1. 정수 배열 {6,9,1,3,5}를 선택 정렬을 이용하여 오름차순으로 출력 해봐라.

|  |
| --- |
| 소스  **public** **class** 연습01 {  **public** **static** **void** main(String[] args) {  **int**[] arry ={6,9,1,3,5};  **int** temp;  **for**(**int** i=0; i<arry.length-1;i++){  **for**(**int** j=i+1;j<arry.length;j++){  **if**(arry[i]>arry[j]){  temp=arry[i];  arry[i]=arry[j];  arry[j]=temp;  }  }  }  System.***out***.print("출력:");  **for**(**int** data :arry)  System.***out***.print(data+" ");  }  } |
| 결과 |

1. car 클래스를 작성해보자.

필드에는 차의 종류와 차의 색깔 변수를 생성하고,

메소드에는 차의 종류와 차의 색깔을 셋팅해주는 메소드를 작성해라.

메인에서 car클래스를 이용해 자신만의 객체를 생성 후 자신이 원하는 차 종과 차의 색깔을 지정해봐라.



|  |
| --- |
| 소스  **package** 객체;  **class** car{  String color;  String cars;  **public** **void** menu(){  System.***out***.println("차종:"+cars);  System.***out***.println("차 색상:"+color);  }  }  **public** **class** mycar {  **public** **static** **void** main(String[] args){  car mcar = **new** car();  car ccar = **new** car();  ccar.color="핑크";  ccar.cars="소나타";  ccar.menu();  System.***out***.println("--------------------");  mcar.color="파랑";  mcar.cars="제네시스";  mcar.menu();  }  } |
| 실행결과 |

1. 간단한 인스타그램 클래스를 작성해보자!

|  |  |
| --- | --- |
| Class | instagram |
| Field | (+):**int** like;  (+):**String** talk; |
| 생성자 |  |
| Method | **(+): void** talk\_like()  **(+): void** talk\_hate()  **(+): void** talk\_upload(String)  **(+): void** print() |

-talk\_like 메소드는 호출시 필드에 있는 like가 증가합니다.

-talk\_hate 메소드는 호출시 필드에 있는 like가 감소합니다.

-talk\_upload 메소드는 매개변수를 통해 글을 작성 후 필드에있는 talk에 저장해줍니다.

-print 메소드는 현재 필드에 저장된 글과 좋아요 개수를 출력해줍니다.

-모든 값들은 테스트 클래스에서 설정해줍니다.

\*테스트 클래스 생성 후 객체 생성해서 테스트한 화면을 보여주세요.



|  |
| --- |
| 소스  **package** 객체;  **import** java.util.\*;  **class** ins {  **public** **int** like=0;  **public** String talk;  **public** **void** talk\_like() {  //int a=0;  **this**.like++;//=++a;  }  **public** **void** talk\_hate() {    **this**.like--;  }  **public** **void** talk\_upload( String str ) {  **this**.talk=str;  }  **public** **void** print() {  System.***out***.println(talk);  System.***out***.println("좋아요 개수는:"+like);  }  }  **public** **class** instagram {  **public** **static** **void** main(String[] args){  Scanner key = **new** Scanner(System.***in***);  String str;  ins myins = **new** ins();  System.***out***.println("메시지 입력");  myins.talk\_upload( str =key.next());  myins.talk\_like();  myins.talk\_like();  myins.talk\_hate();  myins.print();    }  } |
| 결과 |

1. 간단한 은행 클래스를 작성해보자!

|  |  |
| --- | --- |
| Class | Bank |
| Field | (+):**int** pay;  (+):**int** real\_pay;  (+):**double** tex;  (+):**String** name; |
| 생성자 |  |
| Method | **(+): void** setReal\_pay()  **(+): void** setTex() |

-setReal\_pay 메소드는 세금을 제외한 금액을 정수형 변수 real\_pay에 저장하는 메소드입니다.

-setTest 메소드는 원래 월급에 4.7%센트 적용합니다.

-모든 값들은 테스트 클래스에서 설정해줍니다.

\*테스트 클래스 생성 후 객체 생성해서 테스트한 화면을 보여주세요.



|  |
| --- |
| 소스  **import** java.util.\*;  **class** mBank{  **public** **int** pay;  **public** **int** real\_pay;  **public** **double** tex;  **public** String name;  **public** **void** setReal\_pay(){  **this**.real\_pay=(**int**)(pay-tex);  }  **public** **void** setTex(){  **this**.tex=pay\*0.047;  }  }  **public** **class** Bank {  **public** **static** **void** main(String [] args){  Scanner key = **new** Scanner(System.***in***);  mBank bank = **new** mBank();  bank.pay=10000000;  bank.name="이상우";  bank.setTex();  bank.setReal\_pay();  System.***out***.println("이름:"+bank.name);  System.***out***.println("웙급:"+bank.pay);  System.***out***.println("세금:"+bank.tex);  System.***out***.println("세금을 제외한 실제로 지급될 월급:"+bank.real\_pay);  }  } |
| 실행결과 |

1. 날짜를 나타내는 클래스 Date 와 객체를 생성하고 테스트 하는 DateTest 클래스를 다음과 같이 정의하시오

* 클래스 Date

1. 필드:

* 연도, 월, 일

1. 메소드

* void display1() : 날짜를 “2012.7.12” 형식으로 출력
* void display2() : 날짜를 “July 12, 2012” ,형식으로 출력

힌트) 월에 해당하는 영 단어는 문자열 배열로 초기화 한 후 사용

* DateTest 클래스

class DateTest{

public static void main(String[] args){

//Date객체 선언 및 생성

//년, 월, 일 필드값을 키보드로 입력하여 초기화. 단, 모두 정수로 입력

// display1( ), display2( ) 메소드 호출

}

}

|  |
| --- |
| 소스  **package** 객체;  **import** java.util.\*;  **class** Datas {  **int** year;  **int** m;  **int** w;  **void** display1() {  System.***out***.println(year + "." + m + "." + w);  }  **void** display2() {  String[] Month = { "January", "February", "March", "April", "May", "June", "July", "August", "September",  "October", "November", "December" };  System.***out***.println( Month[(m - 1)] + w+"."+year);  }  }  **public** **class** data {  **public** **static** **void** main(String[] args) {  Scanner key = **new** Scanner(System.***in***);  Datas ta = **new** Datas();  System.***out***.println("년도를 입력하세요.");  ta.year = key.nextInt();  System.***out***.println("월를 입력하세요.");  ta.m = key.nextInt();  System.***out***.println("일를 입력하세요.");  ta.w = key.nextInt();  ta.display1();  ta.display2();  }  } |
| 실행 결과 |

1. 영화 Movie 클래스를 정의하여 보자. Movie 클래스는 영화 제목, 평점, 감독, 발표된 연도 등의 필드를 가진다. 영화의 모든 정보를 화면에 출력하는 write()라는 메소드를 구현하라. Movie 클래스를 작성하고 객체를 생성하여서 테스트하라.

|  |
| --- |
| 소스  **package** 객체;  **class** mov{  String Mname;  **double** point;  String movs;  **int** year;  **void** write(){  System.***out***.println("영화 이름:"+Mname);  System.***out***.println("평점:"+point);  System.***out***.println("영화 감독:"+movs);  System.***out***.println("발표 년도:"+year);    }    }  **public** **class** Movie {  **public** **static** **void** main(String [] args){  mov mov =**new** mov();  mov.Mname="노잼영화";  mov.point=9.9;  mov.movs="이상우";  mov.year=2017;  mov.write();    }  } |
| 실행 결과 |

