandı!"); this.tencere.yeniden_doldurma(m);
System.out.println("Tencere tekrar yiyecek ile dolu.");
}
```

```
// Bu kısımda yeniden_doldurma elde edilip(acquire) tencereye M porsiyon yemek koyulduğu
```

```
// ve en son release ile serbest bırakılır
```

```
public void run() {  
    while(true) {  
        try {  
            this.yeniden_doldurma.acquire();  
            this.yeniden_doldurma(this.tencere.getM());  
            this.eat.release();  
  
        } catch (InterruptedException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

## Savage Sınıfı:

```
package dining_savages_semaphore;
```

```
import java.util.concurrent.Semaphore; import  
java.lang.Thread;
```

```
public class Savage extends Thread{  
    private Semaphore mutex, eat, yeniden_doldurma; private Tencere  
    tencere;  
    private int id;
```

```
    public Savage(int id, Semaphore mutex, Semaphore eat, Semaphore  
    yeniden_doldurma, Tencere tencere){
```

```
        this.id = id;
        this.mutex = mutex;
        this.eat = eat;
        this.yeniden_doldurma = yeniden_doldurma;
        this.tencere = tencere;
    }
```

// Bu kısımda vahşi yemek yedikten sonra kaç porsiyon yemek kaldığını belirten metot vardır.

// En son mutex 'i serbest bırakarak diğer Savage'dan geriye kalan porsiyon yazılır.

```
    public void eat() throws InterruptedException {
        int foodLeft = this.tencere.getFood();
        System.out.println("Vahşi" + this.id + " yemek yiyor. Tencerede " +
            foodLeft + " porsiyon yemek kaldı.");
        this.mutex.release();
        Thread.sleep(1000);
    }
```

```
    public void run() {
        while(true) {
            try {
                this.mutex.acquire();
```

// Eğer tencerede yemek yoksa "Vahşi yemek bulamadı çıktısını verir!"

// Release ile yeniden doldurma serbest bırakılır ve eat tutulur.

```
        if(!this.tencere.hasFood()) {
            System.out.println("Vahşi " + this.id + " yemek bulamadı!");
```

```
                this.yeniden_doldurma.release();
                this.eat.acquire();
            }
        }
```

```
        this.eat();
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}
}
```

## Tencere Sınıfı:

```
package dining_savages_semaphore;
```

```
public class Tencere {
    private int available; // tenceredeki porsiyona göre artan ya da azalan değişkenimiz.
    private int m;
```

```
    public Tencere(int m){
        this.available = 0; this.m = m;
    }
```

```
    public int getFood() {
        return this.available -= 1;
    }
```

```
    public void yeniden_doldurma(int m) {
        this.available+=m;
    }
```

```
    public boolean hasFood() {
        if(this.available>0)
            return true;
```

```
        return false;
    }

    public int getM() {
        return m;
    }

    public int getAvailable() {
        return this.available;
    }
}
```