

능동적 사고 방식의

java

강사 박주병

Park Ju Byeong

Park Ju Byeong



Part07 메모리



01 지역변수, 멤버변수

02 클래스 변수(static 변수)

03 메모리 영역

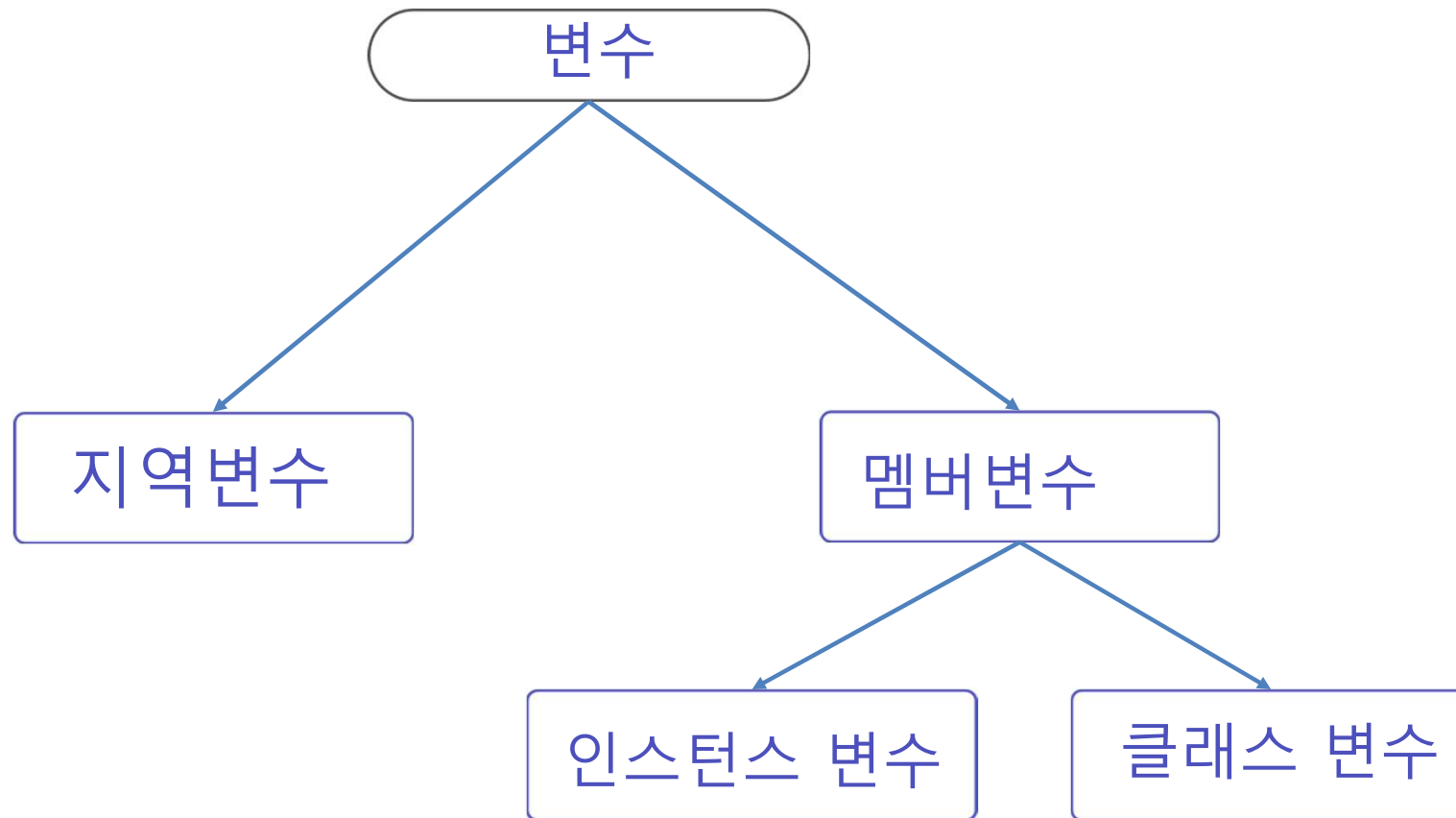
04 실습 문제



01

지역변수, 멤버변수

변수의 종류



```
class test{
```

```
    static int classValue; —————> 멤버변수 -클래스 변수
```

```
    int memberValue; —————> 멤버변수,인스턴스 변수
```

```
    void method1(int parameter) —————> 지역변수  
{
```

```
        int localValue; —————> 지역변수
```

```
        for(int i=0; i <10;i++)
```

```
        {
```

```
            —————> 지역변수
```

```
        }
```

```
    }
```

```
}
```

지역변수

- 메서드 범위 내에서 생성된 변수
- 메서드 실행시 생성되며 메서드 영역이 끝나면 삭제된다.
- 메모리의 스택 영역에 저장된다.

```
public class Test
```

```
{
```

```
    void test()
```

```
    {
```

```
        int age = 30;
```

```
    }
```

```
    void print()
```

```
    {
```

```
        System.out.println(age);
```

```
    }
```

```
}
```

age 변수는 test() 메서드 내에서만 사용가능

```

public class Test
{
    void test()
    {
        for(int i=0;i<5;i++)
        {
            System.out.println(i);
        }

        System.out.println(i);
    }

    void print()
    {
    }
}

```

for문 내에서 생성된 지역변수
이므로 for 영역을 벗어나면
사라진다.

```
public class MyMath {  
  
    int add(int a ,int b)  
    {  
        int result= a+b;  
  
        return result;  
    }  
  
    void test()  
    {  
        System.out.println(a);  
    }  
}
```

매개변수 역시 지역변수이므로
범위를 벗어나서 사용 불가

인스턴스 변수

- 클래스 범위 내에 생성된 변수
- 클래스가 객체화 될때 객체마다 별도로 생성된다.
- 메모리의 힙영역에 저장된다.

```
public class Test
{
    int iv=50;

    void test()
    {
        System.out.println(iv);
    }

    void print()
    {
        System.out.println(iv);
    }
}
```

클래스 내 어디서든 사용가능하다

```
public class Person
{
    String name;
    String RRN;
```

```
public class Main {
```

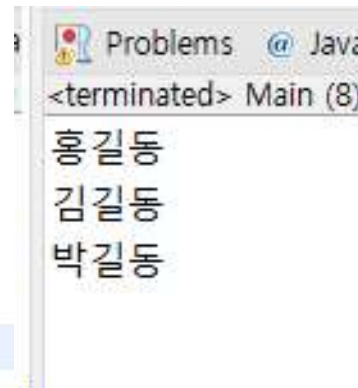
```
    public static void main(String[] args) {
        // TODO Auto-generated method stub
```

```
        Person p1 = new Person();
        Person p2 = new Person();
        Person p3 = new Person();
```

```
        p1.name = "홍길동";
        p2.name = "김길동";
        p3.name = "박길동";
```

```
        System.out.println(p1.name);
        System.out.println(p2.name);
        System.out.println(p3.name);
```

객체마다 별도의 공간을 사용한다.



인스턴스 변수 vs 지역변수

```
5 public class Person
6 {
7     String name;
8     String RRN;
9     ...
10    ...
11    ...
12    ...
13    ...
14    ...
15    ...
16    ...
17    ...
18    ...
19    ...
20    ...
21    ...
22    ...
23    ...
24    ...
25    ...
26    ...
27    ...
28    ...
29    ...
30    ...
31    ...
32    ...
33    ...
34    ...
35    ...
36    ...
37    ...
38    ...
39    ...
40    ...
41    ...
42    ...
43    ...
44    ...
45    ...
46    ...
47    ...
48    ...
49    ...
50    ...
51    ...
52    ...
53    ...
54    ...
55    ...
56    ...
57    ...
58    ...
59    ...
60    ...
61    ...
62    ...
63    ...
64    ...
65    ...
66    ...
67    ...
68    ...
69    ...
70    ...
71    ...
72    ...
73    ...
74    ...
75    ...
76    ...
77    ...
78    ...
79    ...
80    ...
81    ...
82    ...
83    ...
84    ...
85    ...
86    ...
87    ...
88    ...
89    ...
90    ...
91    ...
92    ...
93    ...
94    ...
95    ...
96    ...
97    ...
98    ...
99    ...
100   ...
101   ...
102   ...
103   ...
104   ...
105   ...
106   ...
107   ...
108   ...
109   ...
110   ...
111   ...
112   ...
113   ...
114   ...
115   ...
116   ...
117   ...
118   ...
119   ...
120   ...
121   ...
122   ...
123   ...
124   ...
125   ...
126   ...
127   ...
128   ...
129   ...
130   ...
131   ...
132   ...
133   ...
134   ...
135   ...
136   ...
137   ...
138   ...
139   ...
140   ...
141   ...
142   ...
143   ...
144   ...
145   ...
146   ...
147   ...
148   ...
149   ...
150   ...
151   ...
152   ...
153   ...
154   ...
155   ...
156   ...
157   ...
158   ...
159   ...
160   ...
161   ...
162   ...
163   ...
164   ...
165   ...
166   ...
167   ...
168   ...
169   ...
170   ...
171   ...
172   ...
173   ...
174   ...
175   ...
176   ...
177   ...
178   ...
179   ...
180   ...
181   ...
182   ...
183   ...
184   ...
185   ...
186   ...
187   ...
188   ...
189   ...
190   ...
191   ...
192   ...
193   ...
194   ...
195   ...
196   ...
197   ...
198   ...
199   ...
200   ...
201   ...
202   ...
203   ...
204   ...
205   ...
206   ...
207   ...
208   ...
209   ...
210   ...
211   ...
212   ...
213   ...
214   ...
215   ...
216   ...
217   ...
218   ...
219   ...
220   ...
221   ...
222   ...
223   ...
224   ...
225   ...
226   ...
227   ...
228   ...
229   ...
230   ...
231   ...
232   ...
233   ...
234   ...
235   ...
236   ...
237   ...
238   ...
239   ...
240   ...
241   ...
242   ...
243   ...
244   ...
245   ...
246   ...
247   ...
248   ...
249   ...
250   ...
251   ...
252   ...
253   ...
254   ...
255   ...
256   ...
257   ...
258   ...
259   ...
260   ...
261   ...
262   ...
263   ...
264   ...
265   ...
266   ...
267   ...
268   ...
269   ...
270   ...
271   ...
272   ...
273   ...
274   ...
275   ...
276   ...
277   ...
278   ...
279   ...
280   ...
281   ...
282   ...
283   ...
284   ...
285   ...
286   ...
287   ...
288   ...
289   ...
290   ...
291   ...
292   ...
293   ...
294   ...
295   ...
296   ...
297   ...
298   ...
299   ...
300   ...
301   ...
302   ...
303   ...
304   ...
305   ...
306   ...
307   ...
308   ...
309   ...
310   ...
311   ...
312   ...
313   ...
314   ...
315   ...
316   ...
317   ...
318   ...
319   ...
320   ...
321   ...
322   ...
323   ...
324   ...
325   ...
326   ...
327   ...
328   ...
329   ...
330   ...
331   ...
332   ...
333   ...
334   ...
335   ...
336   ...
337   ...
338   ...
339   ...
340   ...
341   ...
342   ...
343   ...
344   ...
345   ...
346   ...
347   ...
348   ...
349   ...
350   ...
351   ...
352   ...
353   ...
354   ...
355   ...
356   ...
357   ...
358   ...
359   ...
360   ...
361   ...
362   ...
363   ...
364   ...
365   ...
366   ...
367   ...
368   ...
369   ...
370   ...
371   ...
372   ...
373   ...
374   ...
375   ...
376   ...
377   ...
378   ...
379   ...
380   ...
381   ...
382   ...
383   ...
384   ...
385   ...
386   ...
387   ...
388   ...
389   ...
390   ...
391   ...
392   ...
393   ...
394   ...
395   ...
396   ...
397   ...
398   ...
399   ...
400   ...
401   ...
402   ...
403   ...
404   ...
405   ...
406   ...
407   ...
408   ...
409   ...
410   ...
411   ...
412   ...
413   ...
414   ...
415   ...
416   ...
417   ...
418   ...
419   ...
420   ...
421   ...
422   ...
423   ...
424   ...
425   ...
426   ...
427   ...
428   ...
429   ...
430   ...
431   ...
432   ...
433   ...
434   ...
435   ...
436   ...
437   ...
438   ...
439   ...
440   ...
441   ...
442   ...
443   ...
444   ...
445   ...
446   ...
447   ...
448   ...
449   ...
450   ...
451   ...
452   ...
453   ...
454   ...
455   ...
456   ...
457   ...
458   ...
459   ...
460   ...
461   ...
462   ...
463   ...
464   ...
465   ...
466   ...
467   ...
468   ...
469   ...
470   ...
471   ...
472   ...
473   ...
474   ...
475   ...
476   ...
477   ...
478   ...
479   ...
480   ...
481   ...
482   ...
483   ...
484   ...
485   ...
486   ...
487   ...
488   ...
489   ...
490   ...
491   ...
492   ...
493   ...
494   ...
495   ...
496   ...
497   ...
498   ...
499   ...
500   ...
501   ...
502   ...
503   ...
504   ...
505   ...
506   ...
507   ...
508   ...
509   ...
510   ...
511   ...
512   ...
513   ...
514   ...
515   ...
516   ...
517   ...
518   ...
519   ...
520   ...
521   ...
522   ...
523   ...
524   ...
525   ...
526   ...
527   ...
528   ...
529   ...
530   ...
531   ...
532   ...
533   ...
534   ...
535   ...
536   ...
537   ...
538   ...
539   ...
540   ...
541   ...
542   ...
543   ...
544   ...
545   ...
546   ...
547   ...
548   ...
549   ...
550   ...
551   ...
552   ...
553   ...
554   ...
555   ...
556   ...
557   ...
558   ...
559   ...
560   ...
561   ...
562   ...
563   ...
564   ...
565   ...
566   ...
567   ...
568   ...
569   ...
570   ...
571   ...
572   ...
573   ...
574   ...
575   ...
576   ...
577   ...
578   ...
579   ...
580   ...
581   ...
582   ...
583   ...
584   ...
585   ...
586   ...
587   ...
588   ...
589   ...
590   ...
591   ...
592   ...
593   ...
594   ...
595   ...
596   ...
597   ...
598   ...
599   ...
600   ...
601   ...
602   ...
603   ...
604   ...
605   ...
606   ...
607   ...
608   ...
609   ...
610   ...
611   ...
612   ...
613   ...
614   ...
615   ...
616   ...
617   ...
618   ...
619   ...
620   ...
621   ...
622   ...
623   ...
624   ...
625   ...
626   ...
627   ...
628   ...
629   ...
630   ...
631   ...
632   ...
633   ...
634   ...
635   ...
636   ...
637   ...
638   ...
639   ...
640   ...
641   ...
642   ...
643   ...
644   ...
645   ...
646   ...
647   ...
648   ...
649   ...
650   ...
651   ...
652   ...
653   ...
654   ...
655   ...
656   ...
657   ...
658   ...
659   ...
660   ...
661   ...
662   ...
663   ...
664   ...
665   ...
666   ...
667   ...
668   ...
669   ...
670   ...
671   ...
672   ...
673   ...
674   ...
675   ...
676   ...
677   ...
678   ...
679   ...
680   ...
681   ...
682   ...
683   ...
684   ...
685   ...
686   ...
687   ...
688   ...
689   ...
690   ...
691   ...
692   ...
693   ...
694   ...
695   ...
696   ...
697   ...
698   ...
699   ...
700   ...
701   ...
702   ...
703   ...
704   ...
705   ...
706   ...
707   ...
708   ...
709   ...
710   ...
711   ...
712   ...
713   ...
714   ...
715   ...
716   ...
717   ...
718   ...
719   ...
720   ...
721   ...
722   ...
723   ...
724   ...
725   ...
726   ...
727   ...
728   ...
729   ...
730   ...
731   ...
732   ...
733   ...
734   ...
735   ...
736   ...
737   ...
738   ...
739   ...
740   ...
741   ...
742   ...
743   ...
744   ...
745   ...
746   ...
747   ...
748   ...
749   ...
750   ...
751   ...
752   ...
753   ...
754   ...
755   ...
756   ...
757   ...
758   ...
759   ...
760   ...
761   ...
762   ...
763   ...
764   ...
765   ...
766   ...
767   ...
768   ...
769   ...
770   ...
771   ...
772   ...
773   ...
774   ...
775   ...
776   ...
777   ...
778   ...
779   ...
780   ...
781   ...
782   ...
783   ...
784   ...
785   ...
786   ...
787   ...
788   ...
789   ...
790   ...
791   ...
792   ...
793   ...
794   ...
795   ...
796   ...
797   ...
798   ...
799   ...
800   ...
801   ...
802   ...
803   ...
804   ...
805   ...
806   ...
807   ...
808   ...
809   ...
810   ...
811   ...
812   ...
813   ...
814   ...
815   ...
816   ...
817   ...
818   ...
819   ...
820   ...
821   ...
822   ...
823   ...
824   ...
825   ...
826   ...
827   ...
828   ...
829   ...
830   ...
831   ...
832   ...
833   ...
834   ...
835   ...
836   ...
837   ...
838   ...
839   ...
840   ...
841   ...
842   ...
843   ...
844   ...
845   ...
846   ...
847   ...
848   ...
849   ...
850   ...
851   ...
852   ...
853   ...
854   ...
855   ...
856   ...
857   ...
858   ...
859   ...
860   ...
861   ...
862   ...
863   ...
864   ...
865   ...
866   ...
867   ...
868   ...
869   ...
870   ...
871   ...
872   ...
873   ...
874   ...
875   ...
876   ...
877   ...
878   ...
879   ...
880   ...
881   ...
882   ...
883   ...
884   ...
885   ...
886   ...
887   ...
888   ...
889   ...
890   ...
891   ...
892   ...
893   ...
894   ...
895   ...
896   ...
897   ...
898   ...
899   ...
900   ...
901   ...
902   ...
903   ...
904   ...
905   ...
906   ...
907   ...
908   ...
909   ...
910   ...
911   ...
912   ...
913   ...
914   ...
915   ...
916   ...
917   ...
918   ...
919   ...
920   ...
921   ...
922   ...
923   ...
924   ...
925   ...
926   ...
927   ...
928   ...
929   ...
930   ...
931   ...
932   ...
933   ...
934   ...
935   ...
936   ...
937   ...
938   ...
939   ...
940   ...
941   ...
942   ...
943   ...
944   ...
945   ...
946   ...
947   ...
948   ...
949   ...
950   ...
951   ...
952   ...
953   ...
954   ...
955   ...
956   ...
957   ...
958   ...
959   ...
960   ...
961   ...
962   ...
963   ...
964   ...
965   ...
966   ...
967   ...
968   ...
969   ...
970   ...
971   ...
972   ...
973   ...
974   ...
975   ...
976   ...
977   ...
978   ...
979   ...
980   ...
981   ...
982   ...
983   ...
984   ...
985   ...
986   ...
987   ...
988   ...
989   ...
990   ...
991   ...
992   ...
993   ...
994   ...
995   ...
996   ...
997   ...
998   ...
999   ...
1000  ...
```

```
int getAge()
{
    Calendar cal = Calendar.getIn

    int nowYear = cal.get(Calenda
    int nowMonth = cal.get(Calend
```

- 객체의 속성을 나타냄
- 사람은 나이,주민번호,키 등등의 데이터를 지닌다.

- 객체와는 관계 없이 메서드 내부에서 임시로 사용하는 목적
- 나이를 계산할때 잠깐 데이터를 저장하는용도

nowYear 를 멤버변수로 선언해도 기능상의 문제는 없지만 불필요한 메모리를 차지하며 코드결합도가 높아져 좋지 않다.

인스턴스 변수 와 지역변수 이름이 같다면?

```
public class Test
{
    int iv=50;
    void test()
    {
        int iv = 10;
        System.out.println(iv);
    }
}
```

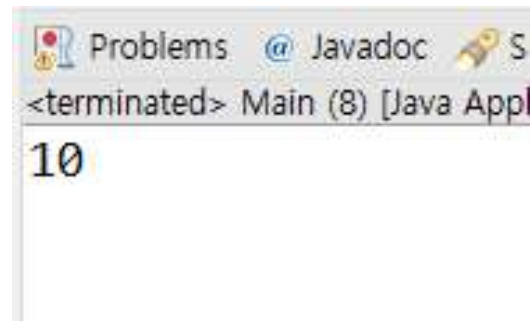
인스턴스 변수

지역변수

```
public class Main {
    public static void main(String[] args) {
        // TODO Auto-generated method stub

        Test t1 = new Test();

        t1.print();
    }
}
```



```
<terminated> Main (8) [Java Appl
10
```

지역변수가 우선시 된다!!

Park Ju Byeong

멤버 변수의 초기화

```
class Person
{
    String name = "홍길동";
    String RRN;

    void showState()
    {
        System.out.println("0");
    }

    String getGender()
    {

```

멤버 변수도 선언과 동시에 초기화 가능하다.

```
113 public static void main(String[] args) {
114     // TODO Auto-generated method stub
115
116     Person p1 = new Person();
117
118     System.out.println(p1.name);
119
120 }
121
```

홍길동

객체 생성시 기본값으로 가지고 있다.

```
class test
```

```
{
```

```
    int v;
```

```
    Person p;
```

```
    void fnc()
```

```
    {
```

```
        int localV;
```

```
        Person localP;
```

```
        System.out.println(v);
```

```
        System.out.println(p);
```

```
        System.out.println(localP);
```

```
        System.out.println(localV);
```

```
    }
```

```
}
```

멤버 변수가 기본형이라면 기본값이 들어가 있다.

멤버 변수가 참조형이라면 null이 기본이다.

<terminated>

0

null

지역변수는 초기화하지 않으면 사용할 수 없다.

String 클래스

- 문자열을 다루기 위한 클래스
- 내부적으로 char[]을 이용하고 있다.
- 객체 생성 없이 사용 할 수 있게 만들어졌다.

```
String str = "문자열이 들어갈수 있습니다.";
```

```
String str = new String("클래스이기에 객체생성해도 된다.");
```

```
System.out.println(str);
```



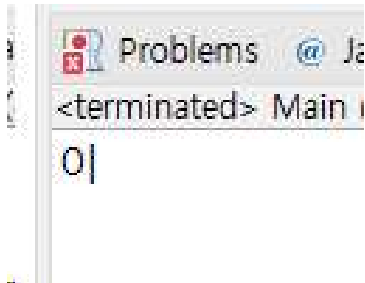
String 클래스의 charAt 메서드

```
String str = "문자열이 들어갈수 있습니다.";
```

0부터 시작하므로 4번째

```
char ch = str.charAt(3);
```

```
System.out.println(ch);
```



String 클래스의 substring메서드

```
String str = "문자열이 들어갈수 있습니다.";
```

```
String str1 = str.substring(0,3);  
String str2 = str.substring(5,7);
```

```
System.out.println(str1);  
System.out.println(str2);
```

0번째 위치 부터 2번째 위치까지

<terminate
문자열

문자열
들어

String 클래스의 split 메서드

```
String str = "문자열이 들어갈수 있습니다.";
String[] temp = str.split(" ");

System.out.println(temp[0]);
System.out.println(temp[1]);
System.out.println(temp[2]);
```

```
<terminated> M
문자열이
들어갈수
있습니다.
```

```
String str = "축구,음악감상,요리,캠핑";
String[] hobbies = str.split(",");

System.out.println(hobbies[0]);
System.out.println(hobbies[1]);
System.out.println(hobbies[2]);
```

```
PROBLEMS
<terminated> Mair
축구
음악감상
요리
```

String 클래스 equals 메서드

```
String temp = "안녕";
```

```
if(temp == "안녕")
```

```
    System.out.println("일치합니다");
```

경우에 따라서 false가 나올수 있다.
자세한건 java2에서 다룬다.



```
String temp = "안녕";
```

```
if(temp.equals("안녕"))
```

```
    System.out.println("일치합니다");
```

1-1 실습문제 (normal)

Person 클래스를 만들어 사용해보자

- 멤버변수 : String name (이름)
String RRN (주민번호)

```
public class Main {  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
  
        Person p1 = new Person();  
        Person p2 = new Person();  
        Person p3 = new Person();  
  
        p1.name = "홍길동";  
        p1.RRN = "750215-1234567";  
  
        p2.name = "김길동";  
        p2.RRN = "850215-1234567";  
  
        p3.name = "박길동";  
        p3.RRN = "040215-2234567";  
  
        System.out.println("이름:" + p1.name + "\t주민번호: " + p1.RRN);  
        System.out.println("이름:" + p2.name + "\t주민번호: " + p2.RRN);  
        System.out.println("이름:" + p3.name + "\t주민번호: " + p3.RRN);  
    }  
}
```

```
<terminated> Main (8) [Java Application] C:\Users\Wzest1\p2\pool\plug  
이름:홍길동      주민번호: 750215-1234567  
이름:김길동      주민번호: 850215-1234567  
이름:김길동      주민번호: 850215-1234567
```

1-1 문제풀이 (normal)

1 1

Park Ju Byeong

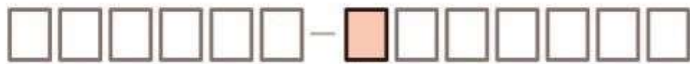
Park Ju Byeong

1-2 실습문제 (hard)

앞서 만든 Person 클래스에 기능을 추가해보자

- 멤버 메서드 : void showState() 이름과 주민번호를 출력한다
 String getGender() 주민번호를 이용해 성별을 반환한다.
- showState 메서드 내부에서 getGender() 메서드를 이용해 성별도 같이 나오도록 하자

주민등록번호 성별 표시



1800년대 태어난 남자 - 9	1900년대 태어난 외국인 남자 - 5
여자 - 0	여자 - 6
1900년대 태어난 남자 - 1	2000년대 태어난 외국인 남자 - 7
여자 - 2	여자 - 8
2000년대 태어난 남자 - 3	
여자 - 4	

```
public class Main {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
  
        Person p1 = new Person();  
        Person p2 = new Person();  
  
        p1.RRN="950101-1234567";  
        p2.RRN="950101-6789123";  
  
        String p1Gender = p1.getGender();  
        String p2Gender = p2.getGender();  
  
        System.out.println(p1Gender);  
        System.out.println(p2Gender);  
  
        p1.showState();  
        p2.showState();  
    }  
}
```

```
Problems JavaDoc Declaration Console X Progress Git Staging  
<terminated> Main (15) [Java Application] C:\Users\USER545W\p2\pool\plugins\org.eclipse.j  
여  
여  
이름: null 주민번호: 950101-1234567      성별: 여  
이름: null 주민번호: 950101-6789123      성별: 여
```

1-2 문제풀이 (hard)

1-2 문제풀이 (hard)

Park Ju Byeong

Park Ju Byeong

1-3 실습문제 (expert)

앞서 만든 Person 클래스에 기능을 추가해보자

- 멤버 메서드 : `int getAge()` 주민번호를 이용해 나이를 반환한다.(만나이)
- 기존의 `showState` 메서드에 `getAge`를 이용하여 나이도 출력 하도록 하자.
- 성별 코드로 몇세기인지를 파악해야 정확한 나이 계산이 가능하다.

아래의 코드는 현재의 년,월,일을 int 형태로 얻는 코드이다.
나이계산에 활용하도록 하자.

```
//현재 년월일을 가져오기 위한 객체 생성
Calendar cal = Calendar.getInstance();

//현재의 년월일을 변수에 저장한다.
int nowYear = cal.get(Calendar.YEAR);
int nowMonth = cal.get(Calendar.MONTH)+1;
int nowDay = cal.get(Calendar.DAY_OF_MONTH);
```

```
Person p1 = new Person();
Person p2 = new Person();
```

```
p1.RRN="010512-3234567";
p2.RRN="950101-2789123";

p1.showState();
p2.showState();
```

```
<terminated> main (8) [Java Application] C:\Users\zest1\p2\p001\plugins\org.eclipse.justj.openjdk.m
이름:null          주민번호:010512-3234567      성별: 남 나이:21
이름:null          주민번호:950101-2789123      성별: 여 나이:28
```

1-3 문제풀이 (expert)

— 02

클래스변수(static 변수)

클래스 변수(static 변수)

- static 키워드를 사용하여 멤버변수 영역에 선언한다.
- 프로그램이 실행될 때 메모리에 올라간다(객체 생성 이전단계)
그러므로 객체 생성 없이 사용 가능하다.
- 객체들끼리 공유하는 변수이다.

```

-
3 public class Test
4 {
5     int iv=50;
6
7     static int cv;
8
9     void print()
10    {
11        int iv = 10;
12        System.out.println(iv);
13    }
14
15 }

```

```

public class Main {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        Test t1= new Test();

        t1.iv = 50; //일반적인 인스턴스변수 사용은 객체 생성을 해야만한다.
    }
}

```

```

public class Main {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

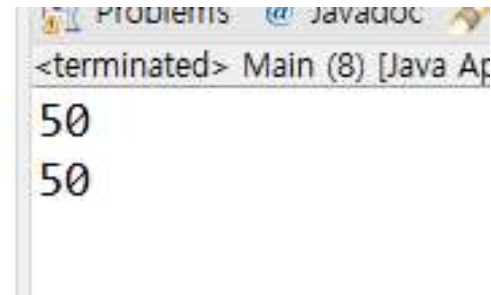
        Test.cv = 50;

        System.out.println(Test.cv);

    }
}

```

```
public class Main {  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
  
        Test.cv = 50;  
  
        System.out.println(Test.cv);  
  
        Test t1 = new Test();  
  
        System.out.println(t1.cv);  
    }  
}
```



<terminated> Main (8) [Java Ap...
50
50

객체를 통해 접근해도 된다.(권장안함)

인스턴스 변수 vs 클래스 변수

```
public class Test
{
    int iv;

    static int cv;
}
```

```
Test t1 = new Test();
Test t2 = new Test();

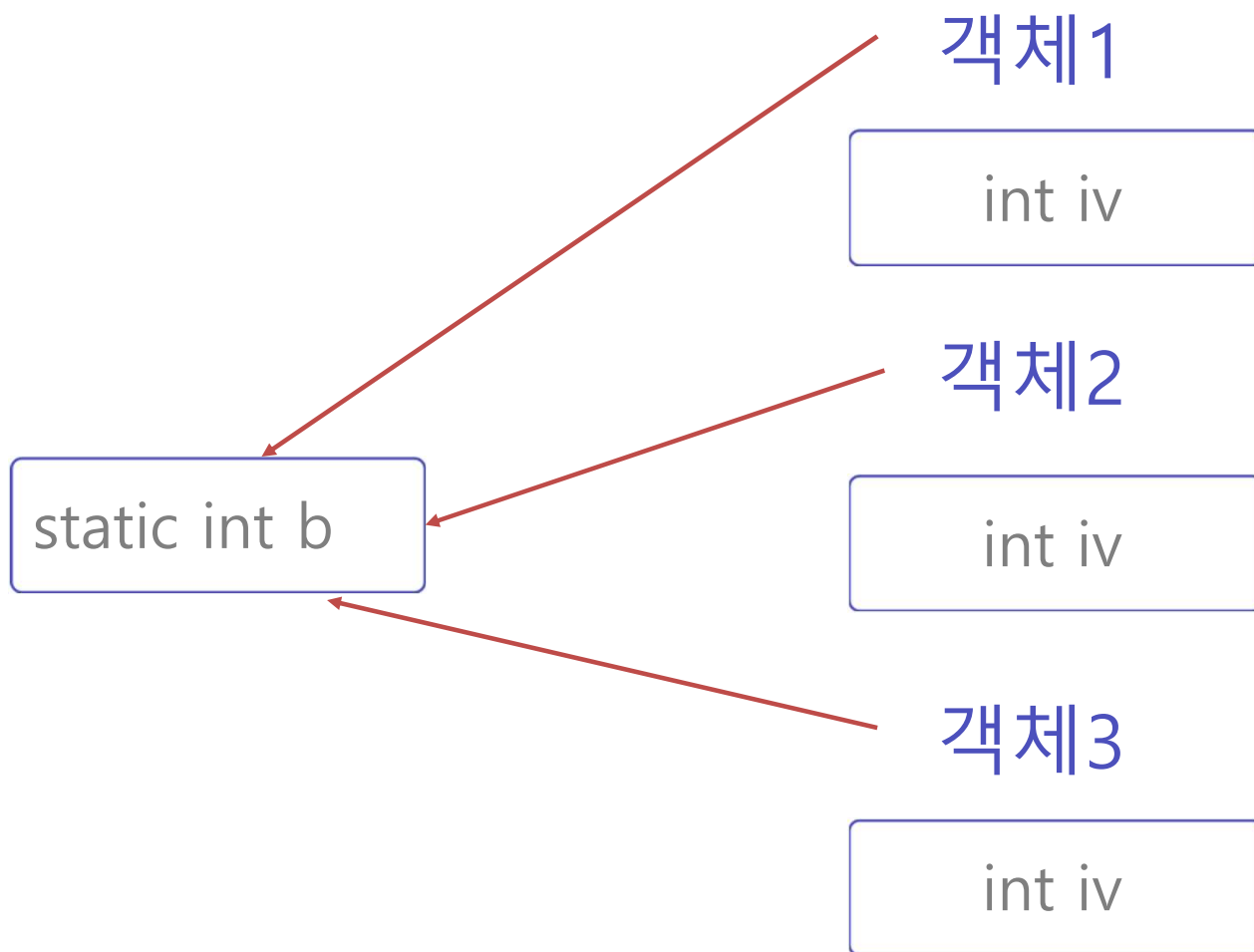
t1.iv = 10;
t2.iv = 20;
System.out.println(t1.iv);
System.out.println(t2.iv);
```

```
t1.cv = 10;
t2.cv = 20;
System.out.println(t1.cv);
System.out.println(t2.cv);
```

```
Console
1 <terminated
2 10
3 20
4 20
5 20
```



```
3 public class Test
4 {
5     int iv;
6
7     static int cv;
8     |
9 }
```





03

메모리 영역

JVM의 메모리 구조

메서드	클래스 정보, static
스택	지역변수, 연산 중간결과
힙	객체, 인스턴스 변수

클래스의 정보들이 각메모리 영역에 나누어서 관리된다.