1. 메인 클래스에서 정수를 입력 받아 객체를 이용해서 만약 입력숫자가 3이라면 4 -4가 나오게끔 클래스를 작성하시오



|  |
| --- |
| 소스  **package** 객체;  **import** java.util.\*;  **class** ber {  **static** **int** pluse(**int** a) {  **return** ++a;  }  **static** **int** minus(**int** a) {  **return** -a;  }  **public** **int** min(**int** a) {  **return** ++a;  }  }  **public** **class** number {  **public** **static** **void** main(String[] args) {  Scanner key = **new** Scanner(System.***in***);  System.***out***.println("정수 입력");  **int** a = key.nextInt();  System.***out***.println("입력 값" + a);  a=ber.*pluse*(a);  System.***out***.println("plus 메소드로 변경 한 값 : "+ a);  a=ber.*minus*(a);  System.***out***.println("minus 메소드로 변경 한 값 : "+ a);  }  } |
| 실행 결과 |

1. 거리,시간,속력 구하기

-거리,시간,속력을 각각 입력받아요(정수형 변수 3개)

-거리=(시간\*속력),시간=(거리/속력),속력=(거리/시간)

-거리메소드,시간메소드,속력메소드 3개를 만들어요.

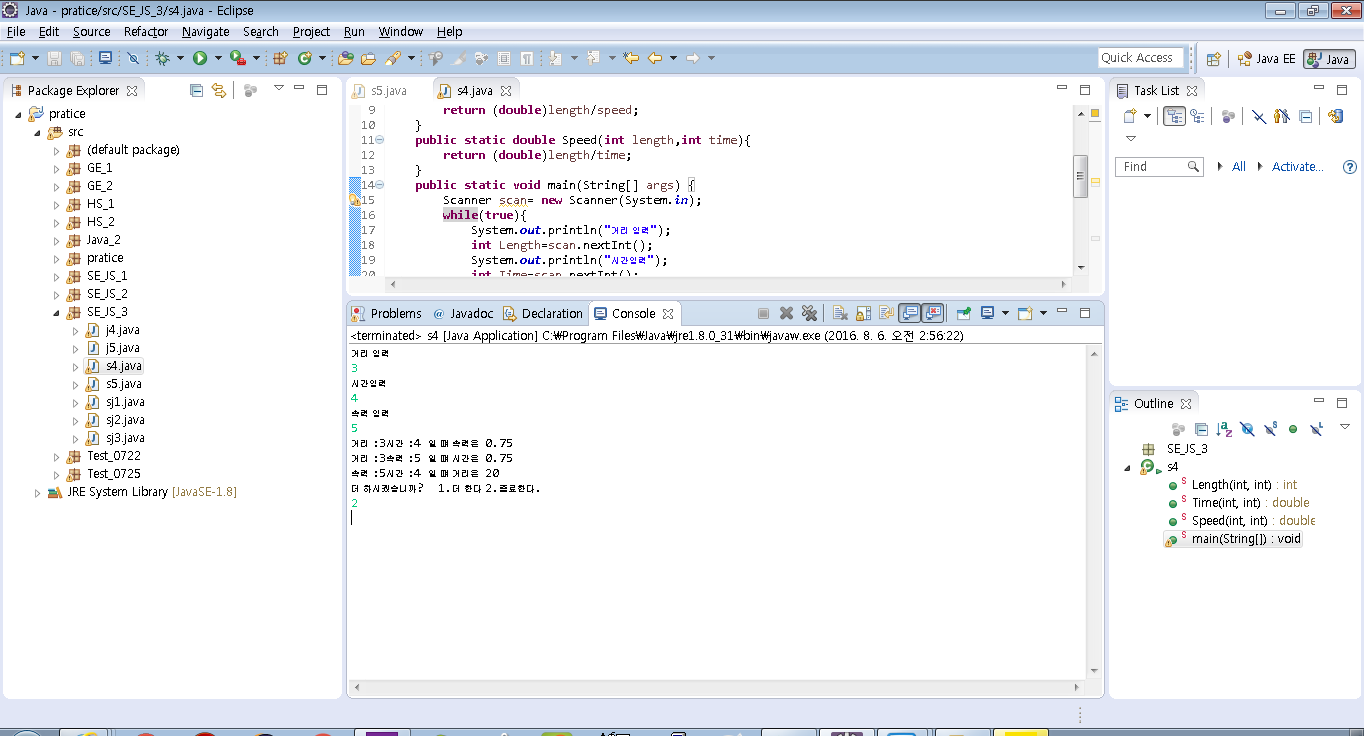
-입력받은 3개의 값을 2개씩 묶어서 메소드로 보냅니다.

