

3. DECORATOR PATTERN

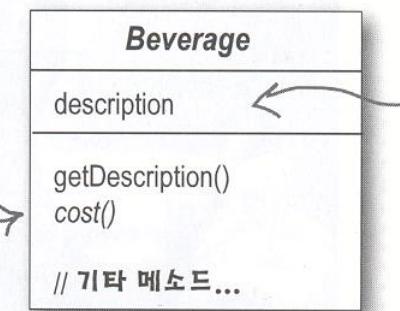
동의대학교 컴퓨터소프트웨어공학과

이종민 교수

스타벅스 기본 음료 클래스 (basic)

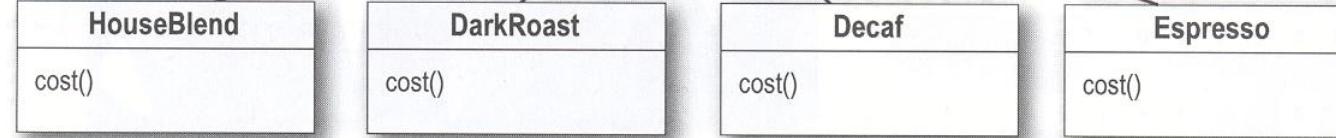
Beverage는 음료를 나타내는 추상 클래스이며, 커피숍에서 판매되는 모든 음료는 이 클래스의 서브클래스가 됩니다.

cost() 메소드는 추상 메소드입니다. 서브클래스에서 그 메소드를 구현해서 새로 정의해야 합니다.



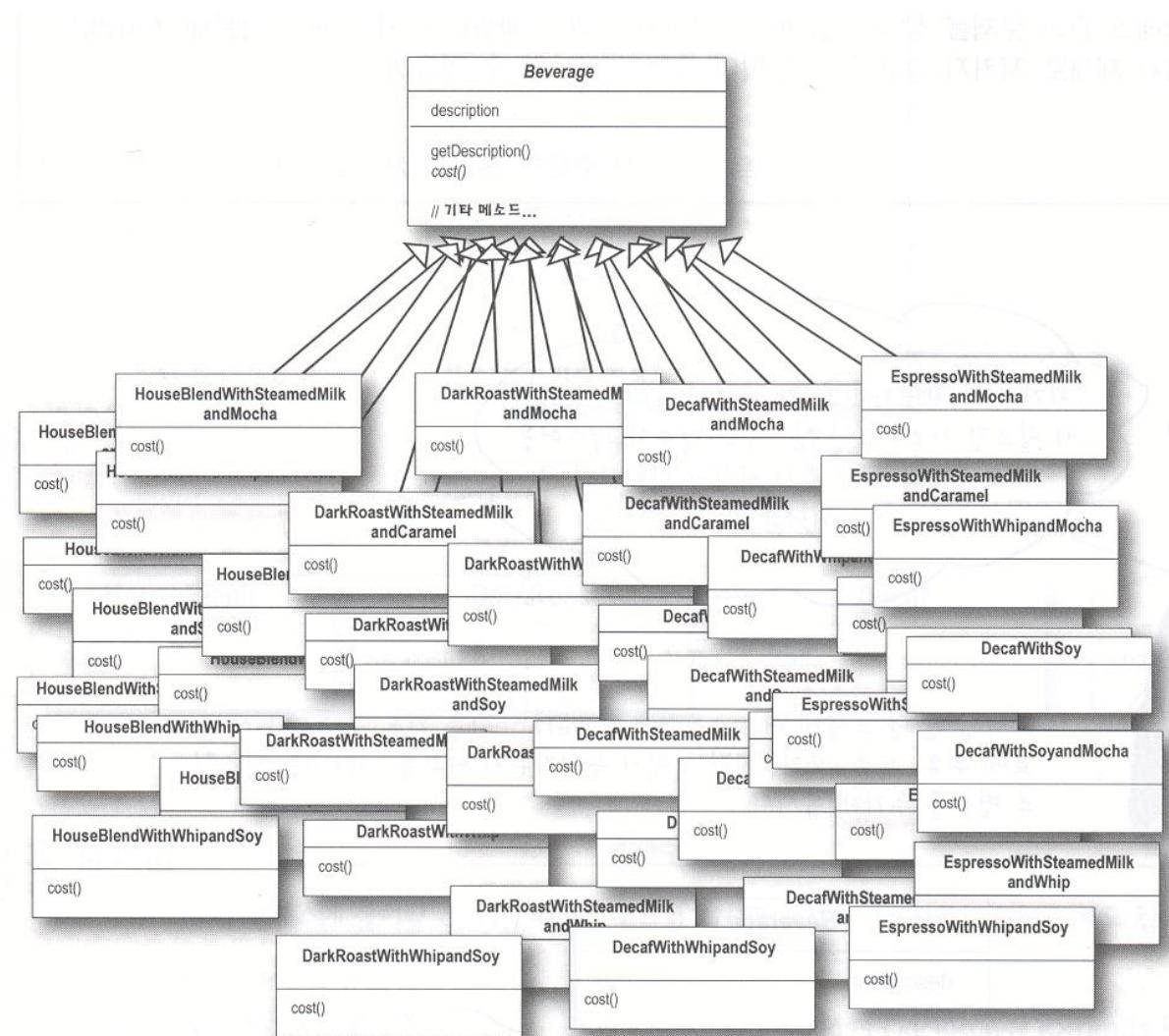
description이라는 인스턴스 변수는 각 서브클래스에서 설정되며, 여기에는 “가장 흙 풍한 다크 로스트 커피” 같은 음료에 대한 설명이 저장됩니다.

description 변수에 저장된 내용은 getDescription() 메소드를 호출하여 알 수 있습니다.

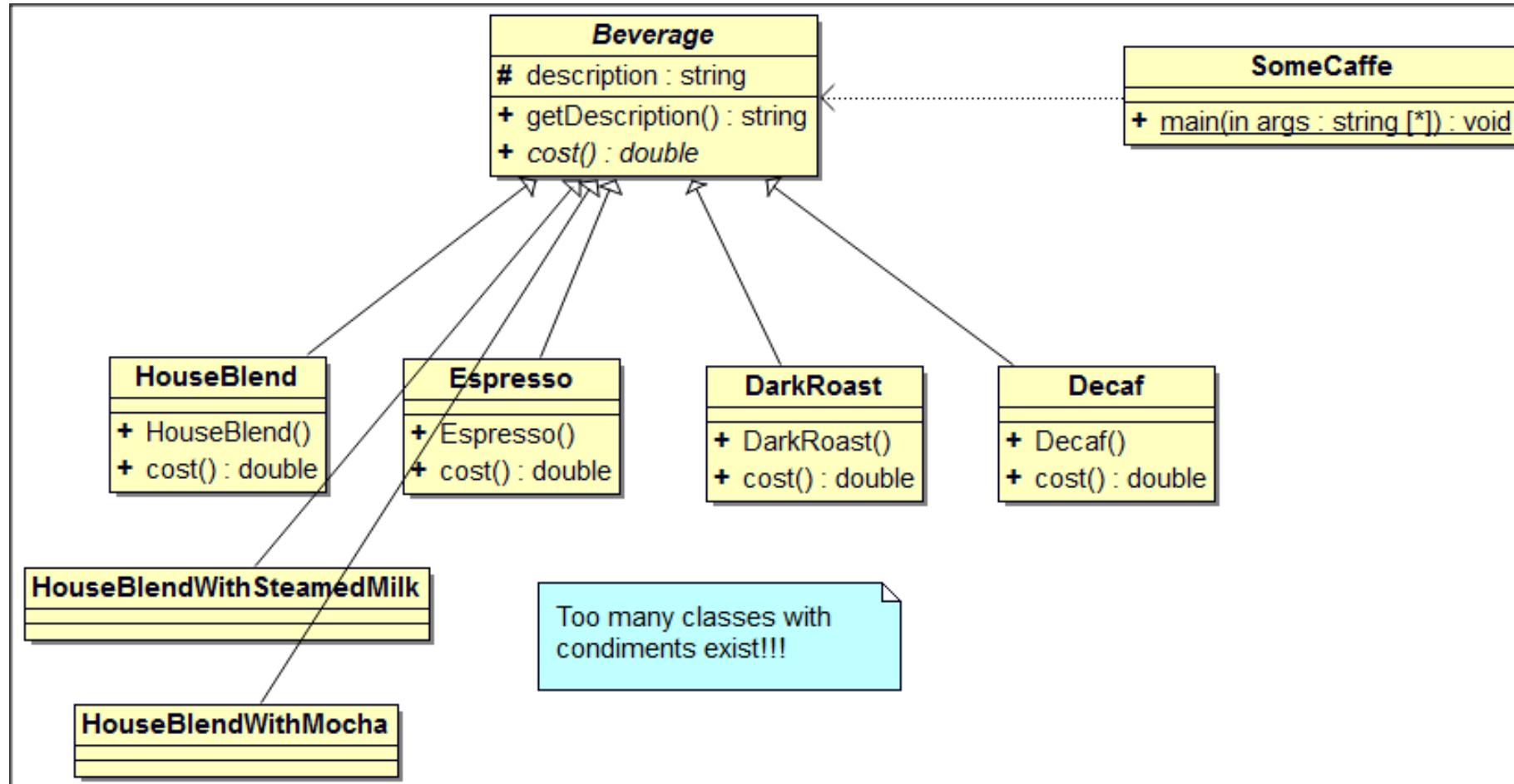


모든 서브클래스에서 음료의 가격을 리턴하는 cost() 메소드를 구현해야 합니다.

