and then executes the instance member variable initializers, if any.

8.4 Property Member Declarations

Property member declarations can be member variable declarations, member function declarations, or member accessor declarations.

PropertyMemberDeclaration:
  MemberVariableDeclaration
  MemberFunctionDeclaration
  MemberAccessorDeclaration

Member declarations without a `static` modifier are called instance member declarations. Instance property member declarations declare properties in the class instance type (section 8.2.4), and must specify names that are unique among all instance property member and parameter property declarations in the containing class, with the exception that instance get and set accessor declarations may pairwise specify the same name.

Member declarations with a `static` modifier are called static member declarations. Static property member declarations declare properties in the constructor function type (section 8.2.5), and must specify names that are unique among all static property member declarations in the containing class, with the exception that static get and set accessor declarations may pairwise specify the same name.

Note that the declaration spaces of instance and static property members are separate. Thus, it is possible to have instance and static property members with the same name.

Except for overrides, as described in section 8.2.3, it is an error for a derived class to declare a property member with the same name and kind (instance or static) as a base class member.

Every class automatically contains a static property member named 'prototype', the type of which is an instantiation of the class type with type Any supplied as a type argument for each type parameter. It is an error to explicitly declare a static property member with the name 'prototype'.

Below is an example of a class containing both instance and static property member declarations:

```
class Point {
    constructor(public x: number, public y: number) { }
    public distance(p: Point) {
        var dx = this.x - p.x;
        var dy = this.y - p.y;
        return Math.sqrt(dx * dx + dy * dy);
    }
    static origin = new Point(0, 0);
    static distance(p1: Point, p2: Point) { return p1.distance(p2); }
}
```

The class instance type 'Point' has the members: