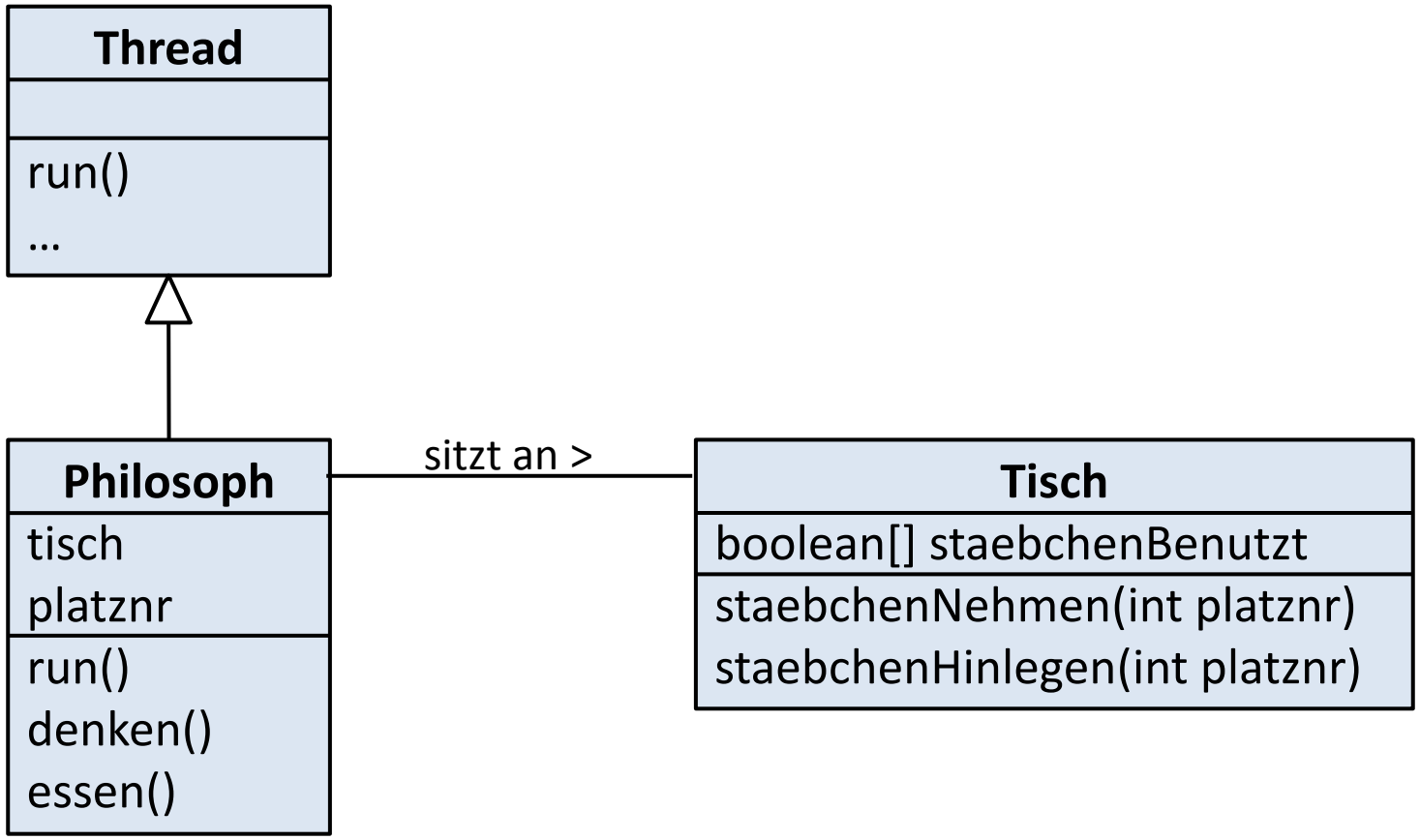
S. 60 / 2 Philosophenproblem



Testklasse erzeugt 5 Philosophen, die an einem Tisch sitzen.
Implementiere zunächst so, dass die Verklemmung auftritt.

```
public void run() {
    while(true){
        denken();
        essen();
    }
}
public void denken(){
    try {
        sleep(500);
    }
    catch (Exception e){}
    System.out.println(Thread.currentThread().
                    getName() + " hat nachgedacht!");
}
```

oder bei denken und essen zufällige Zeit wählen!

```
public void essen(){
    tisch.staebchenNehmen(platznr);
    try {
       sleep(500);
    }
    catch (Exception e){}
    tisch.staebchenNehmen((platznr+1)%5);
    System.out.println(Thread.currentThread().getName() + " isst
                                                          gerade!");
    try {
       sleep(500);
    }
    catch (Exception e){}
    tisch.staebchenHinlegen(platznr);
    tisch.staebchenHinlegen((platznr+1)%5);
    System.out.println(Thread.currentThread().getName() + " hat
                        gegessen und Stäbchen wieder hingelegt!!");
  }
```

```
 public Tisch(){
    staebchenBenutzt = new boolean[5];
    for (int i=0; i<5; i++) {
       staebchenBenutzt[i] = false;
    }
 }

public void staebchenNehmen(int platznr){
     System.out.println(Thread.currentThread().
                  getName()+ " will Stäbchen nehmen!");
     while (staebchenBenutzt[platznr]){}
     staebchenBenutzt[platznr] = true;
  }
 public void staebchenHinlegen(int platznr){
       staebchenBenutzt[platznr] = false;
    }
```

Nun ohne Verklemmung:

- Wandle die Methode essen() so ab, dass stäbchenNehmen(…) bzw. stäbchenHinlegen(…) beide Stäbchen betrifft.
- Synchronisiere diese beiden Methoden.
- Ist nun gewährleistet, dass kein Philosoph verhungert?

```
public void essen(){
        tisch.staebchenNehmen(platznr);
        System.out.println(Thread.currentThread().getName()
                                                + " isst gerade!");
        try {
            sleep(500);
        }
        catch (Exception e){}
        tisch.staebchenHinlegen(platznr);
        System.out.println(Thread.currentThread().getName()
                                    + " hat gegessen und Stäbchen
                                    wieder hingelegt!!");
    }
```

```
public synchronized void staebchenNehmen(int platznr){
    System.out.println(Thread.currentThread().getName()+
          " will essen!");
     while (staebchenBenutzt[platznr]
                              ||staebchenBenutzt[(platznr+1)%5]){
        try {
           System.out.println(Thread.currentThread().
                    getName() + ": mindestens eines der
                    Stäbchen ist belegt!");
           wait();
        }
        catch (InterruptedException e){}
     }
     staebchenBenutzt[platznr] = true;
     staebchenBenutzt[(platznr+1)%5] = true;
     notify();
  }
```

```
public synchronized void staebchenHinlegen(int platznr){
        staebchenBenutzt[platznr] = false;
        staebchenBenutzt[(platznr+1)%5] = false;
        notify();
    }
```