자바 프로그래밍 11장

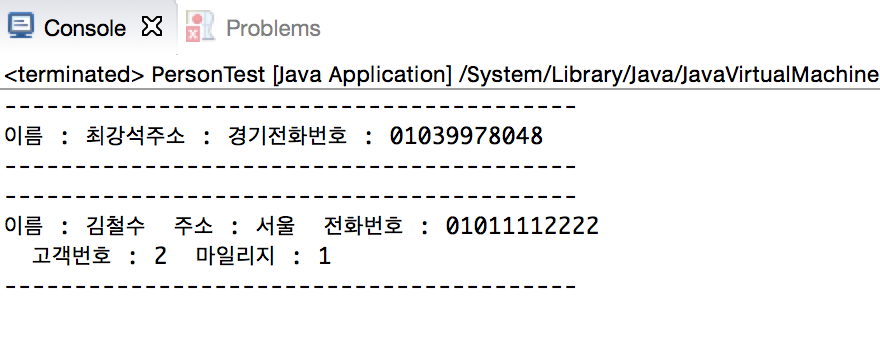
|  |  |
| --- | --- |
| 분반 | 1분반 |
| 학번 | 2012136132 |
| 이름 | 최강석 |

264PAGE 2번문제

1. 문제 : Person클래스를 설계하라. Person클래스는 이름,주소,전화번호를 필드로  
   가진다. 하나이상의 생성자를 정의하고 각필드에 대하여 접근자와 설정자 메소드를 작성하라.  
   이어서 Person을 상속받아서 customer를 작성하여 보자.  
   customer는 고객번호와 마일리지를 필드로 가지고 있다. 한개이상의 생성자를 작성하고 적절한 접근자 메소드와   
   설정자 메소드를 작성하라.
2. 알고리즘 :
   1. person에 대한 클래스를 만들어준다. (이름, 주소, 전화번호를 필드로 가짐)
   2. 생성자, 설정자, 접근자에 대한 정의를 해준다.
   3. person필드를 출력하는 함수를 정의해준다.
   4. Customer클래스를 만들어준다 (person을 상속받는다.)
   5. 생성자, 설정자, 접근자에 대한 정의를 해준다.
   6. Customer필드를 출력하는 함수를 정의해준다.
   7. 메인에 객체를 생성하고 Print()를 호출해준다.
3. 코드 :

|  |
| --- |
| **class** Person{  //이름, 주소, 전화번호를 필드로 선언해준다.  **private** String name;  **private** String address;  **private** String number;    **public** Person(){//person이 매개변수를 받지 않을 생성자.  **this**.name=**null**;  **this**.address=**null**;  **this**.number=**null**;  }  **public** Person(String name, String address, String number){  //모든 매개변수를 받는 생성자.  **this**.name=name;  **this**.address=address;  **this**.number=number;  }  **public** **void** setName(String name){  **this**.name=name;  //이름에 대한 설정자.  }    **public** **void** setAddress(String address){  **this**.address=address;  //주소에 대한 설정자.  }    **public** **void** setNumber(String number){  **this**.number=number;  //전화번호에 대한 설정자.  }    **public** String getName(){  **return** **this**.name;  //이름에 대한 접근자.  }  **public** String getAddress(){  **return** **this**.address;  //주소에 대한 접근자.  }  **public** String getNumber(){  **return** **this**.number;  //전화번호에 대한 접근자.  }  //필드에 대한 정보를 출력하는 함수.  **public** **void** Print(){  System.***out***.println("-----------------------------------------");  System.***out***.println("이름 : "+ getName() +  "주소 : " + getAddress() +  "전화번호 : " + getNumber());  System.***out***.println("-----------------------------------------");  }  }  **class** Customer **extends** Person{//person을 상속받음.  //Customer의 필드로 고객번호와 마일리지를 선언해준다.  **private** **int** cusNum;  **private** **int** mileage;  **public** Customer(){//매개변수를 받지 않을 경우의 생성자.  **this**.cusNum=0;  **this**.mileage=0;  }  **public** Customer(String name, String address, String number, **int** cusNum, **int** mileage){  //매개변수를 모두 받을경우. 필드에 값을 대입을 해줌.  **super**(name, address, number);//부모클래스의 생성자를 불러줌.  **this**.cusNum=cusNum;  **this**.mileage=mileage;  }  //Customer에 대한 접근자와 설정자 함수 정의.  **public** **int** getcusNum(){  **return** cusNum;    }  **public** **int** getmileage(){  **return** mileage;  }  **public** **void** setCusNum(**int** cusNum){  **this**.cusNum=cusNum;    }  **public** **void** setMileage(**int** mileage){  **this**.mileage=mileage;  }  //상속클래스의 필드에 대한정보와 Customer의 필드에 대한 정보를 출력해준다.  **public** **void** Print(){  System.***out***.println("-----------------------------------------");  System.***out***.println("이름 : "+ getName() +  " 주소 : " + getAddress() +  " 전화번호 : " + getNumber());  System.***out***.println(" 고객번호 : " + cusNum +  " 마일리지 : " + mileage);  System.***out***.println("-----------------------------------------");  }  }  **public** **class** PersonTest {  **public** **static** **void** main(String[] args) {    Person p1 = **new** Person("최강석", "경기", "01039978048");//person객체생성.  Customer c1 = **new** Customer("김철수", "서울", "01011112222", 2, 1);//Customer 객체 생성.    p1.Print();//필드에 대한 정보를 출력하는 함수 호출.  c1.Print();//필드에 대한 정보를 출력하는 함수 호출.  }  } |

