訂正　実行例 5.7（配布資料 49 ページ）の Student を次のように変えてもよい。

```
1 function Student2(n, id, y, m, d){
2   Person2.call(this, n, y, m, d);
3   this.id    = id;
4 }
5 Student2.prototype = new Person2();
6 Student2.prototype.constructor = Student2;
```

- Person2 内で定義されているメンバー変数をそのまま利用するために、2 行目で Person2 を呼び出している。
- 単純に呼び出すのでは Person2 内での this が Person2 になる。
- そこで Object のメソッド call を用いると、第 1 番目の引数をそのオブジェクト内での this を書き直すことが可能となる。

このリストでは Person2 内で this を出力している。

class について　ECMAScript は 6 から class によりクラスの宣言が可能となった。今までの例を class で書き直すと次のようになる。

```
1 class Person2{
2   constructor(name, y, m, d){
3     this.name = name;
4     this.year = y,
5     this.month = m,
6     this.day = d;
7   }
8   toString(){
9       return "私の名前は"+this.name+"です";
10    };
11  get age(){
12      var Now = new Date();
13      var Age = Now.getFullYear() - this.year;
14      if((Now.getMonth()+1) < this.month) {
15        Age--;
16      } else {
17        if((Now.getMonth()+1) == this.month &&
18           Now.getDate() < this.day) Age--;
19      }
20      return Age;
21    };
```

```
22   get birthday() {
23       return this.year+"年"+this.month+"月"+this.day+"日";
24     };
25 }
26
27 class Student2 extends Person2{
28   constructor(n, id, y, m, d){
29     super(n, y, m, d);
30     this.id    = id;
31   }
32 }
```

- クラスの宣言は class を用いる (1 行目)。
- コンストラクターは constructor 関数で定義される (2 行目から 7 行目)。
- class 内ではメソッドを宣言する。ここでは 3 つのメソッドが定義されている。
- 27 行目から 32 行目は継承の例である。
- クラス名の後にキーワード extends をつけて親クラスを指定する。親クラスを複数指定する多重継承はできない。
- 29 行目では親クラスのコンストラクタを呼び出すために super を利用している。