

Notes - 1/21: SENT

Saturday, March 12, 2011 5:41 PM

Possible topics:

- var->interface typing
- Continued review of expression typing examples
- Object literals and typed object literals
- Close on overloading
- We think we're at statically resolved with mangling
 - Type name resolution at global scope
 - Function coercion rules
 - Varargs?
 - Typing of ES5 builtins (Strada core library) and Strada language requirements
 - First class types
 - Inheriting from JS objects (dynamically dispatched inheritance?)
 - Arrays
 - INumber -> double coercions and similar
 - Separate compilation
- Mangling?

Status:

- Get prototype running again
- Get working branch for collaboration on front end
- Focus non single file for now

Interface typing:

- Interface erases to var at runtime (in the perf box)

```
interface IBar
{
    function int foo(int i);
}

class Bar
{
    function int foo(int i) {...}
}

IBar bar = new Bar(); // case 0
bar.foo("3");          // case 1
bar.foo();              // case 2
bar.foo(3);             // case 3
```

```
bar.other("hello", 3); // case 4
```

My belief is that:

1. Bar is structurally compatible with IBar, even without interface declaration
2. Case 1 fails to type check. IBar has a `foo` method, and that method takes an `int`, and "3" is not implicitly coercible to `int`.
3. Case2 fails to type check. IBar had a `foo` method, and that method takes 1 argument.
4. Case 3 succeeds.
5. Case 4 fails

Aside:

- Code gen goal is "debuggable" and "readable", not "would have been written by a human"