(거리,시간),(거리,속력),(속력,시간)

-묶은 변수에 맞는 메소드를 보내 공식에 맞는 값을 리턴합니다.

-리턴 값을 출력합니다.

-그리고 더 할건지 종료할건지 if조건문을 사용합니다.

●실행 결과●\

|  |
| --- |
| 소스  **package** 객체;  **import** java.util.\*;  **class** sps {  **static** **int** *km*;  **static** **int** *time*;  **static** **int** *speed*;  **public** **double** tim() {  **return** (**double**) *km* / (**double**) *speed*;  }  **public** **double** sp() {  **return** (**double**) *km* / (**double**) *time*;  }  **public** **double** kiro() {  **return** (*time* \* *speed*);  }  }  **public** **class** spd {  **public** **static** **void** main(String[] args) {  sps obj = **new** sps();  Scanner key = **new** Scanner(System.***in***);  **while** (**true**) {  System.***out***.println("거리 입력");  sps.*km* = key.nextInt();  System.***out***.println("시간 입력");  sps.*time* = key.nextInt();  System.***out***.println("속력 입력");  sps.*speed* = key.nextInt();  System.***out***.println("거리:" + sps.*km* + " 시간:" + sps.*time* + " 일 때 속력은: " + obj.sp());  System.***out***.println("거리:" + sps.*km* + " 속력:" + sps.*speed* + " 일 때 시간은: " + obj.tim());  System.***out***.println("속력:" + sps.*speed* + " 시간:" + sps.*time* + " 일 때 거리는: " + obj.kiro());  System.***out***.println("더 하시겟습니까? 1.더 한다 2.종료한다");  **int** me=key.nextInt();  **if**(me==2)  **break**;  }  }  } |
| 소스 |

1. Factorial!

-factorial값을 입력합니다.

-값을 메소드로 보냅니다 ex )　num=5; fact(num);

-메소드에서 매개변수로 받은 값의 팩토리얼을 구한 값을 리턴합니다.

-리턴 값을 출력 합니다.

●실행결과●

|  |
| --- |
| 소스  **import** java.util.\*;  **class** fact {  **public** **int** facts(**int** num) {  **int** j=1;  **for**(**int** i=1;i<=num;i++){  j\*=i;  }  **return** j;  }  }  **public** **class** not {  **public** **static** **void** main(String[] args) {  Scanner key = **new** Scanner(System.***in***);  System.***out***.print("fact 값 입력:");  **int** ber=key.nextInt();  fact ff =**new** fact();  System.***out***.print("5! 은:"+ff.facts(ber));  }  } |
| 실행 결과 |

1. Up, Down 문제
   1. Up\_Down 클래스를 생성하여 play 메소드에서 답이 나올 때 까지 계속 입력 받는다.
   2. 메인 클래스에서 랜덤 값 생성 후 Up\_Down 클래스로 넘겨준다. (랜덤 값 범위는 1~10까지 이다.)