1. 결과

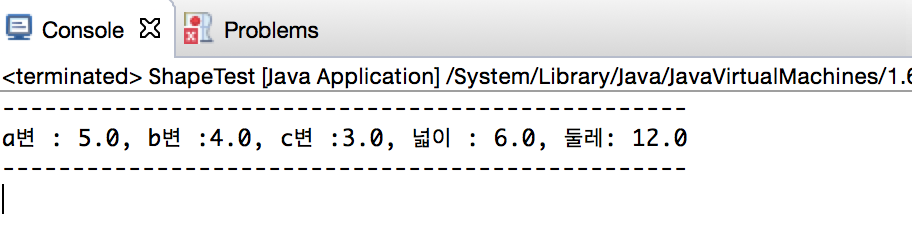


264PAGE 3번문제

1. 문제 : 본문에서 2차원 도형들을 나타내는 클래스들을 작성해본 바 있다. 이 문제에서는 보다 자세하게 구현하여 보자. 수퍼 클래스인 shape에 도형의 위치, 크기 등의 정보를 저장한다. 각각의 필드에 대하여 접근자와 설정자 메소드를 정의히자. 추가로 도형의 둘레, 면적을 계산하는 메소드도 제공한다. 삼각형을나타내는 클래스Hello을 shape에서 상속받아서 작성한다. 삼각형에 맞도록 둘레와 면적을 계산하는 메소드를 재정의한다. 테스트 클래스를 작성하여서 삼각형 객체를 생성하고 각 객체의 모든 정보를 출력한다.
2. 알고리즘 :
   1. Shpae 클래스를 만들어준다. 도형의 위치를 나타낼 x, y와 크기를 필드로 갖는다.
   2. 생성자를 만들어준다.
   3. Triangle 클래스를 받는다. Shape을 상속을 받는다.
   4. 세변을 필드로 갖는다 넓이의 접근자와 둘레에대한 접근자를 정의해준다.(세변을 이용한 넓이를 구하는 공식과 세변을 더한값)
   5. Print()는 필드값을 출력해주는 함수
   6. 메인에 객체를 생성하고 Print()를 호출시켜준다.
3. 코드 :

|  |
| --- |
| **class** Shape {  **private** **int** x, y;  //도형의 위치를 x y로 나타낸다.  //크기등의 정보를 저장하기위해 w,h를 선언해준다.  **private** **int** size;  //크기를 저장할 변수  //매개변수를 안받는 생성자.  **public** Shape() {  x = 0;  y = 0;  size = 0;  }  //매개변수를 모두받는 생성자.  **public** Shape(**int** x, **int** y, **int** size) {  **this**.x = x;  **this**.y = y;  **this**.size=size;  }      **public** **double** getPerimeter() {  **return** 0;  //부모클래스의 함수를 호출할 경우 0을 출력을해준다.  }  }  **class** Triangle **extends** Shape {  **private** **double** a, b, c;  //Shape를 상속받는 Triangle 클래스  //멤버변수로 가로 세로 높이를 받는다.  //각 세변의 길이를 의미한다.  **public** Triangle(**double** a, **double** b, **double** c) {  **super**();  **this**.a = a;  **this**.b = b;  **this**.c = c;    }  **public** **double** getArea() {  **double** s= (a+b+c)/2;  **return** Math.*sqrt*(s\*(s-a)\*(s-b)\*(s-c));  //세변을 이용한 넓이를 구하는 값을 반환을 해준다.  }  **public** **double** getPerimeter() {  **return** a + b + c ;  //세변을 더한 값(둘레)를 반환해준다.  }  **public** **void** Print() {  //도형의 정보에 대해 출력을 해준다.  System.***out***.println("-------------------------------------------------");  System.***out***.printf("a변 : " + a + ", b변 :" + b + ", c변 :" + c);  System.***out***.println(", 넓이 : " + getArea() +", 둘레: " + getPerimeter());  System.***out***.println("-------------------------------------------------");  }  }  **public** **class** ShapeTest {  **public** **static** **void** main(String[] args) {  //객체 생성  Shape s1 = **new** Shape();  Triangle t1 = **new** Triangle(5,4,3);    t1.Print();//출력 호출  }  } |

1. 결과

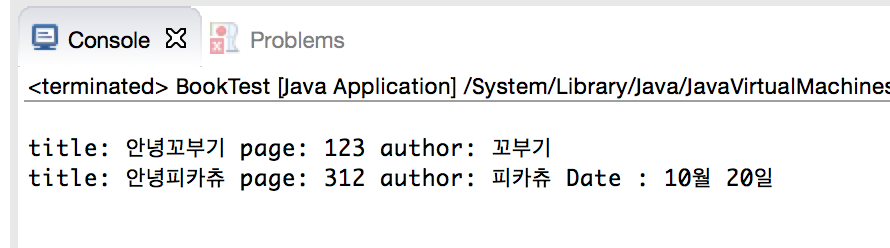


264PAGE 4번문제

1. 문제:일반적인 책을 나타내는 Book 클래스를 상속받아서잡지를 나타내는 Magazine 클래스를 작성하여 보자.Book 클래스는책 이름, 페이지 수, 저자의 정보를 가진다. Magazine 클래스는 추가로 발매일 정보를 가진다. 생성자, 접근자, 설정자를 포함하여서 각각의 클래스를 작성한다. 이들 클래스들의 객체를 만들고 각 객체의 모든 정보를 출력하는 테스트 클래스를 작성하라.
2. 알고리즘 :
   1. Book 클래스를 만들고 필드로 제목, 페이지, 저자를 선언해준다.
   2. 생성자, 접근자, 설정자, 필드를 출력하는 함수를 정의해준다.
   3. Magzine클래스를 만들어준다 (Book을 상속받는다.) Magazine은 출시일의 필드를 추가적으로 갖는다.
   4. 생성자, 접근자, 설정자, 필드를 출력하는 함수를 정의해준다.
   5. 메인에 객체를 생성해주고 Print()함수를 호출해준다.
3. 코드 :

