

On the effective use of abstraction

Anthony Arkles

aja042@mail.usask.ca

December 2nd, 2009

Abstraction?

- Simplifying a concept, hiding implementation details.

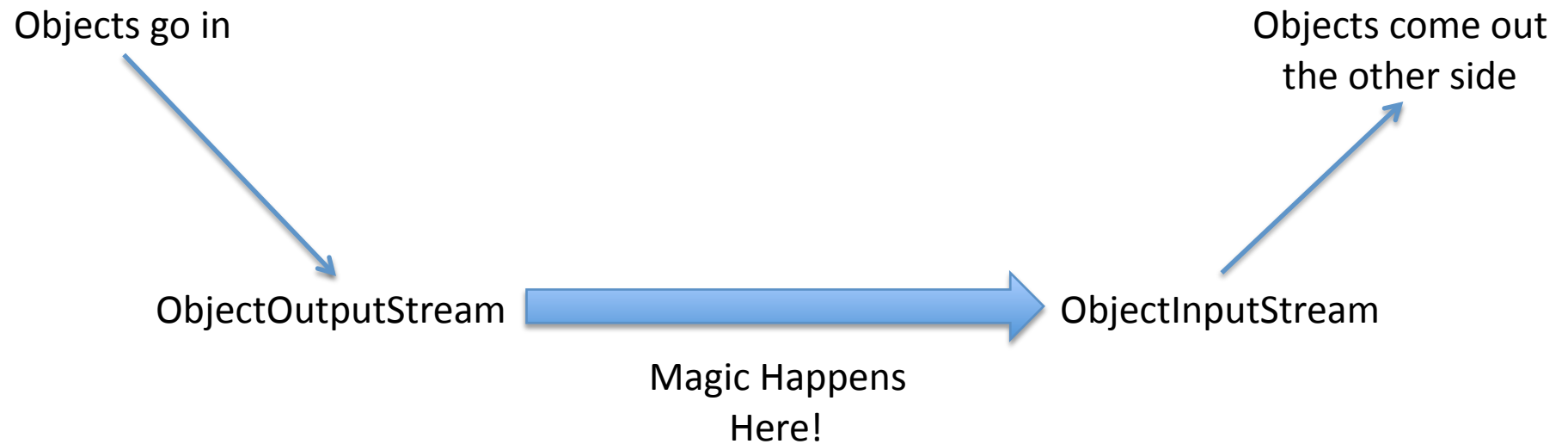
Quantum Mechanics
Dopant Concentrations
Transistors
Logic Gates
Sequential Logic
Logic Modules
Processors
Operating System Kernel
Standard Libraries
Java Virtual Machine
Java Bytecode
Java Standard Library
Java Source Code

You probably don't want
to be writing software up
here



You are here

Java Object Streams



The Server

```
ObjectOutputStream toClient =  
    ...  
Boolean allowed;  
// figure out if the client is  
    allowed  
  
toClient.writeBoolean(allowed)  
if(allowed) {  
    toClient.writeObject(room);  
    toClient.writeObject(msgs);  
    ...  
}
```

The Client

```
ObjectInputStream server = ...  
  
allowed = server.readBoolean()  
if(allowed) {  
    room = server.readObject()  
    msgs = server.readObject()  
    ...  
} else {  
    System.out.println("Error")  
}
```

The Server

```
ObjectOutputStream toClient =
```

```
...
```

```
Boolean allowed;
```

```
// figure out if the client is  
    allowed
```

TRUE

```
toClient.writeBoolean(allowed)
```

```
if(allowed) {
```

```
    toClient.writeObject(room);
```

```
    toClient.writeObject(msgs);
```

```
...
```

```
}
```

Execution makes it here fine

The Client

```
ObjectInputStream server = ...
```

```
allowed = server.readBoolean()
```

```
if(allowed) {
```

```
    room = server.readObject()
```

```
    msgs = server.readObject()
```

```
...
```

```
} else {
```

```
    System.out.println("Error")
```

```
}
```

Execution makes it here fine

The Server

```
ObjectOutputStream toClient =
```

```
...
```

```
Boolean allowed;
```

```
// figure out if the client is  
    allowed
```

```
FALSE
```

```
toClient.writeBoolean(allowed)
```

```
if(allowed) {
```

```
    toClient.writeObject(room);
```

```
    toClient.writeObject(msgs);
```

```
...
```

```
}
```

```
Execution makes it here fine
```

The Client

```
ObjectInputStream server = ...
```

```
allowed = server.readBoolean()
```

```
Execution hangs here
```

```
    room = server.readObject()
```

```
    msgs = server.readObject()
```

```
...
```

```
} else {
```

```
    System.out.println("Error")
```

```
}
```

Buffered IO

- `ObjectOutputStream` is operating with a `Socket`
- Sockets have buffering
- If there isn't enough data, buffers need to be manually flushed

The Server

```
ObjectOutputStream toClient =
```

```
...
```

```
Vector msgs = new Vector();
```

```
toClient.writeObject(msgs);
```

```
// do some other code that  
    adds some messages to the  
    msgs Vector
```

```
toClient.writeObject(msgs);
```

The Client

```
ObjectInputStream server = ...
```

```
Vector msgs;
```

```
// receive the list of messages  
msgs = server.readObject();
```

```
// do some other stuff
```

```
// receive new list of messages  
msgs = server.readObject();
```

