HTW Chur
Hochschule für Technik und Wirtschaft

Fachhochschule Ostschweiz
University of Applied Sciences

Readers/Writers

Start

```
// File Data.java
public class Data {
    Data(int size) {
      value = new int[size];
    }
    int[] value;
}

// File Writer.java
    while (true) {
      for (int j=0; j < data.value.length; j++) {
        data.value[j] = random.nextInt();
      }
    }

// File Reader.java
    while (true) {
      StringBuffer buf = new StringBuffer();
      for (int j=0; j < data.value.length; j++) {
        buf.append(data.value[j]);
        buf.append(", ");
      }
      System.out.println(buf.toString());
    }
```

Step 1

```
// File Data.java
public class Data {
    Data(int size) {
      value = new int[size];
    }
    int[] value;
}

// File Writer.java
    while (true) {
      synchronized(data) {
        for (int j=0; j < data.value.length; j++) {
          data.value[j] = random.nextInt();
        }
      }
    }

// File Reader.java
    while (true) {
      StringBuffer buf = new StringBuffer();
      synchronized(data) {
        for (int j=0; j < data.value.length; j++) {
          buf.append(data.value[j]);
          buf.append(", ");
        }
      }
      System.out.println(buf.toString());
    }
```

Step 2 - 1

```
// File Data.java
public class Data {
    Data(int size) {
      // ...
    }
    int[] value;
    int noReaders = 0;
    int noWriters = 0;
}

// File Writer.java
    while (true) {
        while(data.noReaders != 0
                  || data.noWriters != 0) {
          // do nothing
        }
        data.noWriters++;
        for (int j=0; j < data.value.length; j++) {
          data.value[j] = random.nextInt();
        }
        data.noWriters--;
      }
    }

// ...
```

Step 2 - 2

```
// File Reader.java
   while (true) {
     StringBuffer buf = new StringBuffer();
     while (data.noWriters != 0) {
       // do nothing
     }
     data.noReaders++;
     for (int j=0; j < data.value.length; j++) {
       buf.append(data.value[j]);
       buf.append(", ");
     }
     data.noReaders--;
     System.out.println(buf.toString());
   }
```

Step 3 - 1

```
// File Data.java
public class Data {
    Data(int size) {
      // ...
    }
    int[] value;
    int noReaders = 0;
    int noWriters = 0;
}

// File Writer.java
    while (true) {
      synchronized (data) {
        while(data.noReaders != 0
                  || data.noWriters != 0) {
          // do nothing
        }
        data.noWriters++;
      }
      for (int j=0; j < data.value.length; j++) {
        data.value[j] = random.nextInt();
      }
      data.noWriters--;
    }

// ...
```

Step 3 - 2

```
// File Reader.java
   while (true) {
     StringBuffer buf = new StringBuffer();
     synchronized(data) {
       while (data.noWriters != 0) {
         // do nothing
       }
       data.noReaders++;
     }
     for (int j=0; j < data.value.length; j++) {
       buf.append(data.value[j]);
       buf.append(", ");
     }
     data.noReaders--;
     System.out.println(buf.toString());
   }
```

Step 4 - 1

```
// File Data.java
public class Data {
    Data(int size) {
      // ...
    }
    int[] value;
    int noReaders = 0;
    int noWriters = 0;

    synchronized void requestRead()
                  throws InterruptedException {
      while (noWriters != 0) {
        wait();
      }
      noReaders++;
    }

    synchronized void releaseRead() {
      noReaders--;
      if (noReaders == 0) {
        notifyAll();
      }
    }

    // ...

}
```

Step 4 - 2

```
// ... File Data.java
public class Data {
    // ...
    synchronized void requestWrite() throws
    InterruptedException {
      while (noReaders != 0 || noWriters != 0) {
        wait();
      }
      noWriters++;
    }

    synchronized void releaseWrite() {
      noWriters--;
      if (noWriters == 0) {
        notifyAll();
      }
    }

}

// File Writer.java
    while (true) {
      data.requestWrite();
      for (int j=0; j < data.value.length; j++) {
        data.value[j] = random.nextInt();
      }
      data.releaseWrite();
    }
```

Step 4 - 3

```
// File Reader.java
   while (true) {
     StringBuffer buf = new StringBuffer();
     data.requestRead();
     for (int j=0; j < data.value.length; j++) {
       buf.append(data.value[j]);
       buf.append(", ");
     }
     data.releaseRead();
     System.out.println(buf.toString());
   }
```

Step 5 - 1 (ab Java 5.0)

```
// File Data.java
public class Data {
    Data(int size) {
      value = new int[size];
      lock = new ReentrantReadWriteLock();
    }
    int[] value;
    ReentrantReadWriteLock lock;
}

// File Writer.java
    while (true) {
      data.lock.writeLock().lock();
      try {
        for (int j=0; j < data.value.length; j++) {
          data.value[j] = random.nextInt();
        }
      finally {
        data.lock.writeLock().unlock();
      }
    }
```

Step 5 - 2 (ab Java 5.0)

```
// File Reader.java
   while (true) {
     StringBuffer buf = new StringBuffer();
     data.lock.readLock().lock();
     try {
       for (int j=0; j < data.value.length; j++) {
         buf.append(data.value[j]);
         buf.append(", ");
       }
     } finally {
       data.lock.readLock().unlock();
     }
     System.out.println(buf.toString());
   }
```