|  |
| --- |
| **class** Book{  //필드 선언    **private** String title;//제목.  **private** **int** page;//페이지.  **private** String author;//저자.    **public** Book(){//생성자.  title="";  page=0;  author="";  }  **public** Book(String title, **int** page, String author){//매개벼수를 모두 받을 시의 생성자.  **this**.title=title;  **this**.page=page;  **this**.author=author;  }  //각 필드에대한 접근자와 설정자들을 정의해준다.    **public** **void** setTitle(String title){**this**.title=title;}  **public** **void** setPage(**int** page){**this**.page=page;}  **public** **void** setAuthor(String author){**this**.author=author;}  **public** String getTitle(){**return** **this**.title;}  **public** **int** getPage(){**return** **this**.page;}  **public** String getAuthor(){**return** **this**.author;}  **public** **void** Print(){//필드에 대한 정보출력.  System.***out***.println();  System.***out***.print("title: " + title  + " page: " + page + " author: " + author);  }  }  **class** Magazine **extends** Book{//Book 클래스를 상속받는 Magazine 클래스.  **private** String Date;//출시일 필드.    **public** Magazine(){//생성자(매개변수가 없을시)  **super**();  Date="";  }  **public** Magazine(String title, **int** page, String author, String Date){  //생성자(매개변수를 모두 받을경우)  **super**(title, page, author);//부모클래스의 생성자를 불러준다.  **this**.Date=Date;//필드에 값을 대입해준다.  }  //Date에 대한 설정자와 접근자 정의.  **public** **void** setDate(String Date){  **this**.Date=Date;  }  **public** String getDate(){  **return** Date;  }  **public** **void** Print(){//Magazine에 대한 정보출력.  **super**.Print();//부모클래스의 필드에 대한 정보를 먼저 출력.  System.***out***.println(" Date : " + Date);//Magazine 필드를 출력.  }    }  **public** **class** BookTest {  **public** **static** **void** main(String[] args) {  //객체생성과 Print()함수호출.  Book b1 = **new** Book("안녕꼬부기", 123, "꼬부기");  Magazine m1 = **new** Magazine("안녕피카츄", 312, "피카츄", "10월 20일");  b1.Print();  m1.Print();    }  } |

1. 결과

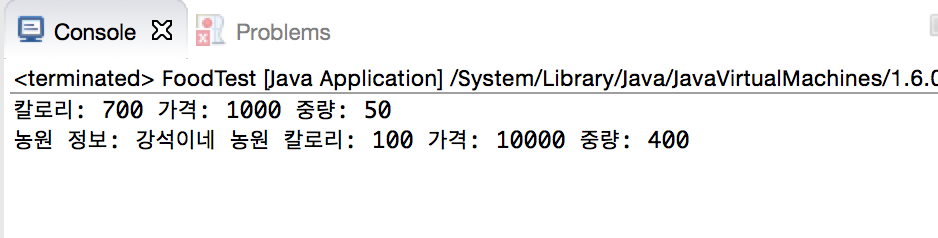


265PAGE 5번문제

1. 문제 : 일반적인 음식을 나타내는 Food클래스를 상속받아서  
   멜론을 나타내는 Melon클래스를 작성하여보자.  
   Food클래스는 칼로리,가격,중량등의 정보를 가진다.  
   Melon클래스는 추가로 경작농원 정보를 가진다.  
   먼저 UML을 그리고 생성자를 포함하여서 자바로 구현하여 보라.
2. 알고리즘 :
   1. Food클래스를 만들어준다. 필드 값으로는 칼로리, 가격, 몸무게를 갖는다.
   2. 생성자, 접근자, 설정자, 필드의 정보를 출력하는 함수를 정의해준다.
   3. Melon클래스를 만들어준다. (food를 상속받는다)
   4. Melon클래스에는 추가적인 필드로 농원 정보를 받는다.
   5. Melon에 대한 생성자, 접근자, 설정자, 필드출력 함수를 정의해준다.
   6. 메인함수에 객체들을 생성하고 Print()함수를 호출해준다.
3. 코드 :

|  |
| --- |
| **class** Food {  **private** **int** cal;//칼로리  **private** **int** price;//가격  **private** **int** weight;//몸무게  Food(){//매개변수가 없을 시의 생성자.  cal=0;  price=0;  weight=0;  }  Food(**int** cal, **int** price, **int** weight) {//매개변수를 모두 받을 시의 생성자.  **this**.cal = cal;  **this**.price = price;  **this**.weight = weight;  }  //각 필드에대한 설정자와 접근자를 정의해준다.  **public** **void** setCal(**int** cal) {**this**.cal = cal;}  **public** **void** setPrice(**int** price) {**this**.price = price;}  **public** **void** setWeight(**int** weight) {**this**.weight = weight;}  **public** **int** getCal() {**return** cal;}  **public** **int** getPrice() {**return** price;}  **public** **int** getWeight() {**return** weight;}  **public** **void** Print(){//필드의 정보를 출력하는 함수.  System.***out***.println("칼로리: " + cal + " 가격: " + price + " 중량: " + weight);  }  }  **class** Melon **extends** Food//Food를 상속받는 Melon class 생성.  {  **private** String name;  **public** Melon(){//매개변수가 없을때의 생성자.  **super**();  name="";  }  **public** Melon(String name, **int** cal, **int** price, **int** weight){  //매개변수를 모두 받을때의 생성자.  **super**(cal,price,weight);  **this**.name = name;//멜론 필드에 값을 대입.  }  //농원정보에 대한 접근자와 설정자들을 정의해준다.  **public** **void** setName(String n) {name = n;}  **public** String getName() {**return** name;}  **public** **void** Print(){//Melon의 정보를 출력해주는함수.  System.***out***.println("농원 정보: " + name + " 칼로리: " + getCal() + " 가격: " +  getPrice() + " 중량: " + getWeight());  }  }  **public** **class** FoodTest {  **public** **static** **void** main(String[] args) {  //객체 생성을해준다.  Food f1= **new** Food(700, 1000, 50);  Melon m1 = **new** Melon("강석이네 농원", 100, 10000, 400);    //객체의 필드를 출력해주는 함수를 호출해준다.  f1.Print();  m1.Print();    }  } |

