Агымдардын бирин-бири бөгөттөө абалдары

Бул абал качан, эки агым синхрондоштурулган объекттердин жубунан циклдик көз каранды болгондо пайда болот. Мисалы, бир агым мониторго x объектисин, ал эми башка агым у объектисин киргизди дейли.

Эгерде x, у объектисинин синхрондоштурулган ыкмасын чакырууга аракет кылса, ал бөгөттөлөт. Ошондой эле, у өз кезегинде x объектисинин синхрондоштурулган ыкмасын чакырганга аракет кылса, анда ал ар дайыма күтүү абалында туш келет, анткени x ке жетүү үчүн, биринчи у өзүнүн бөгөтүн бошотушу керек болчу.

Бирин бири бөгөттөө катасын оңдоодо эки себеп кыйынчылыкты туудурат:

1. Ката жалпысынан алганда, эки агымдын убакытын кванттоо интервалдары белгилүү бир катышта болгондо пайда болгондугу үчүн өтө сейрек кездешет.
2. Ал экиден ашык агымды жана синхрондоштурулган объекттерди камтышы мүмкүн. (Башкача айтканда, бөгөттөө жогоруда сүрөттөлгөн окуялардын ырааттуулугу аркылуу келип чыгышы мүмкүн.)

Бөгөттөөнү толук түшүнүү үчүн аны иш жүзүндө карап көрөлү. Төмөнкү мисалда foo() жана bar() ыкмалары менен эки класс (A жана b) түзүлөт.

```
class A {
  synchronized void bir(B b) {
    String name = Thread.currentThread().getName();
    System.out.println(name + " A.kirdi");
    try {
     Thread.sleep(1000);
```

```
    } catch(Exception e) {
      System.out.println(" A toktotuldu");
    }
    System.out.println(name + " B.last() chakyryldy");
    b.last();
   }

   synchronized void last() {
    System.out.println(" A.last ichinde");
   }
  }
class B {
  synchronized void eki(A a) {
    String name = Thread.currentThread().getName();

    System.out.println(name + " B.kirdi");

    try {
     Thread.sleep(1000);
    } catch(Exception e) {
      System.out.println(" B toktotuldu");
    }
```

```
    System.out.println(name + " A.last() ty chakyruuga araket kyluuda");
    a.last();
   }

   synchronized void last() {
    System.out.println(" A.last tyn ichinde");
   }
  }
class Deadlock implements Runnable {
  A a = new A();
  B b = new B();

  Deadlock() {
    Thread.currentThread().setName("MainThread");
    Thread t = new Thread(this, "RacingThread");
    t.start();

    a.bir(b); //block a in this thread
    System.out.println("bashky agymga kaituu");
   }
   public void run() {
    b.eki(a); // block thread b in another thread
    System.out.println("bashka agymga kaituu");
```

```
  }
  public static void main(String args[]) {
   new Deadlock();
  }
}
```

Бул программаны иштеткенде, сиз төмөнкү натыйжаны көрөсүз:

MainThread A.bir кирди

RacingThread B.eki кирди

MainThread B.last() ыкмасын чакырганга аракет кылууда

RacingThread A.last() ыкмасын чакырганга аракет кылууда

Программа бөгөттөлгөндүктөн, программанын ишин токтотуу үчүн <Ctrl>+<C>

баскычтарын басыш керек. B агымынын b монитору бар экенин, ал a объектиси боюнча мониторду күтүп жатканын көрүнөт. Ошол эле учурда, A агымынын a монитору бар жана b нын кабыл алуусун күтүп жатканын көрүүгө болот. Бул программа эч качан аягына чыкпайт.