메서드 영역

```
class Person {  
    void showState()  
    {  
        System.out.println("이름:" + name + "\t주민번호:" + RRN + "\t성별: " + sex);  
    }  
}
```

static 변수

스택 영역

힙 영역

p1.name = "짱구";

p2.name = "스펀지밥";



프로그램 실행시
만들어진다.

프로그램 실행중
객체 생성시
만들어진다.

```
class Marine {
    int hp=40;
    static int shootingRange=6;
}
```

```
Marine m1 = new Marine();
Marine m2 = new Marine();
Marine m3 = new Marine();
```

```
m1.shootingRange=10;
```

```
System.out.println(m2.shootingRange);
System.out.println(m3.shootingRange);
```

메서드	shootingRange
스택	
힙	m1, m2, m3

```
<terminated>
```

```
10
```

```
10
```

스코프와 라이프사이클

```
2 class test{
3     int iv= 10;
4     static int cv =1;
5     void a()
6     {
7         int i =5;
8         System.out.println(i);
9         System.out.println(cv);
10        System.out.println(iv);
11    }
12
13    void b()
14    {
15
16        System.out.println(i);
17        System.out.println(cv);
18        System.out.println(iv);
19    }
20 }
```

변수종류	스코프	라이프사이클
지역변수	해당지역	생성 : 해당 영역이 수행될때 소멸: 해당 영역 수행이 끝나고 즉시
인스턴스변수	접근제어에 따라 다름 최소 클래스 내부	생성 : 객체 생성시 소멸 : 객체 소멸시
클래스변수	접근제어에 따라 다름 최소 클래스 내부	생성 : 프로그램 실행시 소멸 : 프로그램 종료시

```

122 public class Main {
123
124     public static void main(String[] args) {
125
126
127         System.out.println(Marine.shootingRange);
128     }
129
130 }
131
132
133
134 }
135

```

Problems Javadoc Declaration Search Console X Git Staging History
 <terminated> main [Java Application] C:\Users\Wzest1\p2\pool\plugins\org.eclipse.justj.openjdk.hc
 6

클래스 변수는 객체 생성 없이 쓸 수 있다.

```

Marine m1 = new Marine();
System.out.println(Marine.shootingRange);
System.out.println(m1.shootingRange);

```

static의 의미와 맞지 않는 사용법이다.

클래스 메서드

```
class Marine {  
    int hp=40;  
    static int shootingRange=6;  
  
    void showState()  
    {  
        System.out.println("체력: "+hp+"\t 공격력  
    }  
  
    static void test()  
    {  
        System.out.println("클래스 메서드 호출!");  
    }  
}
```

```
Marine.test();
```

```
Marine m1 = new Marine();  
m1.test();
```


클래스 메서드 VS 인스턴스 메서드

```
class Marine {  
    int hp=40;  
    static int shootingRange=6;  
  
    void showState()  
    {  
        System.out.println("체력: "+hp+"\t"  
    }  
}
```

```
static void test() ————— 객체 생성 이전부터 사용가능하다.  
{  
    System.out.println("체력: "+ hp); ————— 인스턴스 멤버변수는 아직 생성 안되었을수  
}                                있다.
```

클래스 메서드

- 객체 생성없이 바로 사용
- 인스턴스 멤버변수 사용 불가
- 인스턴스 멤버메서드 사용불가
-

```
Math.random();
```

인스턴스 메서드

- 객체를 생성해야 사용가능
- 클래스, 인스턴스, 지역 변수 다 가능
- 클래스, 인스턴스 매서드 둘다 사용 가능

퀴즈

```
class test
{
    int a = 10;
    static int b = 20;
    int c = b;
    static int d = a;
    static void staticMethod()
    {
        System.out.println(a);
        System.out.println(b);
    }

    void instanceMethod()
    {
        System.out.println(a);
        System.out.println(b);
    }

    static void staticMethod2()
    {
        staticMethod();
        instanceMethod();
    }

    void instanceMethod2()
    {
        staticMethod();
        instanceMethod();
    }
}
```

가능!

인스턴스 변수는 아직 생성 안됐을수도 있다 ERROR!

ERROR!

OK!

OK!

OK!

OK!

ERROR!

OK!

OK!