1. 결과

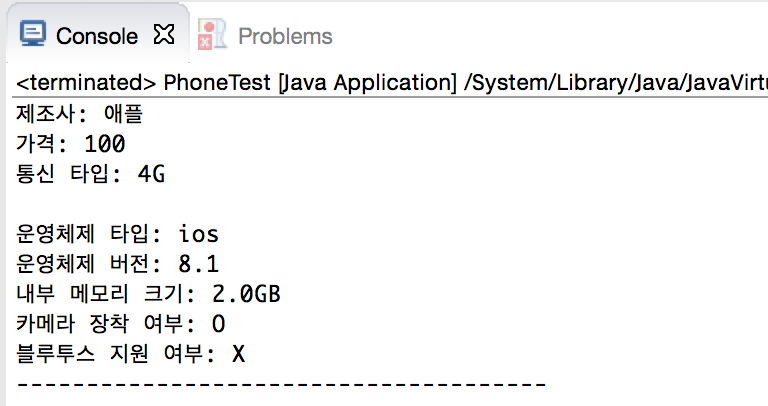


265PAGE 6번문제

1. 문제 :일반적인 휴대폰을 나타내는 Phone 클래스를 작성한다. Phone에는 제조사, 가격, 통신타입(2g 또는 3g) 등의 정보가 저장되어 있다. Phone에서 상속받아서 SmartPhone 클래스를 작성하여 보자. SmartPhone 클래스에는 운영체제 타입, 운영체제 버전, 내부 메모리 크기, 카메라 장착 여부, 블루투스 지원 여부 등의 필드가 추가된다. 생성자, 접근자, 설정자를 포함하여서 각각의 클래스를 작성한다. 이들 클래스의 객체를 만들고 각 객체의 모든 정보를 출력하는 테스트 클래스를 작성하라.
2. 알고리즘:
   1. Phone클래스를 만들어준다 필드값으로는 제조사, 가격, 통신타입을 선언해준다.
   2. 생성자, 접근자, 설정자, 필드에 대한 정보를 출력하는 함수를 만들어준다.
   3. SmartPhone클래스를 만들어준다.(Phone을 상속을 받는다.)
   4. SmartPhone은 추가적인 필드로 운영체제, 버전, 메모리, 카메라 유무, 블루투스 유무를 받는다.
   5. 생성자, 설정자, 접근자, 필드에 대한 정보를 출력하는 함수를 정의해준다.
   6. 메인 함수에 객체를 생성하고 Print()를 호출해준다.
3. 코드 :