|  |
| --- |
| 소스  **package** 객체;  **import** java.util.\*;  **class** updown{  **public** **void** up\_down(**int** nan , **int** s){  **if**(nan >s){  System.***out***.println("UP");  }  **else** **if**(nan <s){  System.***out***.println("DOWN");  }    }  }  **public** **class** dom {  **public** **static** **void** main(String[] args) {  Scanner key = **new** Scanner(System.***in***);  updown ud = **new** updown();  **int** su = (**int**)(Math.*random*()\*10+1);  **while**(**true**){  System.***out***.println("값 입력");    **int** gg=key.nextInt();  System.***out***.println("...");  ud.up\_down(su , gg);  **if**(su==gg){  System.***out***.println("정답");  **break**;  }  }  }  } |
| 결과 |

1. 배열을 종합적으로 메소드와 연관 지어 코딩 해보자
   1. Array\_Set 클래스를 작성해라
      1. 필드
         1. 필드에는 사이즈 10개짜리 배열을 2개 선언 및 생성하고 하나의 배열에는 사이즈 20짜리 배열을 선언 및 생성해라.
      2. 생성자
         1. *메인으로부터 넘겨받은 배열을 필드에 있는 배열에 대입해줘라*
      3. *메소드*
         1. *Printsort()*
            1. *이 메소드를 호출 할 경우 선언된 모든 배열을 출력해줘라*
         2. *Sort(int in)*
            1. *메인으로부터 in값(정렬을 원하는 배열의 번호)을 넘겨받아라*
            2. *정렬을 원하는 배열을 정렬해줘라*
         3. *Hapsort()*
            1. *선언된 두개의 배열을 하나의 배열에 합쳐줘라*
   2. Test 클래스

