|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **[유헬스2\_2주\_실습과제]** | **학번:** | **20175105** | **이름:** | **곽영주** |

* **메소드 오버로딩** : 메소드 이름은 같고 매개변수 타입과 개수, 반환타입을 모두 다르게 정의

// 2개의 정수의 합을 구하는 sum()과 3개의 정수의 합을 구하는 sum() 메소드 오버로딩

**public** **class** Calculation {

//클래스는 필드 또는 메소드만으로도 구성 가능

**void** sum(**int** a, **int** b){ //정수형 매개변수 2개를 가지는 메소드- 오버로딩

System.***out***.println(a+b);

}

**void** sum(**int** a, **int** b, **int** c){ //정수형 매개변수 3개를 가지는 메소드 – 오버로딩

System.***out***.println(a+b+c);

}

//작성된 클래스를 테스트 하기 위해 별도의 테스트 클래스를 작성하지 않아도 됨

//작성된 클래스 내부에 main() 메소드를 통하여 테스트

**public** **static** **void** main(String args[]){

Calculation obj=**new** Calculation();

obj.sum(10,20,30); //인수 타입과 개수가 일치하는 오버로딩 된 메소드 호출

obj.sum(10,20);

}

}

* **프로그램 과제 1** : Calculation 클래스에 다음과 같은 메소드를 오버로딩을 이용하여 정의하시오

1개의 double형 배열을 매개변수로 받아 합을 계산하여 반환하는 sum() 메소드

1개의 int 형 배열을 매개변수로 받아 합을 계산하여 반환하는 sum() 메소드

|  |
| --- |
| **[추가된 메소드]**  **double** sum(**double**[] dim){  **double** hap=0;  **for**(**double** data : dim)  hap += data;  **return** hap;  }    **int** sum(**int**[] dim){  **int** hap=0;  **for**(**int** data : dim)  hap += data;  **return** hap;  } |

* **프로그램 과제 2** : 추가된 메소드를 호출하여 반환 받은 결과를 출력하는 문장을 main()에 추가. 단, 매개변수로 전달되는 배열은 무명 배열을 사용할 것

|  |
| --- |
| **[추가된 문장]**  System.***out***.println(obj.sum(**new** **double**[] {2.1, 6.5, 3.5}));  System.***out***.println(obj.sum(**new** **int**[] {2,8,6})); |
| **[실행 결과]** |

* **생성자, 접근자, 설정자, 객체 배열**
* 생성자 : 객체 초기화, 클래스 이름과 동일해야 함
* 설정자 메소드 : 필드 초기화 & 변경
* 접근자 메소드 : 필드 반환

**import** java.util.\*;

**public** **class** Distance {

**private** String name;

**private** **int** distance;

**public** Distance(){ //디폴트 생성자

**this**("", 0); //오버로딩 된 생성자 호출

}

**public** Distance(String name, **int** distance) { //생성자

**this**.name = name;

**this**.distance = distance;

}

**public** String getName() { //접근자(gtter method)

**return** name;

}

**public** **void** setName(String name) { //설정자(setter method)

**this**.name = name;

}

**public** **int** getDistance() {

**return** distance;

}

**public** **void** setDistance(**int** distance) {

**this**.distance = distance;

}

**public** String toString(){

**return** "name = " + name + "\tdistance = " + distance;

}

//객체배열을 매개변수로 받아 가장 큰 거리값을 갖는 객체 반환

//Distance 클래스안의 main()에서 객체를 생성하고 테스트

**public** **static** **void** main(String[] args) {

Distance[] obj=**new** Distance[5]; //객체 배열 선언과 생성

Scanner in=**new** Scanner(System.***in***);