추가 항목을 고려한 클래스



실습용 클래스 다이어그램(basic)



Beverage.java

```
6  package cse.design_pattern.ch03.basic;
7
8  public abstract class Beverage {
9
10     protected String description;
11
12
13     public String getDescription() {
14         return description;
15     }
16
17
18     public abstract double cost();
19
20 }
```

HouseBlend.java

```
6  package cse.design_pattern.ch03.basic;
7
8   /* ...4 lines */
12  public class HouseBlend extends Beverage {
13
14       public HouseBlend() {
15      description = "House Blend Coffee";
16  }
17
18       @Override
19       public double cost() {
20      return .89;
21  }
22
23 }
```

Espresso.java

```
6 package cse.design_pattern.ch03.basic;
7
8 /* ...4 lines */
12 public class Espresso extends Beverage {
13
14     public Espresso() {
15         description = "Espresso";
16     }
17
18     @Override
19     public double cost() {
20         return 1.99;
21     }
22 }
```

HouseBlendWithSteamedMilk.java

```
6  package cse.design_pattern.ch03.basic;
7
8  /**...4 lines */
12 public class HouseBlendWithSteamedMilk extends Beverage {
13
14     public HouseBlendWithSteamedMilk() {
15         description = "House Blend Coffee with Steamed Milk";
16     }
17
18     /**
19      * 우유 추가한 하우스 블렌드 커피 가격
20      * 새로운 첨가물이 있을 때마다 새로운 클래스를 만들어야 합니다!
21      *
22      * @return 우유 값 0.1 추가
23      */
24     @Override
25     ① public double cost() {
26         return .89 + .1;
27     }
28
29 }
```

HouseBlendWithMocha.java

```
6  package cse.design_pattern.ch03.basic;
7
8  /* ...4 lines */
12 public class HouseBlendWithMocha extends Beverage {
13
14     public HouseBlendWithMocha() {
15         description = "House Blend Coffee with Mocha";
16     }
17
18     /**
19      * 모카 추가한 하우스 블렌드 커피 가격
20      * 새로운 첨가물이 있을 때마다 새로운 클래스를 만들어야 합니다!
21      *
22      * @return 모카 값 0.2 추가
23      */
24     @Override
25     public double cost() {
26         return .89 + .2;
27     }
28
29 }
```

SomeCaffe.java

```
6 package cse.design_pattern.ch03.basic;
7
8 /**
9 */
10
11
12 public class SomeCaffe {
13
14 /**
15 */
16
17 public static void main(String[] args) {
18     HouseBlend hb = new HouseBlend();
19     Espresso esp = new Espresso();
20
21     System.out.println("HouseBlend 커피는 얼마인가요?");
22     System.out.println(String.format("HouseBlend는 %.2f입니다.", hb.cost()));
23
24     System.out.println("Espresso 커피는 얼마인가요?");
25     System.out.printf("Espresso %.2f입니다.%n", esp.cost());
```

```
27 |     System.out.println("HouseBlend 커피에 우유를 추가하면 얼마인가요?");  
28 |     System.out.println("HouseBlend 클래스만 있어 "  
29 |         + "새로 HouseBlendWithSteamedMilk 클래스를 만들어야 합니다.");  
30 |     System.out.printf("우유 추가한 HouseBlend는 %.2f입니다.%n",  
31 |         new HouseBlendWithSteamedMilk().cost());  
32 |  
33 |     System.out.println("HouseBlend 커피에 모카를 추가하면 얼마인가요?");  
34 |     System.out.println("HouseBlend 클래스만 있어 "  
35 |         + "새로 HouseBlendWithMocha 클래스를 만들어야 합니다.");  
36 |     System.out.printf("모카 추가한 HouseBlend는 %.2f입니다.%n",  
37 |         new HouseBlendWithMocha().cost());  
38 | }  
39 | }  
40 | }
```

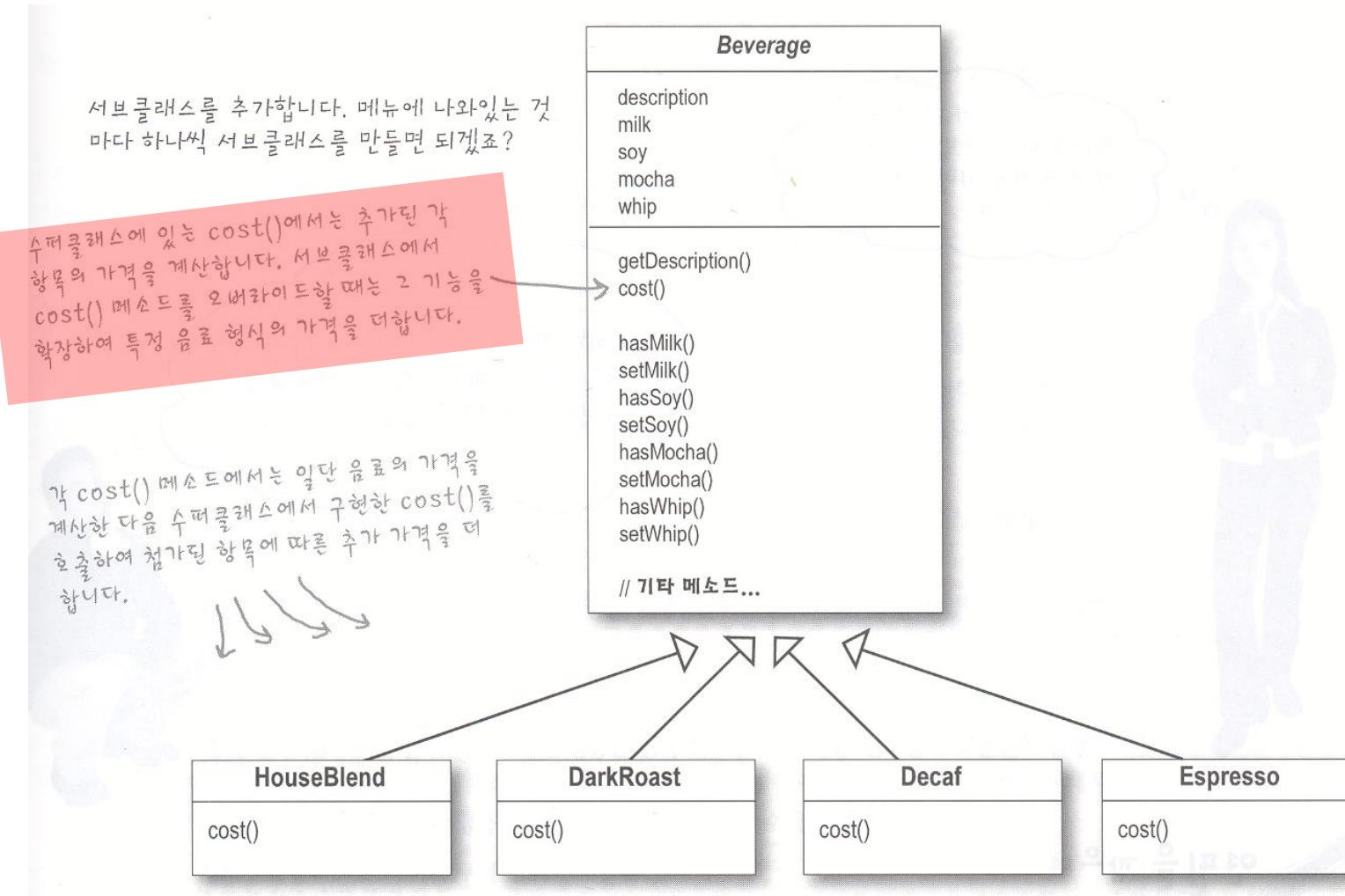
실행 결과

```
--- exec-maven-plugin:1.2.1:exec (default-cli) @ desing_pattern ---
HouseBlend 커피는 얼마인가요?
HouseBlend는 0.89입니다.
Espresso 커피는 얼마인가요?
Espresso 1.99입니다.
HouseBlend 커피에 우유를 추가하면 얼마인가요?
HouseBlend 클래스만 있어 새로 HouseBlendWithSteamedMilk 클래스를 만들어야 합니다.
우유 추가한 HouseBlend는 0.99입니다.
HouseBlend 커피에 모카를 추가하면 얼마인가요?
HouseBlend 클래스만 있어 새로 HouseBlendWithMocha 클래스를 만들어야 합니다.
모카 추가한 HouseBlend는 1.09입니다.
```

문제점

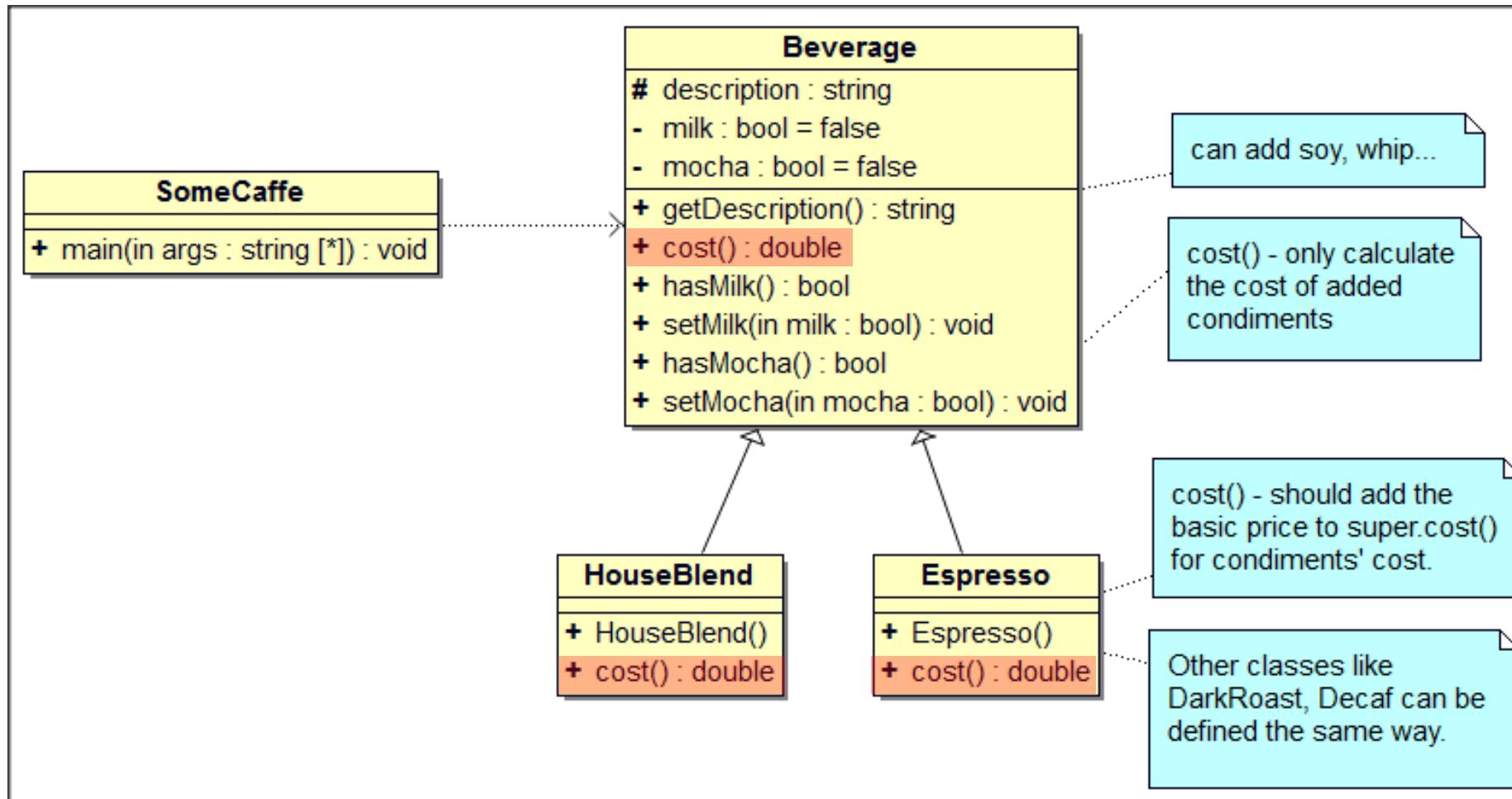
- 첨가물 추가에 따라서 cost() 메서드가 변경되어야 하므로 새로운 클래스 필요
- cost() 메서드 관점에서 중복된 코드 많이 존재

