Note that it is seldom meaningful to include a string index signature in a class because it constrains all instance properties of the class. However, numeric index signatures can be useful to control the element type when a class is used in an array-like manner.

8.6 Code Generation

This section describes the structure of the JavaScript code generated from TypeScript classes.

8.6.1 Classes Without Extends Clauses

A class with no `extends` clause generates JavaScript equivalent to the following:

```
var <ClassName> = (function () {
    function <ClassName>(<ConstructorParameters>) {
        <DefaultValueAssignments>
        <ParameterPropertyAssignments>
        <MemberVariableAssignments>
        <ConstructorStatements>
    }
    <MemberFunctionStatements>
    <StaticVariableAssignments>
    return <ClassName>;
})();
```

ClassName is the name of the class.

ConstructorParameters is a comma separated list of the constructor's parameter names.

DefaultValueAssignments is a sequence of default property value assignments corresponding to those generated for a regular function declaration, as described in section 6.5.

ParameterPropertyAssignments is a sequence of assignments, one for each parameter property declaration in the constructor, in order they are declared, of the form

```
this.<ParameterName> = <ParameterName>;
```

where *ParameterName* is the name of a parameter property.

MemberVariableAssignments is a sequence of assignments, one for each instance member variable declaration with an initializer, in the order they are declared, of the form

```
this.<MemberName> = <InitializerExpression>;
```

where *MemberName* is the name of the member variable and *InitializerExpression* is the code generated for the initializer expression.

ConstructorStatements is the code generated for the statements specified in the constructor body.