**for**(**int** i=0; i<obj.length;i++){

System.***out***.print("[" + (i+1)+"]" + " name과 distance 입력 : ");

obj[i]=**new** Distance(in.next(), in.nextInt() ); //객체 생성시 생성자 호출

}

**for**(Distance instance : obj) //객체 배열 내용 출력

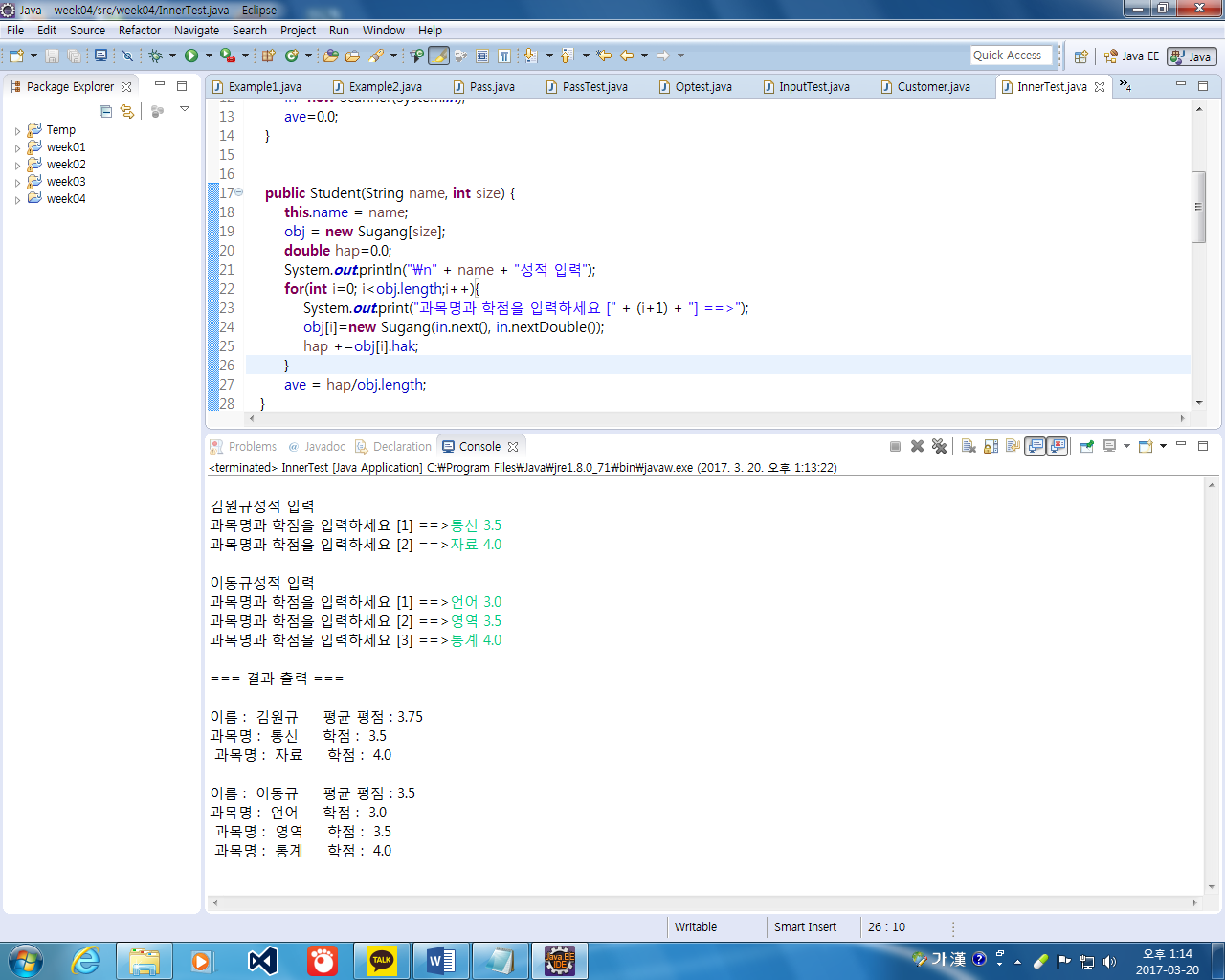
System.***out***.println(instance);