개선 1 (enhancement1)



실습용 클래스 다이어그램

- enhancement01



Beverage.java

```
6  package cse.design_pattern.ch03.enhancement1;
7
8  public abstract class Beverage {
9      protected String description;
10     private boolean milk = false;
11     private boolean mocha = false;
12     // soy, whip 등 필요 항목 더 추가 가능
13
14     public String getDescription() {
15         return description;
16     }
17
18     /**
19      * 추가된 항목의 가격만 계산함.
20      * 서브 클래스에서 cost() 메소드를 오버라이드할 때는
21      * 그 기능을 확장하여 특정 음료 형식의 가격을 더함.
22      *
23      * @return 추가된 항목의 가격
24      */
25
26     public double cost() {
27         double cost = 0.0;
28
29         if (hasMilk()) {
30             cost += 0.1;
31         }
32         if (hasMocha()) {
33             cost += 0.2;
34         }
35         return cost;
36     }
37
38
39
40 }
```

```
42     public boolean hasMilk() {  
43         return milk;  
44     }  
45  
46     public void setMilk(boolean milk) {  
47         this.milk = milk;  
48     }  
49  
50     public boolean hasMocha() {  
51         return mocha;  
52     }  
53  
54     public void setMocha(boolean mocha) {  
55         this.mocha = mocha;  
56     }  
57  
58 }
```

HouseBlend.java

```
6  package cse.design_pattern.ch03.enhancement1;
7
8
9  public class HouseBlend extends Beverage {
10
11    private String description = "House Blend Coffee";
12
13    public HouseBlend() {
14        this.description = "House Blend Coffee";
15    }
16
17    public String getDescription() {
18        return description;
19    }
20
21    public double cost() {
22        return .89 + super.cost();
23    }
24
25    public void setDescription(String description) {
26        this.description = description;
27    }
28}
```

Espresso.java

```
6  package cse.design_pattern.ch03.enhancement1;
7
8  public class Espresso extends Beverage {
9
10     public Espresso() {
11             description = "Espresso";
12   }
13
14       /**
15        * @return super.cost() 추가
16        */
17
18   @Override
19       public double cost() {
20             return 1.99 + super.cost();
21   }
22
23
24 }
```

SomeCaffe.java

```
6  package cse.design_pattern.ch03.enhancement1;
7
8  public class SomeCaffe {
9
10
11
12
13
14
15  * @param args the command line arguments
16
17  public static void main(String[] args) {
18      HouseBlend hb = new HouseBlend();
19      Espresso esp = new Espresso();
20
21      System.out.println("HouseBlend 커피는 얼마인가요?");
22      System.out.println(String.format("HouseBlend는 %.2f입니다.", hb.cost()));
23
24      System.out.println("Espresso 커피는 얼마인가요?");
25      System.out.println(String.format("Espresso %.2f입니다.", esp.cost()));
26
27      System.out.println("HouseBlend 커피에 우유를 추가해 주세요.");
28      hb.setMilk(true);
29      System.out.println(String.format("우유 추가한 HouseBlend는 %.2f입니다.",
30          hb.cost()));
31
32      System.out.println("HouseBlend 커피에 우유를 한 번 두 추가하면 얼마인가요?");
33      System.out.println("현재 클래스 구조로는 더 이상 추가할 수 없습니다.");
34
35
36  }
```

실행 결과

```
--- exec-maven-plugin:1.2.1:exec (default-cli) @ desing_pattern ---
HouseBlend 커피는 얼마인가요?
HouseBlend는 0.89입니다.
Espresso 커피는 얼마인가요?
Espresso 1.99입니다.
HouseBlend 커피에 우유를 추가해 주세요.
우유 추가한 HouseBlend는 0.99입니다.
HouseBlend 커피에 우유를 한 번 두 번 추가하면 얼마인가요?
현재 클래스 구조로는 더 이상 추가할 수 없습니다.
```

문제점

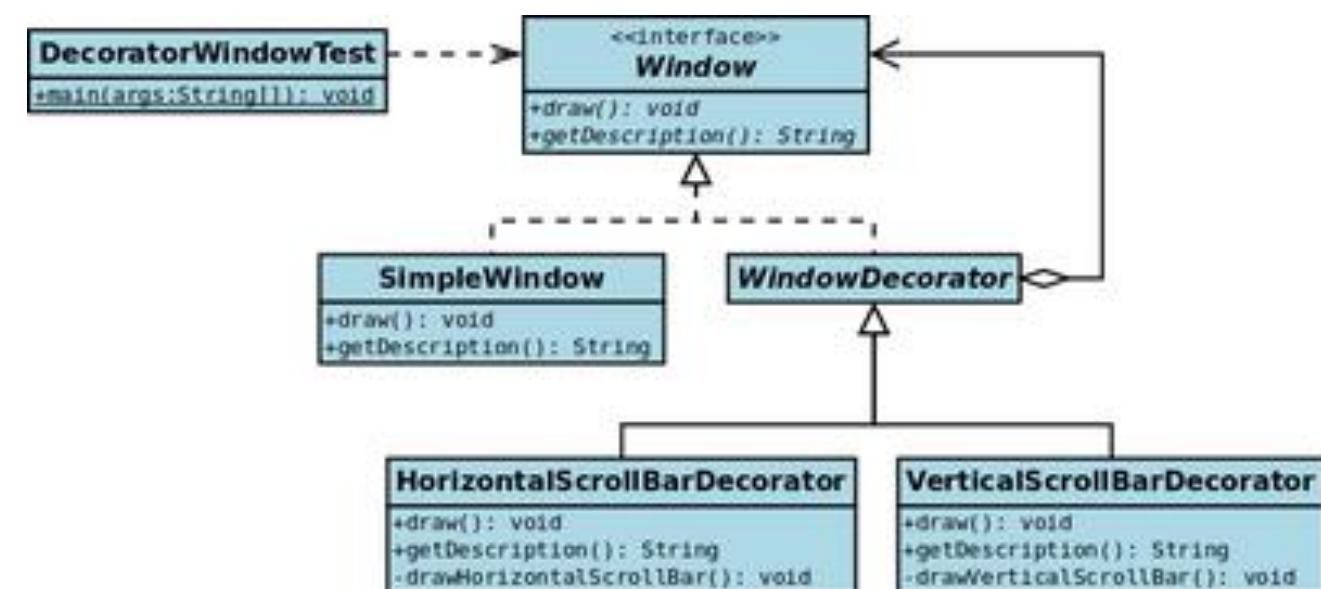
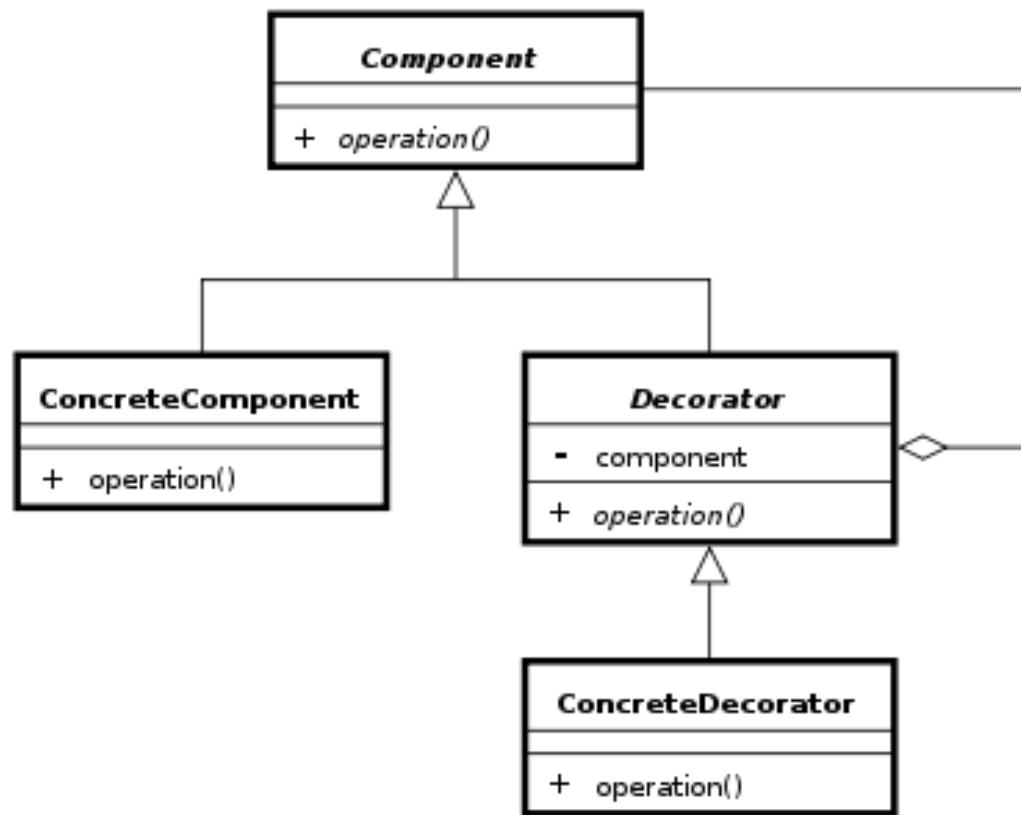
- 첨가물 가격 변동 시 Beverage 클래스 수정해야 함.
- 첨가물 종류가 많아지면 Beverage 클래스에
 - 새로운 메소드 추가: set~(), has~()
 - cost() 메소드 수정 필요
- Tea와 같은 새로운 음료 출시될 때 Beverage를 상속하면
 - hasWhip() 과 같은 메소드 상속받아 휘핑 크림 추가 가능
→ 별로 좋은 생각 아닌 듯!
- 동일 첨가물을 두 번 추가하려면?
 - 기존의 set~() 메소드로는 반영 불가