참조형 타입의 매개변수

```
public class Person
{
    String name;
    String RRN;

    int money;

    void send(Person p1)
    {
        //매개변수로 넘어온 객체의 금액을 내가 가진 금액만큼 증가 시킨다.
        p1.money+=money;
        //나의 돈은 0원으로 만든다.
        money=0;
    }
}
```

```
public class Main {

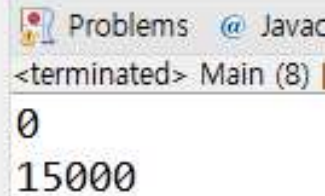
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Person p1 = new Person();
        Person p2 = new Person();

        p1.money = 5000;

        p2.money = 10000;

        //p1이 p2에게 돈을 준다.
        p1.send(p2);

        System.out.println(p1.money);
        System.out.println(p2.money);
    }
}
```



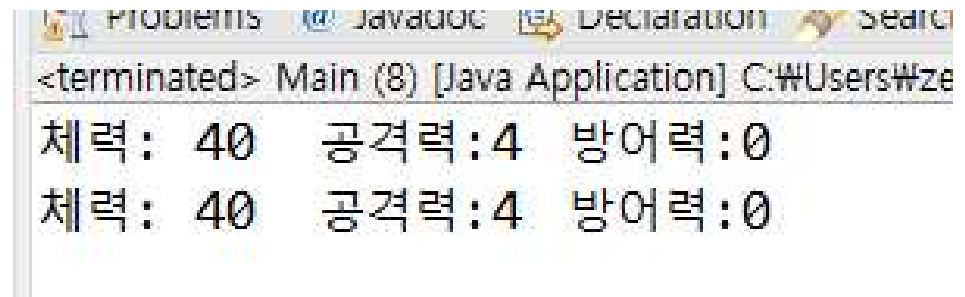
Problems @ Javac
<terminated> Main (8)
0
15000

2-1 실습문제 (normal)

Marine 클래스를 만들고 객체를 생성하여 사용해보자.

- 멤버변수 : int hp , int power(공격력) ,int armor(방어력)
- 멤버메서드 : showState() 객체의 상태를 표시 한다.

```
public class Main {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        Marine m1 = new Marine();  
        Marine m2 = new Marine();  
  
        m1.showState();  
        m2.showState();  
    }  
}
```



```
<terminated> Main (8) [Java Application] C:\#Users#ze  
체력: 40 공격력:4 방어력:0  
체력: 40 공격력:4 방어력:0
```

2-1 문제풀이 (normal)

2-2 실습문제 (hard)

Marine 클래스의 powerUp, armorUp 메서드를 만들자.

- 모든 객체가 같은 공격력과 방어력을 가지며 업그레이드시 모든 객체가 다같이 올라가야 한다.)
- power , armor 변수가 수정 되어야 한다.

```
Marine marine1 = new Marine();  
Marine marine2 = new Marine();  
  
marine1.powerUp();  
marine1.armorUp();  
  
marine1.showState();  
marine2.showState();
```

클래스명	Marine	
메서드	powerUp()	매개변수:없음 내용: 모든 유닛의 power를 1증가 시킨다. 리턴:없음
	armorUp()	매개변수:없음 내용: 모든 유닛의 armor를 1증가 시킨다. 리턴:없음

Problems Javadoc Declaration

<terminated> Main [Java Application] C:\Users\WU

```
체력: 40    공격력: 5    방어력: 1  
체력: 40    공격력: 5    방어력: 1
```


2-2 문제풀이 (hard)

Pat.

ng

Park J.

2-3 실습문제 (hard)

Marine 클래스에 attack 메서드를 구현하자

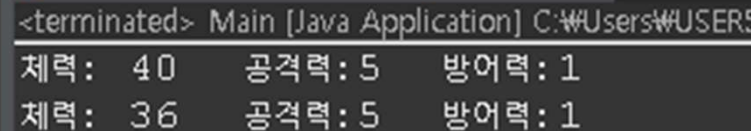
- 멤버메서드 : void attack(Marine target)
- 매개변수로 받은 Marine 객체의 체력을 감소시킨다.
- 공격시 (공격력 - 상대방의 방어력) 만큼 상대 체력을 감소 시킨다.

```
Marine marine1 = new Marine();
Marine marine2 = new Marine();

marine1.powerUp();
marine1.armorUp();

marine1.attack(marine2);

marine1.showState();
marine2.showState();
```



```
<terminated> Main [Java Application] C:\Users\WUSER5
체력: 40 공격력: 5 방어력: 1
체력: 36 공격력: 5 방어력: 1
```

2-3 문제풀이 (hard)

2-4 실습문제 (expert)

Marine 클래스에 기능을 추가하자

- 멤버변수: Point position(유닛의 위치정보를 저장)
int shootingRange(공격 사정거리)
- 멤버메서드 : int getDistance(Marine target) 매개변수로 받은 유닛과의 거리를 반환한다.
- attack 메서드에서 getDistance() 와 shootingRange 변수를 이용하여 공격 가능거리가 아니면 공격을 못한다고 출력하자.

Point 클래스를 사용하기 위해서 java.awt.Point 를 import 해야 한다.

제공근 : Math.sqrt 제공 : Math.pow(2,3) -> 2의3제곱 -> 8

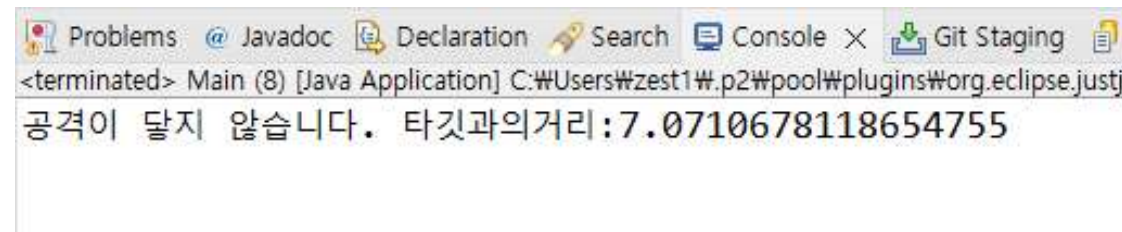
```
Marine m1 = new Marine();
Marine m2 = new Marine();

m1.position = new Point(2,2);

m2.position = new Point(7,7);

m1.attack(m2);|
```

$$\text{거리} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$



2-4 문제풀이 (expert)

Park Ju Byeong

Park Ju Byeong



THANK YOU



강사 박주병