}

}

|  |
| --- |
| **[실행 결과]** |

* **프로그램 과제 3 :** 다음과 같이 성적을 처리하는 프로그램을 작성하세요. 출력은 제시된 결과를 참고할 것



**class** Student{

//필드 : 이름(String), 평균(double), 수강과목과 학점(Sugang[]) , 전용멤버 – 본인작성

//이름과 평균 필드에 대한 getter 메소드, 이름 필드에 대한 setter 메소드 – 본인작성

Scanner in;

//인스턴스 블럭 초기화

{

in=**new** Scanner(System.***in***);

ave=0.0;

}

//생성자 : 이름과 수강과목수를 매개변수로 받는다

// 수강과목수 만큼 객체배열을 생성한 후 과목명과 학점을 입력 받는다

//학점 평균을 계산하여 필드 ave에 저장

//본인 작성

//객체 내용 문자열로 반환 – 본인작성

**public** String toString()

{

}

}

//수강 과목과 학점을 처리하기 위한 클래스

**class** Sugang{

//필드 : 과목명(String), 학점(double), 전용멤버 – 본인작성

//모든 필드에 대한 getter 와 setter 메소드

//생성자 : 매개변수로 받은 값을 필드로 초기화, 과목명과 학점 – 본인작성

//객체 내용을 문자열로 반환 : 과목명과 학점 – 본인작성

}

**public** **class** InnerTest {

**public** **static** **void** main(String[] args) {

//객체 배열 선언 & 생성 & 초기화

Student[] st={**new** Student("김원규", 2), **new** Student("이동규", 3)};

System.***out***.println("\n=== 결과 출력 ===");

**for**(Student temp : st){

System.***out***.println(temp);

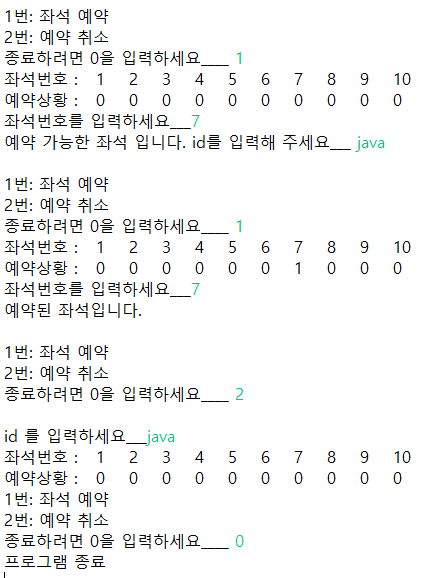
}

}

}

|  |
| --- |
| **[프로그램 소스]**  **public** **class** Student {  **private** String name;  **private** **double** ave;  **private** Sugang[] su;  Scanner in;  {  in = **new** Scanner(System.***in***);  ave = 0.0;  }    **public** String getName(){  **return** name;  }    **public** **double** getAve(){  **return** ave;  }    **public** **void** setName(String name){  **this**.name = name;  }      **public** Student(String name, **int** cnt){  **this**.name = name;  su = **new** Sugang[cnt];  System.***out***.println(name + " 성적 입력");  **for**(**int** i=0; i<su.length; i++){  System.***out***.println("과목명과 학점을 입력하세요 ["+(i+1)+"] ==> ");  su[i] = **new** Sugang(in.next(),in.nextDouble());  ave += su[i].getHak(); // 합 계산  }  ave = ave / su.length; // 평균 계산  }    **public** String toString(){  String str = "이름 : " + name + "\t평균 평점 : " + ave;  **for**(Sugang data : su)  str += data.toString();  **return** str;  }  }  **public** **class** Sugang {  **private** String obj;  **private** **double** hak;    **public** String getObj(){  **return** obj;  }  **public** **void** setObj(String obj){  **this**.obj = obj;  }    **public** **double** getHak(){  **return** hak;  }  **public** **void** setHak(**double** hak){  **this**.hak = hak;  }    **public** Sugang(String obj, **double** hak){  **this**.obj = obj;  **this**.hak = hak;  }    **public** String toString(){  **return** "\n과목명 : " + obj + "\t학점 : " + hak;  }  }  **public** **class** InnerTest {  **public** **static** **void** main(String[] args){  Student[] st = {**new** Student("김원규",2), **new** Student("이동규",3)};    System.***out***.println("\n=== 결과 출력 ===");  **for**(Student data : st)  System.***out***.println(data);  }  } |
| **[실행 결과]** |