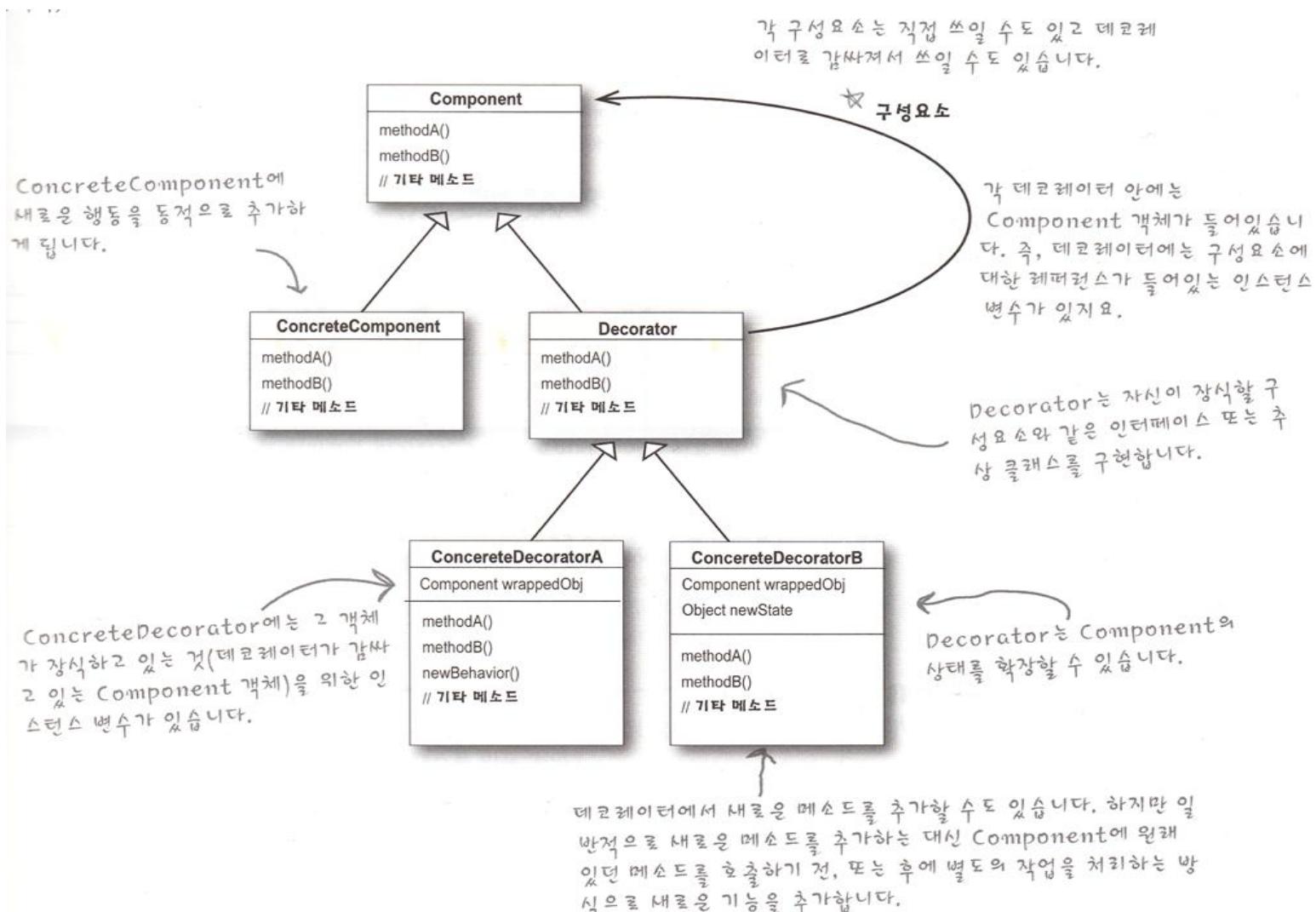
Open-Closed Principle

- Software entities (classes, modules, functions, etc.) should be **open for extension**, but **closed for modification**.
- 확장에는 열려 있고 수정에는 닫혀 있는 코드 구조 필요
- **기존 코드는 건드리지 않은 채로 확장을 통해서 새로운 행동을 간단하게 추가할 수 있다면!!!**
- 참고: <http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOOD>

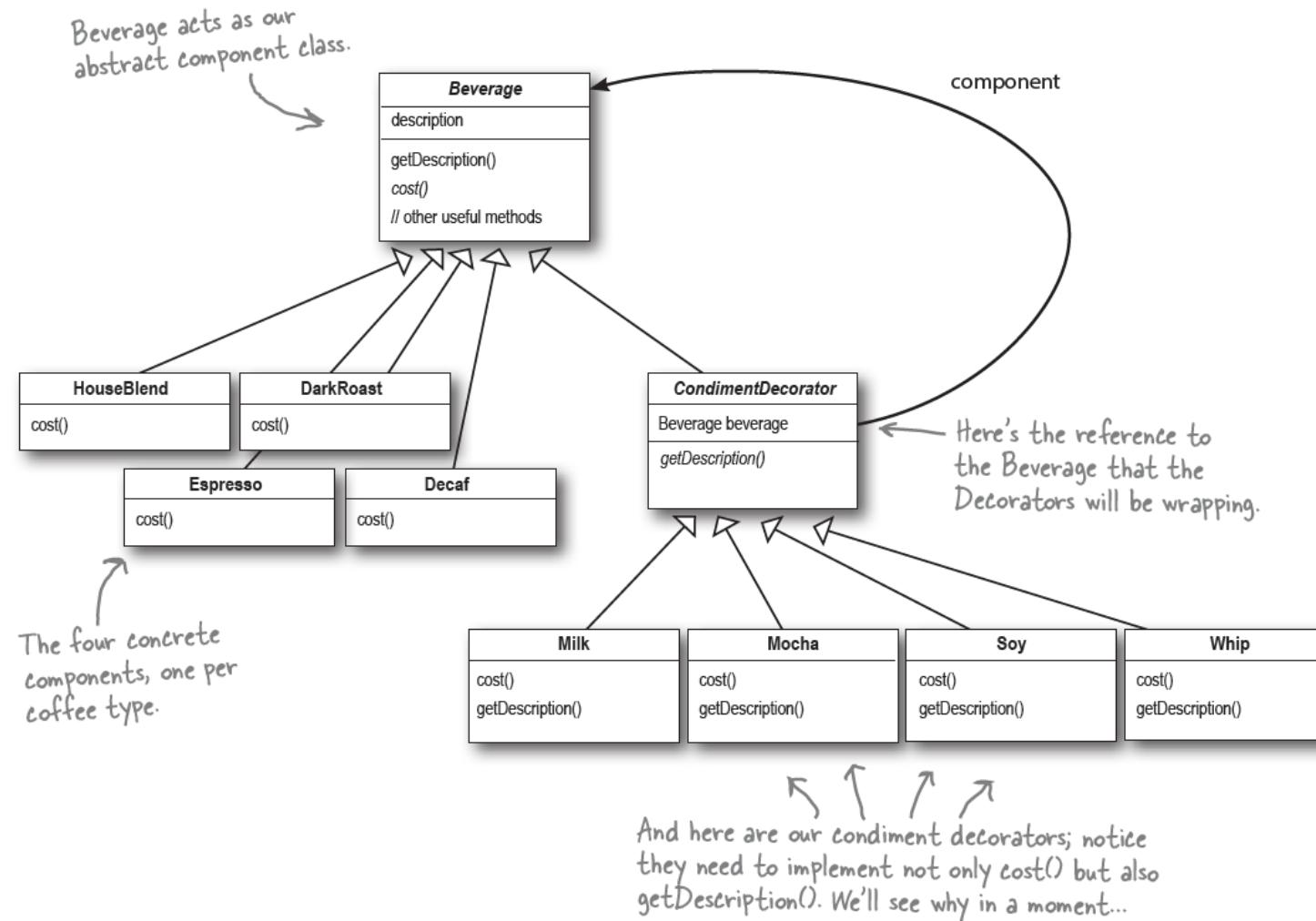
Decorator Pattern

- A design pattern that allows new/additional behavior to be added to an existing class dynamically.