**int** sort1[] = { 1, 19, 16, 5, 8, 17, 2, 6, 11, 10 };

**int** sort2[] = { 18, 3, 9, 14, 20, 12, 15, 7, 13, 4 };

이와 같은 배열 두개를 선언해라

Array\_set 클래스에 대한 객체를 생성하면서 sort1, sort2 배열을 넘겨줘라

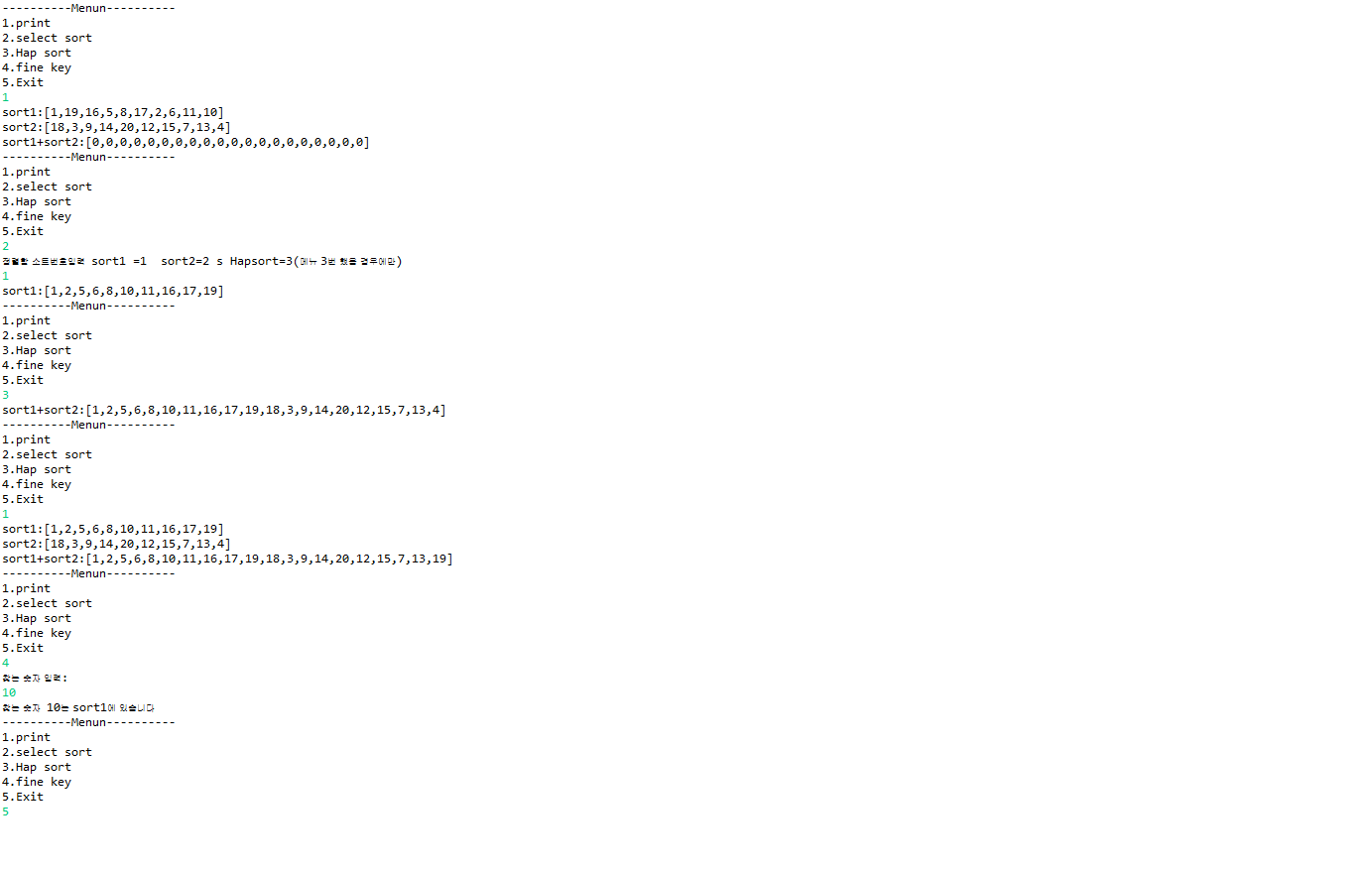
5가 입력되기 전까지는 계속 무한 반복을 진행해라

1번을 입력할 경우 선언된 배열을 모두 출력해줘라

2번을 입력할 경우 원하는 배열을 정렬해줘라

3번을 입력할 경우 선언된 배열 두개를 하나로 합쳐줘라

4번을 입력할 경우 검색을 원하는 값을 입력 받고 어디 배열에 있는지 알려줘라



|  |
| --- |
| 소스  **package** 객체;  **import** java.util.Scanner;  **class** Array\_Set {  **int**[] arry1 = **new** **int**[10];  **int**[] arry2 = **new** **int**[10];  **int**[] ar20 = **new** **int**[20];  **public** Array\_Set(**int**[] ar, **int**[] ar2) {  arry1 = ar;  arry2 = ar2;  }  **public** **void** Printsort() {  **if** (ar20[0] == 0) {  System.***out***.print("arry1 :");  **for** (**int** i : arry1) {  System.***out***.print(i + " ");  }  System.***out***.println();  System.***out***.print("arry2 :");  **for** (**int** i2 : arry2) {  System.***out***.print(i2 + " ");  }  } **else** {  System.***out***.print("arry1 :");  **for** (**int** i : arry1) {  System.***out***.print(i + " ");  }  System.***out***.println();  System.***out***.print("arry2 :");  **for** (**int** i2 : arry2) {  System.***out***.print(i2 + " ");  }  System.***out***.println();  System.***out***.print("arry1+arry2 :");  **for** (**int** i3 : ar20) {  System.***out***.print(i3 + " ");  }  }  System.***out***.println();  }  **public** **void** Sort(**int** in) {  **switch** (in) {  **case** 1:  **int** temp;  **for** (**int** i = 0; i < arry1.length - 1; i++) {  **for** (**int** j = i + 1; j < arry1.length; j++) {  **if** (arry1[i] > arry1[j]) {  temp = arry1[i];  arry1[i] = arry1[j];  arry1[j] = temp;  }  }  }  System.***out***.print("arry1 :");  **for** (**int** i : arry1) {  System.***out***.print(i + " ");  }  System.***out***.println();  **break**;  **case** 2:  **for** (**int** i = 0; i < arry2.length - 1; i++) {  **for** (**int** j = i + 1; j < arry2.length; j++) {  **if** (arry2[i] > arry2[j]) {  temp = arry2[i];  arry2[i] = arry2[j];  arry2[j] = temp;  }  }  }  System.***out***.print("arry2 :");  **for** (**int** i2 : arry2) {  System.***out***.print(i2 + " ");  }  System.***out***.println();  **break**;  **case** 3:  **if** (ar20[0] == 0) {  System.