* **프로그램 과제 4 :** 제시된 결과처럼 실행되는 좌석예약 프로그램을 작성하시오**.** 단, 객체 배열을 사용할 것



* **1번 메뉴를 선택하면 원하는 좌석번호를 입력하고 예약 가능여부를 처리한다**
* **2번 메뉴를 선택하면 취소하고자 하는 좌석의 id를 입력 받아 예약 상황을 0으로 값을 처리한다. 만약 해당 id가 없으면 “해당 아이디 없음”을 출력한다.**
* **0번을 선택하면 반복문을 종료한다.**

**public** **class** Reservation {

//필드 선언 : 좌석번호, id, 예약상황 – 전용필드

//각 필드에 대한 설정자와 접근자

**}**

**public** **class** ReservationTest {

public static void main(String[] args){

// Reservation 객체 배열 선언 & 생성, 배열 크기 10

//반복문을 사용하여 메뉴처리

}

**}**

|  |
| --- |
| **[프로그램 소스]**  **public** **class** Reservation {  **private** **int** num;  **private** String id;  **private** **int** res;    **public** Reservation(**int** num, **int** res){  **this**.num = num;  **this**.res = res;  }    **public** **int** getNum(){  **return** num;  }  **public** **void** setNum(**int** num){  **this**.num = num;  }    **public** String getId(){  **return** id;  }  **public** **void** setId(String id){  **this**.id = id;  }    **public** **int** getRes(){  **return** res;  }  **public** **void** setRes(**int** res){  **this**.res = res;  }  }  **import** java.util.\*;  **public** **class** ReservationTest {  **public** **static** **void** main(String[] args){  Reservation[] obj = **new** Reservation[10];  Scanner key = **new** Scanner(System.***in***);  **int** num; //좌석 번호 입력    **for**(**int** i=0; i<obj.length; i++){  obj[i] = **new** Reservation(i+1,0);  }    **while**(**true**){  System.***out***.println("\n1번: 좌석 예약");  System.***out***.println("2번: 예약 취소");  System.***out***.println("종료하려먼 0을 입력하세요\_\_ ");  **int** a = key.nextInt();// 1번, 2번, 0번 입력    **if**(a == 1){  System.***out***.print("좌석 번호 : ");  **for**(**int** i=0; i<obj.length; i++){// 좌석 번호  System.***out***.print(" "+obj[i].getNum());  }  System.***out***.print("\n예약 상황 : ");  **for**(**int** i=0; i<obj.length; i++){// 예약 상황  System.***out***.print(" "+obj[i].getRes());  }  System.***out***.print("\n좌석번호를 입력하세요\_\_");  num = key.nextInt();    **if**(obj[num-1].getRes() == 0){  obj[num-1].setRes(1);  System.***out***.print("예약 가능한 좌석 입니다. id를 입력해 주세요\_\_");  obj[num-1].setId(key.next());  }  **else** **if**(obj[num-1].getRes() != 0){  System.***out***.print("예약된 좌석입니다.");  }  }    **if**(a == 2){  System.***out***.println("\nid를 입력하세요\_\_");  String str = key.next();    **for**(**int** i=0; i<obj.length; i++){  **if**(str.equals(obj[i].getId())){  obj[i].setRes(0);  }  }    System.***out***.print("좌석 번호 : ");  **for**(**int** i=0; i<obj.length; i++){// 좌석 번호  System.***out***.print(" "+obj[i].getNum());  }  System.***out***.print("\n예약 상황 : ");  **for**(**int** i=0; i<obj.length; i++){// 예약 상황  System.***out***.print(" "+obj[i].getRes());  }  }    **if**(a == 0){  System.***out***.println("프로그램 종료");  **break**;  }  }  }  } |
| **[실행 결과]** |