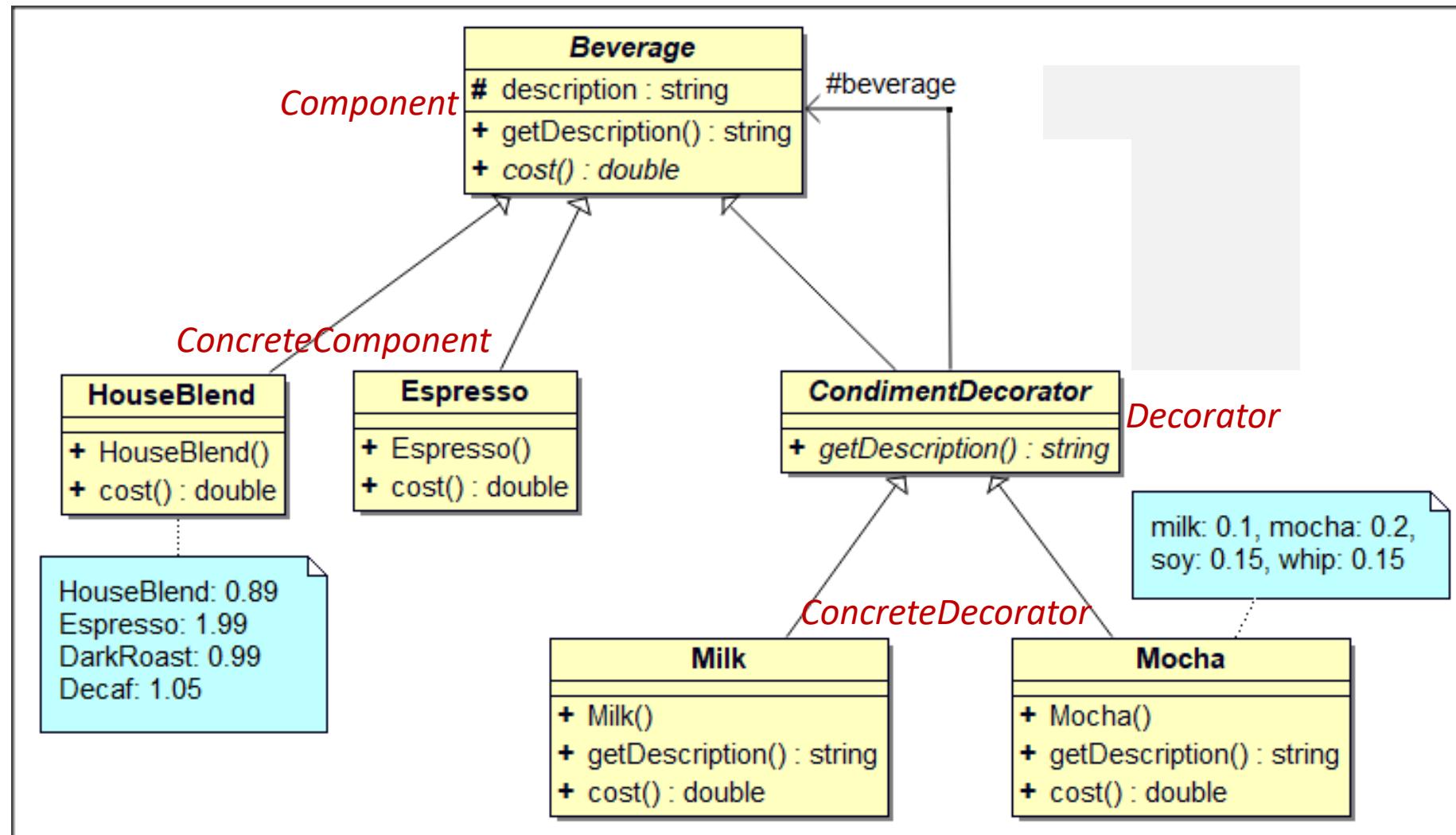




음료수 꾸미기 (Decorating our Beverages)

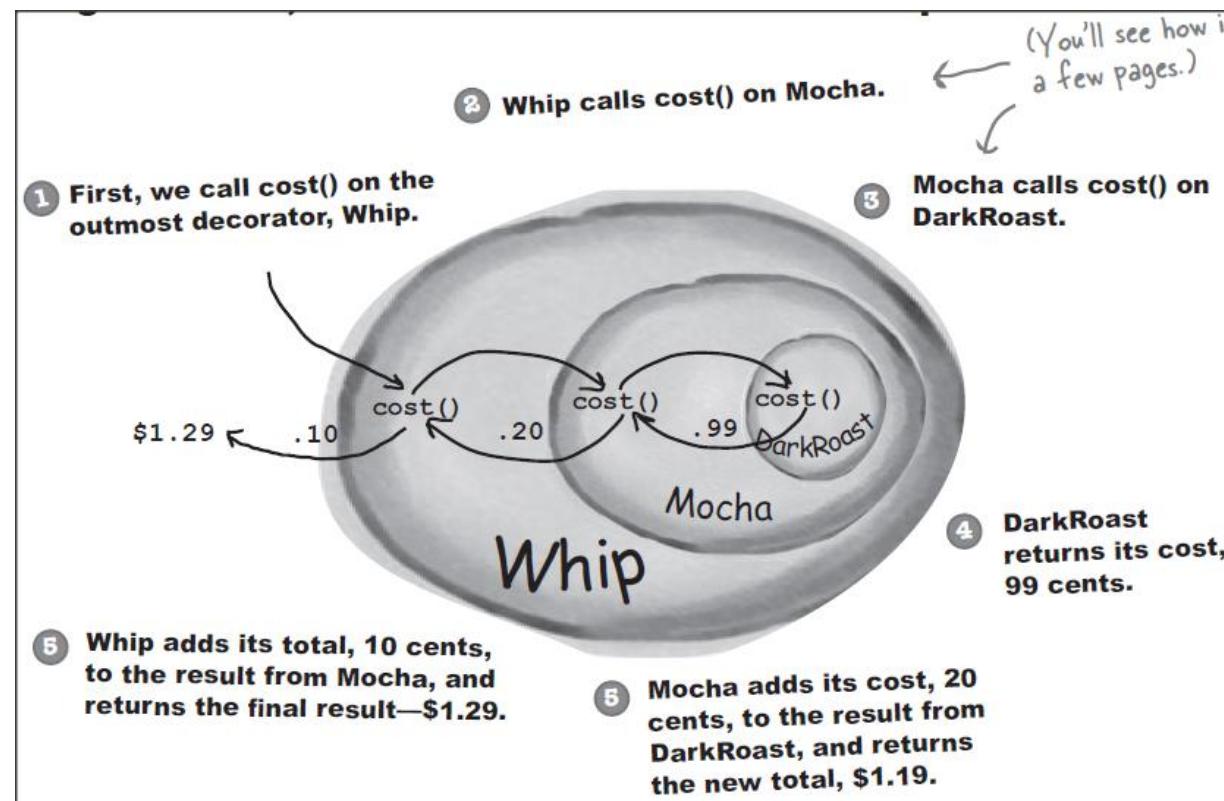


개선 2 (enhancement2)



데코레이터 사용 방법

- new Whip(new Mocha(new DarkRoast())).cost()



Beverage.java

- Component에 해당

```
1 package cse.design_pattern.ch03.enhancement2;
2
3 public abstract class Beverage {
4
5     protected String description = "Unknown Beverage";
6
7     public String getDescription() {
8         return description;
9     }
10
11     public abstract double cost();
12 }
```

CondimentDecorator.java

- Decorator에 해당

```
2 package cse.design_pattern.ch03.enhancement02;
3
4 public abstract class CondimentDecorator extends Beverage {
5     protected Beverage beverage;
6
7     @Override
8     public abstract String getDescription();
9
10 }
```

HouseBlend.java

- ConcreteComponent에 해당

```
1 package cse.design_pattern.ch03.enhancement2;
2
3 public class HouseBlend extends Beverage {
4
5     public HouseBlend() {
6         description = "House Blend Coffee";
7     }
8
9     public double cost() {
10        return .89;
11    }
12 }
```

Espresso.java

- ConcreteComponent에 해당

```
1 package cse.design_pattern.ch03.enhancement2;
2
3 public class Espresso extends Beverage {
4
5     public Espresso() {
6         description = "Espresso";
7     }
8
9     public double cost() {
10        return 1.99;
11    }
12 }
```

Milk.java

- ConcreteDecorator에 해당

```
1 package cse.design_pattern.ch03.enhancement02;
2
3 public class Milk extends CondimentDecorator {
4
5     public Milk(Beverage beverage) {
6         this.beverage = beverage;
7     }
8
9
10    public String getDescription() {
11        return beverage.getDescription() + ", Milk";
12    }
13
14    public double cost() {
15        return .10 + beverage.cost();
16    }
}
```

Mocha.java

- ConcreteDecorator에 해당

```
1 package cse.design_pattern.ch03.enhancement02
2
3 public class Mocha extends CondimentDecorator {
4
5     public Mocha(Beverage beverage) {
6         this.beverage = beverage;
7     }
8
9     public String getDescription() {
10        return beverage.getDescription() + ", Mocha";
11    }
12
13    public double cost() {
14        return .20 + beverage.cost();
15    }
16 }
```

