Notes - 11/18/2013

Monday, November 18, 2013 10:01 AM

- News
- Modeling non-inherited static members on JavaScript libraries (Bill 11/16, attached)
  - Proposal is to not inherit the static members added to a symbol by a module, rather only the static members inside of the class are inherited by subclasses
  - Issues:
    - ○ The __extends function currently is general case, but would copy over non-inherited members.
      - One solution would be to copy specific members by hand
      - Or, we could just copy them over and only make them visible at design team. This is preferred
    - ○ Having two separate types, one for C and one for the static 'this' inside of C being different adds complication to the type checking

```
class C {
      static foo(): string;
      static baz() {
            this.foo();
      }
}

class D extends C { }

module D {
      foo(): number;  //<-- error at compile time
}

class E extends D {
      bar() {
            return this.foo();
      }
}
```

- Generic classes with constraints unable to extend non-generic classes (Dan 11/15, attached)
  - Related, should this be allowed?
    ```
    class EEvent {
       target: Object;
    }

    class MyEvent extends EEvent {
       target: any;
    }
    ```
  - Two solutions:
    - ○ Stick with 'any' given to the prototypes. And then, we use assignment compatibility check instead of subtype check when a class subtyping via 'extends' and 'implements' and for index checks.
    - ○ Give an existential type for constrained types, and then use that for the prototype

- Rules for recursive/circular type inference
  - Return type inference (Anders 11/13, attached)
  - Need a spec that describes what constructs are allowed to be used in a class that references itself or else the type becomes 'any'. These would then be errors with no-implicit-any.
- Infinitely expanding type reference when Generic Type derives from Non-Generic type (Paul 11/14, attached)
  - Likely a bug in implementation. Could not discover where infinitely expanding type reference occurs
- Can't subtract dates. [820486]
  - Fine for a 1.1, we can improve these as we go forward
  - Action: reserve the names now for date, regex, and error
- Closing on: should all generic types be covariant? [813197] [808522]
  -