***out***.println(" 아직 입력되지 않았습니다.");  **break**;  }  **else** {  **for** (**int** i = 0; i < ar20.length - 1; i++) {  **for** (**int** j = i + 1; j < ar20.length; j++) {  **if** (ar20[i] > ar20[j]) {  temp = ar20[i];  ar20[i] = ar20[j];  ar20[j] = temp;  }  }  }  }  System.***out***.print("arry1+arry2 :");  **for** (**int** i3 : ar20) {  System.***out***.print(i3 + " ");  }  System.***out***.println();  **break**;  }  }  **public** **void** Hapsort() {  **int** a = 10;  **for** (**int** i = 0; i < arry1.length; i++) {  ar20[i] = arry1[i];  ar20[a] = arry2[i];  a++;  **if** (i == 19)  i = 0;  }  System.***out***.print("arry1+arry2 :");  **for** (**int** data : ar20)  System.***out***.print(data + " ");  System.***out***.println();  }  **public** **void** search(Scanner key){    **int** aar1 = 0, aar2 = 0 ;  **int** s = 0, s1 = 0 ;  System.***out***.print("찾는 숫자 입력 ");  **int** aa = key.nextInt();  **for** (**int** i = 0; i < arry1.length; i++) {  **if** (arry1[i] == aa) {  aar1++;  s = i;  **break**;  }  **else** **if** (arry2[i] == aa) {  aar2++;  s1 = i;  **break**;  }  }    **if** (aar1 > 0)  System.***out***.println("찾는 숫자" + aa + "는 arry1 배열" + (s + 1) + "번째에 있습니다.");  **else** **if**(aar1 == 0)  System.***out***.println("찾는 숫자가 없습니다.");    **else** **if** (aar2 > 0)  System.***out***.println("찾는 숫자" + aa + "는 arry2 배열" + (s1 + 1) + "번째에 있습니다.");  **else** **if**(aar2 == 0)  System.***out***.println("찾는 숫자가 없습니다.");  }  }  **public** **class** Set {  **public** **static** **void** main(String[] args) {  Scanner key = **new** Scanner(System.***in***);  **int**[] ar = **new** **int**[] { 1, 19, 16, 5, 8, 17, 2, 6, 11, 10 };  **int**[] ar2 = **new** **int**[] { 18, 3, 9, 14, 20, 12, 15, 7, 13, 4 };  Array\_Set arr = **new** Array\_Set(ar, ar2);  **while** (**true**) {  System.***out***.println("-------------------");  System.***out***.println("매뉴를 선택하세요.");  System.***out***.println("1. 배열 출력 ");  System.***out***.println("2.배열 정렬");  System.***out***.println("3. 배열 합치기");  System.***out***.println("4. 원소 위치 검색");  System.***out***.println("5. 종료");  System.***out***.println("-------------------");  **int** menu = key.nextInt();  **if** (menu == 5)  **break**;  **switch** (menu) {  **case** 1:  arr.Printsort();  **break**;  **case** 2:  System.***out***.print("정렬할 배열의 번호 입력 1.arry1 2.arry2 3.ar20(3번 매뉴를 사용한 경우 만 사용가능합니다.");  **int** as = key.nextInt();  arr.Sort(as);  **break**;  **case** 3:  arr.Hapsort();  **break**;  **case** 4:  arr.search(key);  **break**;  }  }  }  } |
| 실행 결과 |