StarbuzzCoffee.java

```
1 package cse.design_pattern.ch03.enhancement2;
2
3 public class StarbuzzCoffee {
4
5     기본 하우스 블렌드
6
7     6
8
9     7
10    우유 추가
11
12    9
13    10
14    모카 추가
15
16    12
17    13
18    모카 추가
19    (더블모카)
20
21    16
22
23    17
24
25    18
26
27    19
28
29    20
30
31    21
32
33    22
34
35    23
36
37    24
38
39    25
40
41    26
42
43    27
44
45    28
46
47    29
48
49    20
50
51    21
52
53    22
54
55    23
56
57    24
58
59    25
60
61    26
62
63    27
64
65    28
66
67    29
68
69    20
70
71    21
72
73    22
74
75    23
76
77    24
78
79    25
80
81    26
82
83    27
84
85    28
86
87    29
88
89    20
90
91    21
92
93    22
94
95    23
96
97    24
98
99    25
100   26
101
102   27
103
104   28
105
106   29
107
108   20
109
10
110   21
111
112   22
113
114   23
115
116   24
117
118   25
119
120   26
121
122   27
123
124   28
125
126   29
127
128   20
129
129   21
130
130   22
131
131   23
132
132   24
133
133   25
134
134   26
135
135   27
136
136   28
137
137   29
138
138   20
139
139   21
140
140   22
141
141   23
142
142   24
143
143   25
144
144   26
145
145   27
146
146   28
147
147   29
148
148   20
149
149   21
150
150   22
151
151   23
152
152   24
153
153   25
154
154   26
155
155   27
156
156   28
157
157   29
158
158   20
159
159   21
160
160   22
161
161   23
162
162   24
163
163   25
164
164   26
165
165   27
166
166   28
167
167   29
168
168   20
169
169   21
170
170   22
171
171   23
172
172   24
173
173   25
174
174   26
175
175   27
176
176   28
177
177   29
178
178   20
179
179   21
180
180   22
181
181   23
182
182   24
183
183   25
184
184   26
185
185   27
186
186   28
187
187   29
188
188   20
189
189   21
190
190   22
191
191   23
192
192   24
193
193   25
194
194   26
195
195   27
196
196   28
197
197   29
198
198   20
199
199   21
200
200   22
201
201   23
202
202   24
203
203   25
204
204   26
205
205   27
206
206   28
207
207   29
208
208   20
209
209   21
210
210   22
211
211   23
212
212   24
213
213   25
214
214   26
215
215   27
216
216   28
217
217   29
218
218   20
219
219   21
220
220   22
221
221   23
222
222   24
223
223   25
224
224   26
225
225   27
226
226   28
227
227   29
228
228   20
229
229   21
230
230   22
231
231   23
232
232   24
233
233   25
234
234   26
235
235   27
236
236   28
237
237   29
238
238   20
239
239   21
240
240   22
241
241   23
242
242   24
243
243   25
244
244   26
245
245   27
246
246   28
247
247   29
248
248   20
249
249   21
250
250   22
251
251   23
252
252   24
253
253   25
254
254   26
255
255   27
256
256   28
257
257   29
258
258   20
259
259   21
260
260   22
261
261   23
262
262   24
263
263   25
264
264   26
265
265   27
266
266   28
267
267   29
268
268   20
269
269   21
270
270   22
271
271   23
272
272   24
273
273   25
274
274   26
275
275   27
276
276   28
277
277   29
278
278   20
279
279   21
280
280   22
281
281   23
282
282   24
283
283   25
284
284   26
285
285   27
286
286   28
287
287   29
288
288   20
289
289   21
290
290   22
291
291   23
292
292   24
293
293   25
294
294   26
295
295   27
296
296   28
297
297   29
298
298   20
299
299   21
300
300   22
301
301   23
302
302   24
303
303   25
304
304   26
305
305   27
306
306   28
307
307   29
308
308   20
309
309   21
310
310   22
311
311   23
312
312   24
313
313   25
314
314   26
315
315   27
316
316   28
317
317   29
318
318   20
319
319   21
320
320   22
321
321   23
322
322   24
323
323   25
324
324   26
325
325   27
326
326   28
327
327   29
328
328   20
329
329   21
330
330   22
331
331   23
332
332   24
333
333   25
334
334   26
335
335   27
336
336   28
337
337   29
338
338   20
339
339   21
340
340   22
341
341   23
342
342   24
343
343   25
344
344   26
345
345   27
346
346   28
347
347   29
348
348   20
349
349   21
350
350   22
351
351   23
352
352   24
353
353   25
354
354   26
355
355   27
356
356   28
357
357   29
358
358   20
359
359   21
360
360   22
361
361   23
362
362   24
363
363   25
364
364   26
365
365   27
366
366   28
367
367   29
368
368   20
369
369   21
370
370   22
371
371   23
372
372   24
373
373   25
374
374   26
375
375   27
376
376   28
377
377   29
378
378   20
379
379   21
380
380   22
381
381   23
382
382   24
383
383   25
384
384   26
385
385   27
386
386   28
387
387   29
388
388   20
389
389   21
390
390   22
391
391   23
392
392   24
393
393   25
394
394   26
395
395   27
396
396   28
397
397   29
398
398   20
399
399   21
400
400   22
401
401   23
402
402   24
403
403   25
404
404   26
405
405   27
406
406   28
407
407   29
408
408   20
409
409   21
410
410   22
411
411   23
412
412   24
413
413   25
414
414   26
415
415   27
416
416   28
417
417   29
418
418   20
419
419   21
420
420   22
421
421   23
422
422   24
423
423   25
424
424   26
425
425   27
426
426   28
427
427   29
428
428   20
429
429   21
430
430   22
431
431   23
432
432   24
433
433   25
434
434   26
435
435   27
436
436   28
437
437   29
438
438   20
439
439   21
440
440   22
441
441   23
442
442   24
443
443   25
444
444   26
445
445   27
446
446   28
447
447   29
448
448   20
449
449   21
450
450   22
451
451   23
452
452   24
453
453   25
454
454   26
455
455   27
456
456   28
457
457   29
458
458   20
459
459   21
460
460   22
461
461   23
462
462   24
463
463   25
464
464   26
465
465   27
466
466   28
467
467   29
468
468   20
469
469   21
470
470   22
471
471   23
472
472   24
473
473   25
474
474   26
475
475   27
476
476   28
477
477   29
478
478   20
479
479   21
480
480   22
481
481   23
482
482   24
483
483   25
484
484   26
485
485   27
486
486   28
487
487   29
488
488   20
489
489   21
490
490   22
491
491   23
492
492   24
493
493   25
494
494   26
495
495   27
496
496   28
497
497   29
498
498   20
499
499   21
500
500   22
501
501   23
502
502   24
503
503   25
504
504   26
505
505   27
506
506   28
507
507   29
508
508   20
509
509   21
510
510   22
511
511   23
512
512   24
513
513   25
514
514   26
515
515   27
516
516   28
517
517   29
518
518   20
519
519   21
520
520   22
521
521   23
522
522   24
523
523   25
524
524   26
525
525   27
526
526   28
527
527   29
528
528   20
529
529   21
530
530   22
531
531   23
532
532   24
533
533   25
534
534   26
535
535   27
536
536   28
537
537   29
538
538   20
539
539   21
540
540   22
541
541   23
542
542   24
543
543   25
544
544   26
545
545   27
546
546   28
547
547   29
548
548   20
549
549   21
550
550   22
551
551   23
552
552   24
553
553   25
554
554   26
555
555   27
556
556   28
557
557   29
558
558   20
559
559   21
560
560   22
561
561   23
562
562   24
563
563   25
564
564   26
565
565   27
566
566   28
567
567   29
568
568   20
569
569   21
570
570   22
571
571   23
572
572   24
573
573   25
574
574   26
575
575   27
576
576   28
577
577   29
578
578   20
579
579   21
580
580   22
581
581   23
582
582   24
583
583   25
584
584   26
585
585   27
586
586   28
587
587   29
588
588   20
589
589   21
590
590   22
591
591   23
592
592   24
593
593   25
594
594   26
595
595   27
596
596   28
597
597   29
598
598   20
599
599   21
600
600   22
601
601   23
602
602   24
603
603   25
604
604   26
605
605   27
606
606   28
607
607   29
608
608   20
609
609   21
610
610   22
611
611   23
612
612   24
613
613   25
614
614   26
615
615   27
616
616   28
617
617   29
618
618   20
619
619   21
620
620   22
621
621   23
622
622   24
623
623   25
624
624   26
625
625   27
626
626   28
627
627   29
628
628   20
629
629   21
630
630   22
631
631   23
632
632   24
633
633   25
634
634   26
635
635   27
636
636   28
637
637   29
638
638   20
639
639   21
640
640   22
641
641   23
642
642   24
643
643   25
644
644   26
645
645   27
646
646   28
647
647   29
648
648   20
649
649   21
650
650   22
651
651   23
652
652   24
653
653   25
654
654   26
655
655   27
656
656   28
657
657   29
658
658   20
659
659   21
660
660   22
661
661   23
662
662   24
663
663   25
664
664   26
665
665   27
666
666   28
667
667   29
668
668   20
669
669   21
670
670   22
671
671   23
672
672   24
673
673   25
674
674   26
675
675   27
676
676   28
677
677   29
678
678   20
679
679   21
680
680   22
681
681   23
682
682   24
683
683   25
684
684   26
685
685   27
686
686   28
687
687   29
688
688   20
689
689   21
690
690   22
691
691   23
692
692   24
693
693   25
694
694   26
695
695   27
696
696   28
697
697   29
698
698   20
699
699   21
700
700   22
701
701   23
702
702   24
703
703   25
704
704   26
705
705   27
706
706   28
707
707   29
708
708   20
709
709   21
710
710   22
711
711   23
712
712   24
713
713   25
714
714   26
715
715   27
716
716   28
717
717   29
718
718   20
719
719   21
720
720   22
721
721   23
722
722   24
723
723   25
724
724   26
725
725   27
726
726   28
727
727   29
728
728   20
729
729   21
730
730   22
731
731   23
732
732   24
733
733   25
734
734   26
735
735   27
736
736   28
737
737   29
738
738   20
739
739   21
740
740   22
741
741   23
742
742   24
743
743   25
744
744   26
745
745   27
746
746   28
747
747   29
748
748   20
749
749   21
750
750   22
751
751   23
752
752   24
753
753   25
754
754   26
755
755   27
756
756   28
757
757   29
758
758   20
759
759   21
760
760   22
761
761   23
762
762   24
763
763   25
764
764   26
765
765   27
766
766   28
767
767   29
768
768   20
769
769   21
770
770   22
771
771   23
772
772   24
773
773   25
774
774   26
775
775   27
776
776   28
777
777   29
778
778   20
779
779   21
780
780   22
781
781   23
782
782   24
783
783   25
784
784   26
785
785   27
786
786   28
787
787   29
788
788   20
789
789   21
790
790   22
791
791   23
792
792   24
793
793   25
794
794   26
795
795   27
796
796   28
797
797   29
798
798   20
799
799   21
800
800   22
801
801   23
802
802   24
803
803   25
804
804   26
805
805   27
806
806   28
807
807   29
808
808   20
809
809   21
810
810   22
811
811   23
812
812   24
813
813   25
814
814   26
815
815   27
816
816   28
817
817   29
818
818   20
819
819   21
820
820   22
821
821   23
822
822   24
823
823   25
824
824   26
825
825   27
826
826   28
827
827   29
828
828   20
829
829   21
830
830   22
831
831   23
832
832   24
833
833   25
834
834   26
835
835   27
836
836   28
837
837   29
838
838   20
839
839   21
840
840   22
841
841   23
842
842   24
843
843   25
844
844   26
845
845   27
846
846   28
847
847   29
848
848   20
849
849   21
850
850   22
851
851   23
852
852   24
853
853   25
854
854   26
855
855   27
856
856   28
857
857   29
858
858   20
859
859   21
860
860   22
861
861   23
862
862   24
863
863   25
864
864   26
865
865   27
866
866   28
867
867   29
868
868   20
869
869   21
870
870   22
871
871   23
872
872   24
873
873   25
874
874   26
875
875   27
876
876   28
877
877   29
878
878   20
879
879   21
880
880   22
881
881   23
882
882   24
883
883   25
884
884   26
885
885   27
886
886   28
887
887   29
888
888   20
889
889   21
890
890   22
891
891   23
892
892   24
893
893   25
894
894   26
895
895   27
896
896   28
897
897   29
898
898   20
899
899   21
900
900   22
901
901   23
902
902   24
903
903   25
904
904   26
905
905   27
906
906   28
907
907   29
908
908   20
909
909   21
910
910   22
911
911   23
912
912   24
913
913   25
914
914   26
915
915   27
916
916   28
917
917   29
918
918   20
919
919   21
920
920   22
921
921   23
922
922   24
923
923   25
924
924   26
925
925   27
926
926   28
927
927   29
928
928   20
929
929   21
930
930   22
931
931   23
932
932   24
933
933   25
934
934   26
935
935   27
936
936   28
937
937   29
938
938   20
939
939   21
940
940   22
941
941   23
942
942   24
943
943   25
944
944   26
945
945   27
946
946   28
947
947   29
948
948   20
949
949   21
950
950   22
951
951   23
952
952   24
953
953   25
954
954   26
955
955   27
956
956   28
957
957   29
958
958   20
959
959   21
960
960   22
961
961   23
962
962   24
963
963   25
964
964   26
965
965   27
966
966   28
967
967   29
968
968   20
969
969   21
970
970   22
971
971   23
972
972   24
973
973   25
974
974   26
975
975   27
976
976   28
977
977   29
978
978   20
979
979   21
980
980   22
981
981   23
982
```

```
18 Beverage b2 = new Mocha(  
19     new Mocha(  
20         new Milk(  
21             new HouseBlend())));  
22 System.out.println(b2.getDescription() + " $" + b2.cost());  
23 }  
24 }
```