|  |
| --- |
| **class** Phone {  //Phone에 대한 필드값.  **private** String maker;//제조사.  **private** **int** price;//가격.  **private** String type;//통신타입.  **public** Phone(String maker, **int** price, String type) {//매개변수를 모두 받을시의 생성자.  //필드에 각 값을 대입을 해준다.  **this**.maker = maker;  **this**.price = price;  **this**.type = type;  }  //각 필드에 대한 접근자와 설정자들을 정의해준다.  **public** String getMaker() {**return** maker; }  **public** **void** setMaker(String maker) {**this**.maker = maker;}  **public** **int** getPrice() {**return** price;}  **public** **void** setPrice(**int** price) {**this**.price = price;}  **public** String getType() {**return** type;}  **public** **void** setType(String type) {**this**.type = type;}  //필드에대한 정보를 출력해주는 함수.  **public** **void** Print() {  System.***out***.println("제조사: " + maker + "\n가격: " + price + "\n통신 타입: " + type);  }  }  **class** SmartPhone **extends** Phone {  //Phone을 상속받는 SmartPhone 클래스.  String os;// 운영체제.  String version;//버전.  **double** memory;//메모리.  **boolean** camera;//카메라 유무.  **boolean** bluetooth;//블루투스 유무.  **public** SmartPhone(String maker, **int** price, String type, String os, String version, **double** memory, **boolean** camera,  **boolean** bluetooth) {  //매개변수를 모두받을시의 생성자.  **super**(maker, price, type);//부모클래스의 생성자를 먼저 호출.  **this**.os = os;  **this**.version = version;  **this**.memory = memory;  **this**.camera = camera;  **this**.bluetooth = bluetooth;  //SmartPhone에 대한 필드를 대입을 해준다.  }  //필드에 대한 접근자와 설정자를 정의해준다.  **public** String getOs() {**return** os;}  **public** **void** setOs(String os) {**this**.os = os;}  **public** String getVersion() {**return** version;}  **public** **void** setVersion(String version) {**this**.version = version;}  **public** **double** getMemory() { **return** memory;}  **public** **void** setMemory(**double** memory) {**this**.memory = memory;}  **public** **boolean** isCamera() {**return** camera;}  **public** **void** setCamera(**boolean** camera) {**this**.camera = camera;}  **public** **boolean** isBluetooth() {**return** bluetooth;}  **public** **void** setBluetooth(**boolean** bluetooth) {**this**.bluetooth = bluetooth;}  **public** **void** Print() {  //필드의 정보를 출력해준다.  **super**.Print();//부모클래스의 필드를 먼저 출력해준다.  System.***out***.println("\n운영체제 타입: " + os + "\n운영체제 버전: " + version + "\n내부 메모리 크기: " + memory  + "GB\n카메라 장착 여부: " + (camera ? "O" : "X") + "\n블루투스 지원 여부: " + (bluetooth ? "O" : "X"));  System.***out***.println("--------------------------------------");  //SmartPhone의 필드를 출력.  //boolean 타입들은 삼항연산자를 이용해서 O, X로 출력을 해준다.  }  }  **public** **class** PhoneTest {  **public** **static** **void** main(String[] args) {  //객체 생성과 Print()함수 호출  SmartPhone sp1 = **new** SmartPhone("애플", 100, "4G", "ios", "8.1", 2.00, **true**, **false**);  sp1.Print();  }  } |

1. 결과



265PAGE 7번문제

1. 문제 : 다음 그림에 해당하는 클래스를 작성하여 보자. 모든 학생은 이름, 학번, 소속 학과, 학년, 이수 학점 수를 가진다.추가적으로 학부생은 소속 동아리명을 가지고 있고 대학원생은 조교 유형과 장학금 비율을 가진다.조교 유형에는 교육 조교와 연구 조교가있으며 장학금 비율은 0과 1사이의 값이다.각 클래스는 적절한 생성자 메소드, 접근자 메소드, 변경자 메소드를 가진다.이러한 클래스들의 객체를 만들고 각 객체의 모든 정보를 출력하는 테스트 클래스를 작성하라.
2. 알고리즘 :
   1. Student클래스를 만들어준다 필드로는 이름, 학번, 학과, 학년, 이수학점이 들어간다.
   2. 생성자, 접근자, 설정자, 필드를 출력하는 함수를 만들어준다.
   3. UnderGraduate클래스를 만들어준다.(Student를 상속을 는다.)추가적인 필드로는 동아리가 들어간다.
   4. UnderGraduate의 생성자, 접근자, 설정자, 정보를 출력하는 함수를 만들어준다.
   5. Graduate클래스를 만들어준다.(Student를 상속 받는다.) 추가적인 필드로는 조교유형, 장학금 비율이 들어가게된다.
   6. Graduate의 생성자, 접근자 설정자, 정보를 출력하는 함수를 만들어준다.
   7. 메인 함수에 객체를 생성하고 print()를 호출해준다.
3. 코드 :

