Notes - 11/7

Monday, November 07, 2011 1:13 PM

- Construct
  - Remove constructor signatures
- Constructor functions
  - Want to break things down to constituent pieces

```
Interface Point {
}
Function Point(x, y) : Point {}
Class X extends Point implements Point requries Point

Class Printable requires Document {
 function print() {
  var doc = getText();
 }
}

class Document {
 getText() …
}

Class MyDocument extends Document, Printable {

}

//or

Class MyDocument extends Printable { // requires of an extends become implments

}

// This is a class
Interface Point requires … {
}
Function Point(x, y) : Point {}
```

- Extern classes vs. regular classes
-
- Functions as any
- Mixins
- Brands

- Private methods
- Enums
  - Treated as a primitive type
  - Branded primitive

- No construct signatures
- Speculate that only allow construct on 'new'

- Functions don't infer to return void
  - But void annotation is okay for "return;" or no returns

- Constructors are gone
- Brands are gone
- Enums are branded members
- Extends/implements/requires
  - If you have a requires you cannot be newed
- New can take any or any function type