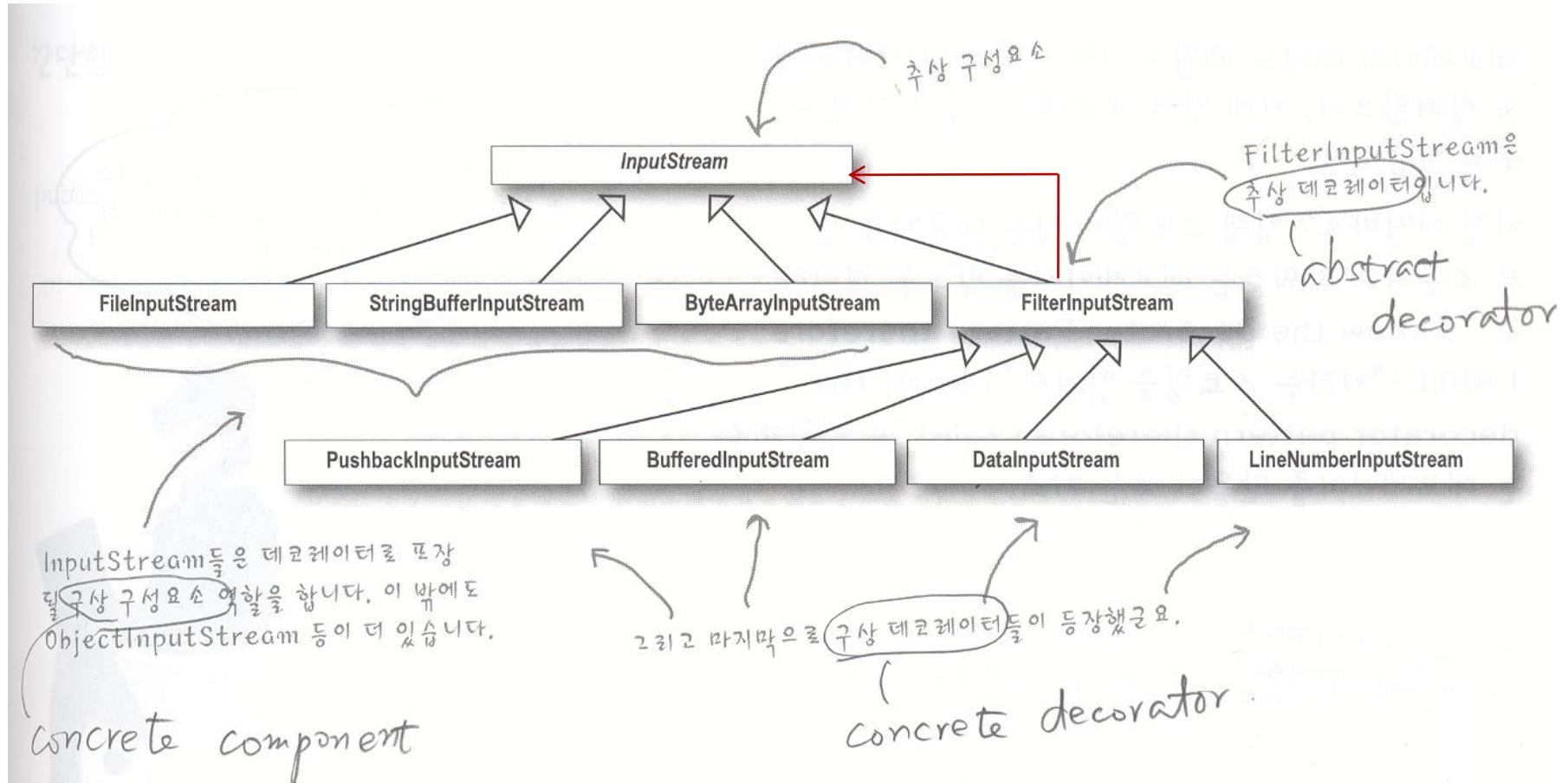
데코레이터

컴포넌트

House Blend Coffee, Milk, Mocha, Mocha \$1.39

java.io 클래스

- Decorator pattern이 적용된 예



java.io.*

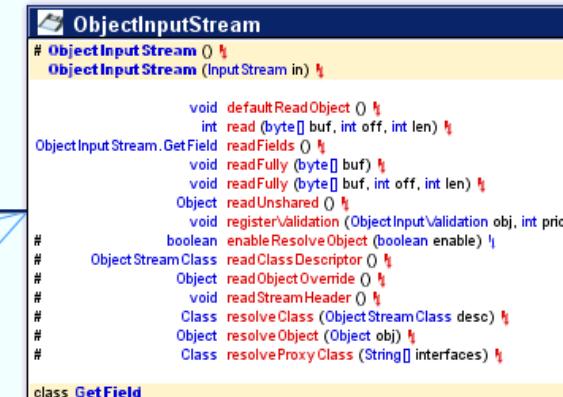
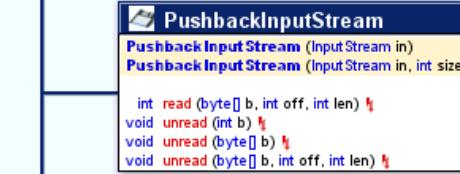
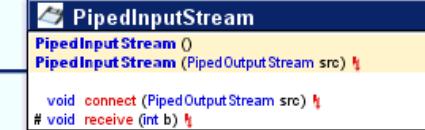
InputStream