|  |
| --- |
| **class** Student {  //학생클래스에는 필드로 이름, 학번, 학과, 학년, 이수학점이 들어간다.  **protected** String name;  **private** String number;  **private** String part;  **private** **int** grade;  **private** **int** passCredit;    **public** Student(){//매개변수가 없을 때의 생성자.  **this**("","","",0,0);  }  **public** Student(String name, String number, String part, **int** grade, **int** passCredit ){  //매개변수를 전부 받을때 생성자  //각 값들을 대입해준다.  **this**.name = name;  **this**.number = number;  **this**.part = part;  **this**.grade = grade;  **this**.passCredit=passCredit;  }  //각 필드에대한 설정자들을 정의해준다.  **public** **void** setName(String name){ **this**.name=name;}  **public** **void** setNumber(String number){ **this**.number=number;}  **public** **void** setPart(String part){ **this**.part=part;}  **public** **void** setGrade(**int** grade){**this**.grade=grade;}  **public** **void** setpassCredit(**int** passCredit){**this**.passCredit=passCredit;}    //각 필드에 대한 접근자들을 정의해준다.  **public** String getName(){ **return** name;}  **public** String getNumber(){ **return** number;}  **public** String getPart(){ **return** part;}  **public** **int** getGrade(){**return** grade;}  **public** **int** getpassCredit(){**return** passCredit;}  //학생 클래스의 필드의 정보를 출력해준다.  **public** **void** Print(){  System.***out***.printf( "이름 : %s 학번 : %s 학과 : %s 학년 : %d 이수학점 : %d ", name, number, part, grade, passCredit);  }  }  **class** UnderGraduate **extends** Student{  //재학생은 동아리(club)을 필드값으로 가진다.  **private** String club;    //재학생에 대한 생성자.(매개변수가 없을 시)  **public** UnderGraduate(){  **super**();  club="";  }  //재학생에 대한 생성자(매개변수를 전부 받을 )  **public** UnderGraduate(String name, String number, String part, **int** grade, **int** passCredit,  String club){  **super**(name, number, part, grade, passCredit);//부모클래스의 생성자를 불러준다.  **this**.club = club;//재학생클래스의 필드를 대입해준다.  }  //동아리에 대한 설정자와 접근자를 정의해준다.  **public** **void** setClub(String club){ **this**.club = club;}  **public** String getClub(){**return** club;}  **public** **void** Print(){  **super**.Print();//부모클래스의 출력을 먼저부른다.  System.***out***.printf(" 동아리 : %s", club);//나머지 재학생 필드에 대한 정보를 출력해준다.  }        }  **class** Graduate **extends** Student{//대학원생에 대한 생성자.  **private** String type;//조교유형.  **private** **double** scholarshipRate;//장학금 비율.    **public** Graduate(){//생성자(매개변수가 없을 시)  **super**();  type="";  scholarshipRate=0.0;  }  //생성자(매개벼수를 전부 받을 시)  **public** Graduate(String name, String number, String part, **int** grade, **int** passCredit,  String type, **double** scholarshipRate){    **super**(name, number, part, grade, passCredit);//부모클래스의 생성자를 불러준다.  //나머지 필드에 대한 대입을 해준다.  **this**.type = type;  **this**.scholarshipRate=scholarshipRate;    }  //필드에 대한 접근자와 설정자를 정의해준다.  **public** **void** setType(String type){ **this**.type = type;}  **public** **void** setScholarchip(**double** ScholarshipRate){**this**.scholarshipRate=ScholarshipRate;}  **public** String getType(){**return** type;}  **public** **double** getScholarshipRate(){**return** scholarshipRate;}    **public** **void** Print(){  **super**.Print();//부모클래스의 필드를 먼저출력.  System.***out***.printf(" 조교유형 : %s 장학금 비율 : %.1f", type, scholarshipRate);  //대학원생에대한 필드 출력.  }  }  **public** **class** StudentTest {  **public** **static** **void** main(String[] args) {  //객체를 만들어준다.  Graduate g1 = **new** Graduate("꼬북이", "2007123456", "전자공학과", 1, 150, "교육조교", 0.5 );  UnderGraduate u1 = **new** UnderGraduate("최강석", "2012136132", "컴퓨터공학과", 2, 60, "피버");    g1.Print();//출력호출  System.***out***.println();  u1.Print();//출력호출  }  } |

1. 결과