The Server

```
ObjectOutputStream toClient =
```

```
...
```

```
Vector msgs = new Vector();
```

Empty vector the first time

```
toClient.writeObject(msgs);
```

```
// do some other code that  
    adds some messages to the  
    msgs Vector
```

Non-empty Vector the second time

```
toClient.writeObject(msgs);
```

The Client

```
ObjectInputStream server = ...
```

```
Vector msgs;
```

```
// receive the list of messages
```

```
msgs = server.readObject();
```

Empty Vector comes out

```
// do some other stuff
```

```
// receive new list of messages
```

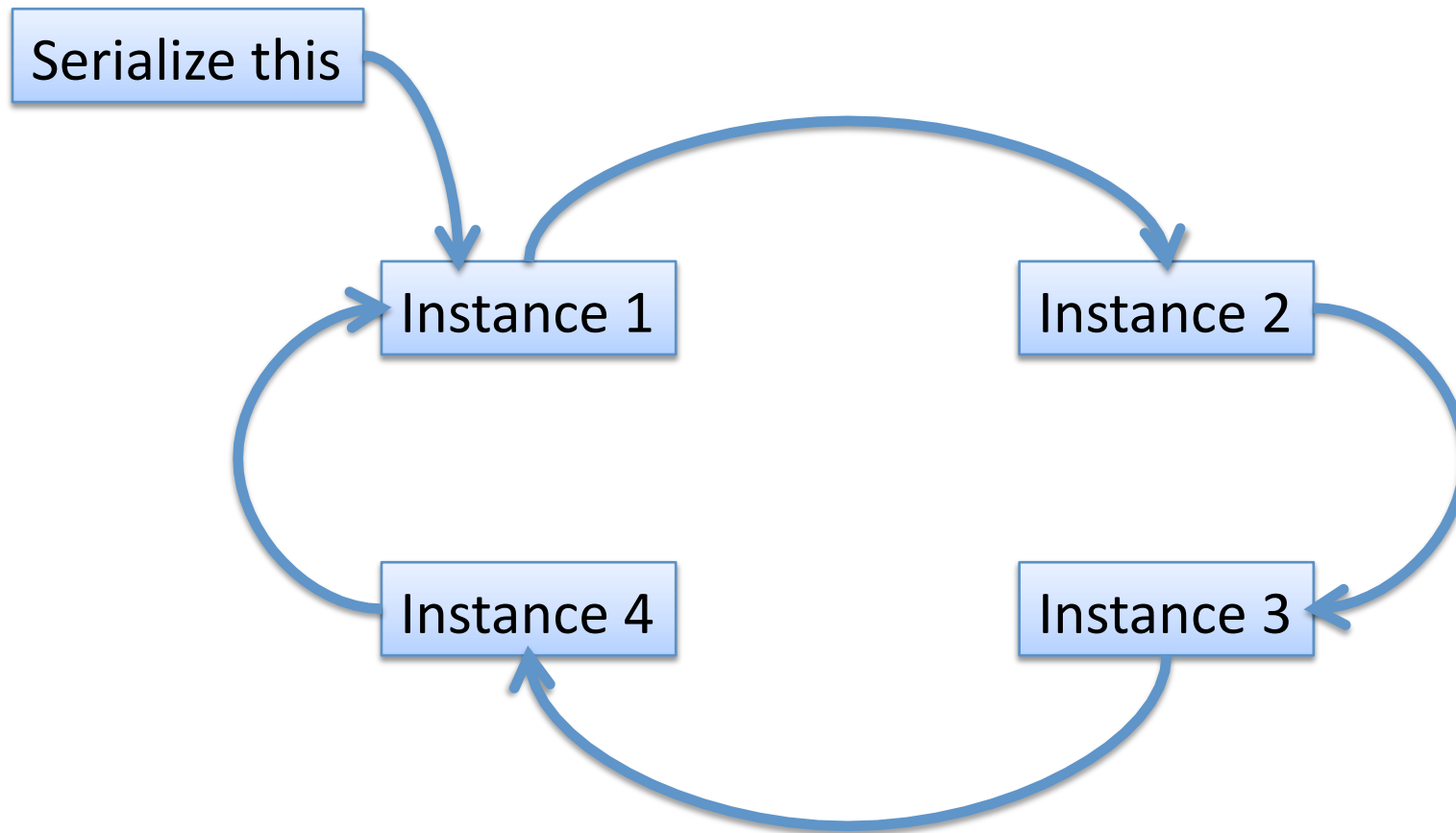
```
msgs = server.readObject();
```

Another empty Vector comes out?!

Java Object Serialization

- Convert an object instance to/from a bytestream (using introspection)
- If an instance has references to other instances, the other instances are serialized too.
- How?

Cyclic References



So why the empty Vector?

- ObjectOutputStream keeps a single Serializer for the whole instance lifetime
- The Vector has the same id both times, so the second time it only sends the address
- The deserializer on the other end just returns the same empty Vector again...

More Examples

- HTTP Caching
- Python threading
- “Why can’t I get a packet to Japan in less than 28ms?”

Take-home Message

If you don't want to spend a lot of time debugging subtle bugs, take the time to understand how abstractions work underneath.