Methods declared in supertypes are hidden in subtypes

<https://docs.oracle.com/javase/8/docs/api/java/io/InputStream.html>



3장. Decorator Pattern

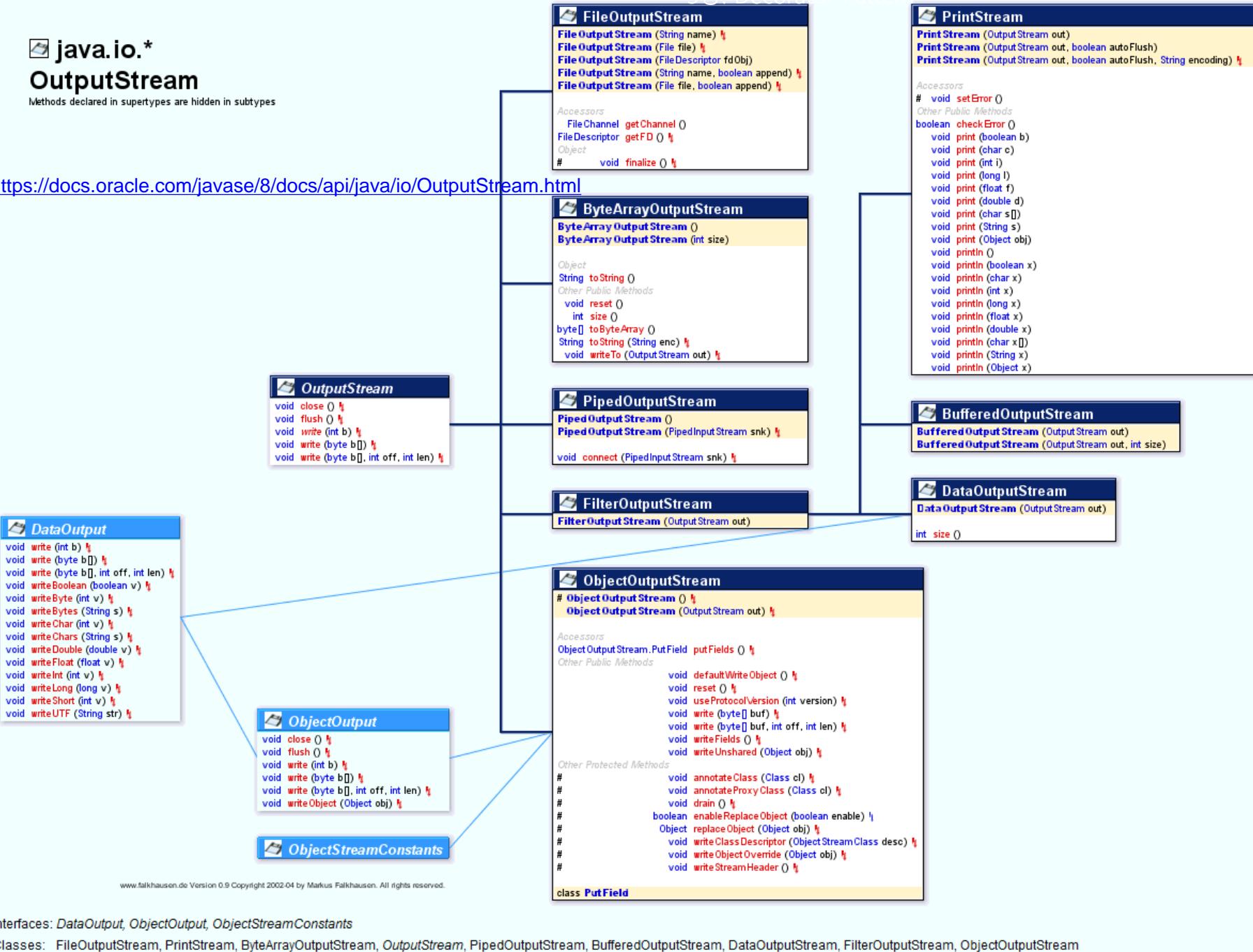


java.io.*

OutputStream

Methods declared in supertypes are hidden in subtypes

<https://docs.oracle.com/javase/8/docs/api/java/io/OutputStream.html>

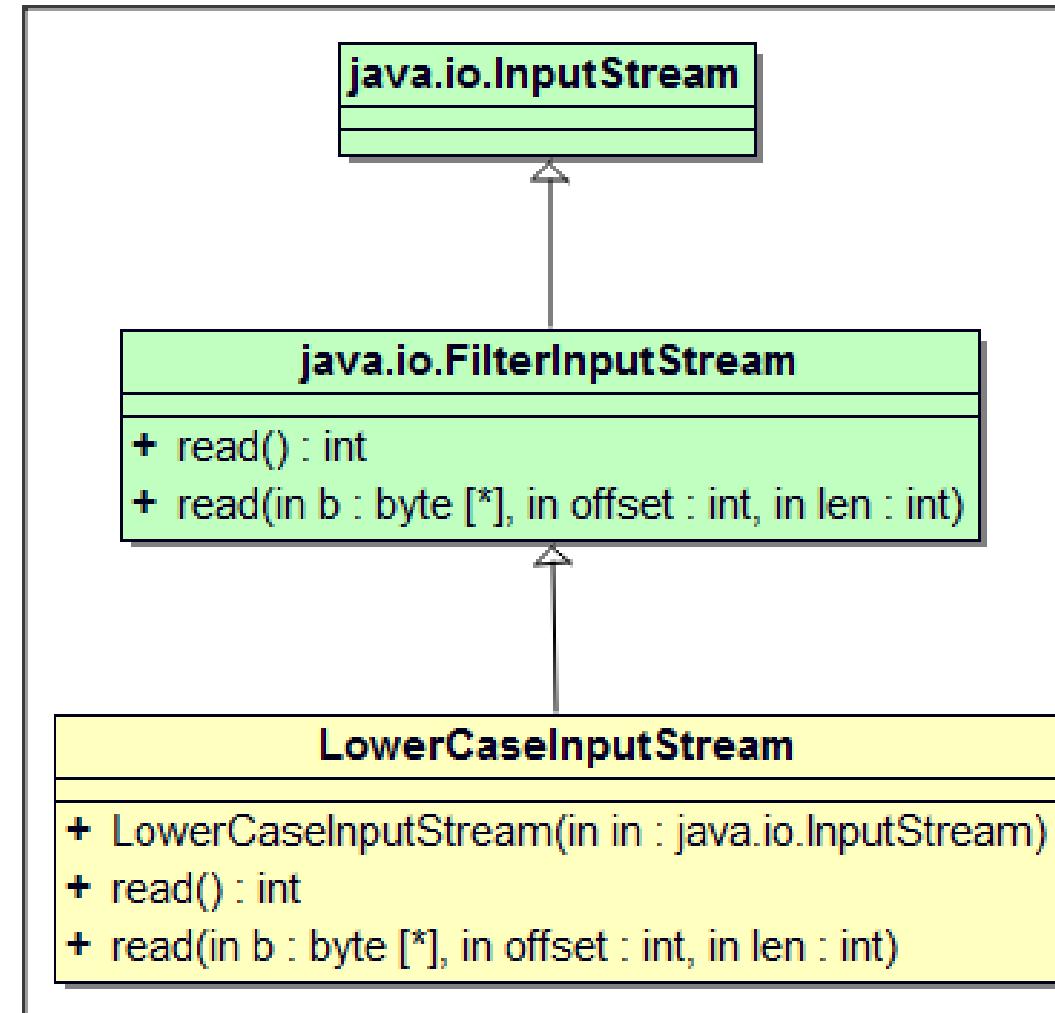


자바 I/O DECORATOR 실습

cse.design_pattern.ch03.io 패키지

AI: JDK21에서 decorator pattern이 적용된 클래스나 인터페이스를 알려줘.

실습 클래스 다이어그램 (io)

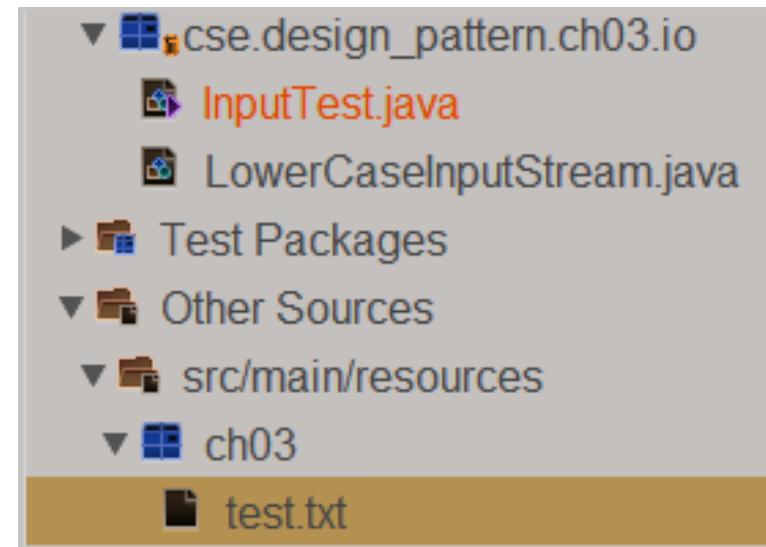


LowerCaseInputStream.java

```
1 package cse.design_pattern.ch03.io;
2
3 import java.io.*;
4
5 public class LowerCaseInputStream extends FilterInputStream {
6
7     public LowerCaseInputStream(InputStream in) {
8         super(in);
9     }
10
11    @Override
12    public int read() throws IOException {
13        int c = super.read();
14        return (c == -1 ? c : Character.toLowerCase((char) c));
15    }
16
17    @Override
18    public int read(byte[] b, int offset, int len) throws IOException {
19        int result = super.read(b, offset, len);
20        for (int i = offset; i < offset + result; i++) {
21            b[i] = (byte) Character.toLowerCase((char) b[i]);
22        }
23        return result;
24    }
25 }
```

test.txt

- Other Sources > **src/main/resources** (**src/main** 폴더 아래에 **resources** 폴더 생성해야 함)
 - ch03 폴더 생성 → Empty File... 생성



- test.txt 내용

1

I know the Decorator Pattern therefore I RULE!

InputTest.java

```
1 package cse.design_pattern.ch03.io;
2
3 import java.io.*;
4
5 public class InputTest {
6
7     public static void main(String[] args) throws IOException {
8
9         System.out.println("File.getPath(): " + new java.io.File(".").getPath());
10        System.out.println("File.getAbsolutePath(): " + new java.io.File(".").getAbsolutePath());
11        System.out.println("File.getCanonicalPath(): " + new java.io.File(".").getCanonicalPath());
12
13        System.out.printf("Working Directory = %s\n", System.getProperty("user.dir"));
14        System.out.println("-----");
```

```
File.getPath(): .
File.getAbsolutePath(): D:\NetBeansProjects\design_pattern\
File.getCanonicalPath(): D:\NetBeansProjects\design_pattern
Working Directory = D:\NetBeansProjects\design_pattern
-----
```

```
16 try {  
17      InputStream in  
18     = new LowerCaseInputStream(  
19         new BufferedInputStream(  
20             new FileInputStream("./src/main/resources/ch03/test.txt")));  
21  
22      int c;  
23     while ((c = in.read()) >= 0) {  
24         System.out.print((char) c);  
25     }  
26  
27     in.close();  
28 } catch (IOException e) {  
29      e.printStackTrace();  
30 }  
31 }  
32 }
```

i know the decorator pattern therefore i rule!

참고. InputTest2.java

```
1 package cse.design_pattern.ch03.io;
2
3 import java.io.File;
4 import java.io.FileInputStream;
5 import java.io.IOException;
6 import java.io.InputStream;
7 import java.net.URL;
8 import java.util.logging.Level;
9 import java.util.logging.Logger;
10
11 public class InputTest2 {
12
13     public void method01() {
14         InputStream resource = getClass().getClassLoader().getResourceAsStream("ch03/test.txt");
15         try (InputStream is = new LowerCaseInputStream(resource)) {
16             while (is.available() > 0) {
17                 System.out.print((char) is.read());
18             }
19         } catch (IOException e) {
20             System.out.println("오류 발생: " + e.getMessage());
21         }
22     }
}
```

```
24  /** getFile() 사용 ...3 lines */
25
26
27  public void method02() {
28      URL url = this.getClass().getClassLoader().getResource("ch03/test.txt");
29      if (url != null) {
30          try {InputStream is = new LowerCaseInputStream(
31              new FileInputStream(url.getFile())));
32              while (is.available() > 0) {
33                  System.out.print((char) is.read());
34              }
35          } catch (IOException ex) {
36              Logger.getLogger(InputTest2.class.getName()).log(Level.SEVERE, null, ex);
37          }
38      } else {
39          System.out.println("url이 null입니다.");
40      }
41  }
42
43  public static void main(String[] args) throws IOException {
44      new InputTest2().method01();
45      new InputTest2().method02();
46  }
47 }
```

--- exec-maven-plugin:1.5.0:exec (default-cli) @
i know the decorator pattern therefore i rule!
i know the decorator pattern therefore i rule!