|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **[유헬스2\_7주\_실습과제]** | **학번:** | **20175105** | **이름:** | **곽영주** |

* 중간 점검

1. 다음 프로그램 시 오류가 발생한다. 오류메시지를 제시하고, 오류가 발생하지 않도록 수정한 후 결과를 제시하시오

class A {

final public int myMethod(int a, int b) { return 0; }

}

class B extends A {

public int myMethod(int a, int b) {return 1; }

}

public class Test{

public static void main(String args[]) {

A b = new B();

System.out.println("x = " + b.myMethod(0, 1));

}

}

|  |
| --- |
| **[오류메시지]** |
| **[수정 후 결과]** |

1. 다음과 같은 클래스 정의에서 질문에 답하라.

**class** ClassA {

**public** **void** methodOne(**int** i) { }

**public** **void** methodTwo(**int** i) { }

**public** **static** **void** methodThree(**int** i) { }

**public** **static** **void** methodFour(**int** i) { }

}

**class** ClassB **extends** ClassA {

**public** **static** **void** methodOne(**int** i) { }

**public** **void** methodTwo(**int** i) { }

**public** **void** methodThree(**int** i) { }

**public** **static** **void** methodFour(**int** i) { }

}

(1) 어떤 메소드가 수퍼 클래스의 메소드를 재정의하고 있는가? (힌트) 정적 메소드로 인스턴스 메소드를 재정의할 수 없다.

(2) 어떤 메소드가 수퍼 클래스의 메소드를 가리고 있는가? (힌트) 서브 클래스에서 똑같은 정적 메소드를 정의하는 경우, 가린다고(hide) 한다. 이 경우에, 서브 클래스 객체에서 호출되면 서브 클래스의 정적 메소드가 호출되고 만약 수퍼 클래스 객체에서 호출되면 수퍼 클래스의 정적 메소드가 호출된다.

(3) 컴파일 오류를 지적하라. 실제로 컴파일 해 보아도 좋다.

|  |
| --- |
| **[1번 해답]**  **public** **void** methodTwo(**int** i) { }  **public** **static** **void** methodFour(**int** i) { } |
| **[2번 해답]**  **public** **static** **void** methodFour(**int** i) { } |
| **[3번 해답]**   * methodOne 메소드는 수퍼클래스에서 정적메소드가 아니기 때문에 서브클래스에서 재정의할 때 정적메소드로 하면 안된다. * methodThree 메소는 수퍼클래스에서 정적메소드이기 때문에 서브클래스에서 재정의할 때 정적메소드로 재정의해야 한다.   **public** **class** ClassA {  **public** **void** methodOne(**int** i) { }  **public** **void** methodTwo(**int** i) { }  **public** **static** **void** methodThree(**int** i) { }  **public** **static** **void** methodFour(**int** i) { }  }  **public** **class** ClassB **extends** ClassA {  **public** **void** methodOne(**int** i) { }  **public** **void** methodTwo(**int** i) { }  **public** **static** **void** methodThree(**int** i) { }  **public** **static** **void** methodFour(**int** i) { }  } |

1. 제시된 프로그램에 존재하는 오류를 제시하고 수정하시오.

**public** **class** Bike{

**private** **int** gear;

**public** **int** speed;

}

**public** **class** MountainBike expands Bike{

**public** **int** seatHeight;

**public** **void** MountainBike(**int** g){

**super**();

gear=g;

}

}

|  |
| --- |
| **[해답]**  expands -> extends  **public** **void** MountainBike(**int** g) // 생성자이기 때문에 void 삭제  수퍼클래스에 gear 초기화하는 생성자 생성  **super**(); gear=g; -> **super**(g);  ---------------------------------------------------------  **public** **class** Bike {  **private** **int** gear;  **public** **int** speed;    **public** Bike(**int** gear) {  **this**.gear = gear;  }  }  **public** **class** MountainBike **extends** Bike{  **public** **int** seatHeight;  **public** MountainBike(**int** g){  **super**(g);  }  } |

1. 다음 프로그램의 출력을 쓰시오.

**class** Animal{

String name;

**public** Animal() {

name = "UNKNOWN";

System.*out*.println("동물입니다:" + name);

}

**public** Animal(String name) {

**this**.name = name;

System.*out*.println("동물입니다:" + name);

}

}

**class** Lion **extends** Animal{

**public** Lion() { System.*out*.println("사자입니다."); }

**public** Lion(String name) {

**super**(name);

System.*out*.println("사자입니다.");

}

}

**public** **class** Test{

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

Lion lion=**new** Lion("Brave");

Lion lion2=**new** Lion();

}

}

|  |
| --- |
| **[해답]**  동물입니다:Brave  사자입니다.  동물입니다:UNKNOWN  사자입니다. |

1. 다음 프로그램의 출력을 쓰시오.

class A {

public void f(A a) {

System.out.print("fa(A) ");

}

public void f(B b) {

System.out.print("fa(B) ");

}

}

class B extends A {

public void f(A a) {

System.out.print("fb(A) ");

}

public void f(B b) {

System.out.print("fb(B) ");

}

}

public class Test {

public static void main(String[] args) {

A a = new A();

B b = new B();

A ba = (A) b;

a.f(a);

a.f(b);

b.f(a);

b.f(b);

a.f(ba);

b.f(ba);

ba.(a);

ba.f(b);

ba.f(ba);

}

}

|  |
| --- |
| **[해답]**  fa(A) fa(B) fb(A) fb(B) fa(A) fb(A) fb(A) fb(B) fb(A) |

* 프로그램 과제

1. 다음을 만족하는 클래스 Person을 작성하시오. 단, Person 클래스에 대한 객체 생성은 불가

* 필드 :
* 이름(name, String), 나이(age, int), 자식클래스에서만 접근
* 메소드
* 생성자 : 매개변수로 받은 값을 이름과 나이 필드로 초기화
* disPlay() : 이름과 나이 출력, 형식매개변수, 반환값 없음

1. Customer 클래스 : Person 클래스 상속

* 필드:
* 고객구분(type, String), 포인트(point, int), private
* 메소드 :
* 생성자 : 이름과 나이는 매개변수로 받은 값으로 초기화 하고(부모클래스 생성자 사용), 고객구분과 포인트는 매개변수로 한 개의 문자를 받아 다음과 같이 처리

|  |  |  |
| --- | --- | --- |
| 문자 | 고객구분 | 포인트 |
| g 또는 G | 일반 | 500 |
| v 또는 V | 우수 | 1000 |
| s 또는 S | 최우수 | 1500 |

* 포인트 필드에 대한 접근자
* disPlay() : 부모 클래스 disPlay() 재정의

이름과 나이(부모클래스 메소드 사용) 고객구분, 포인트 출력

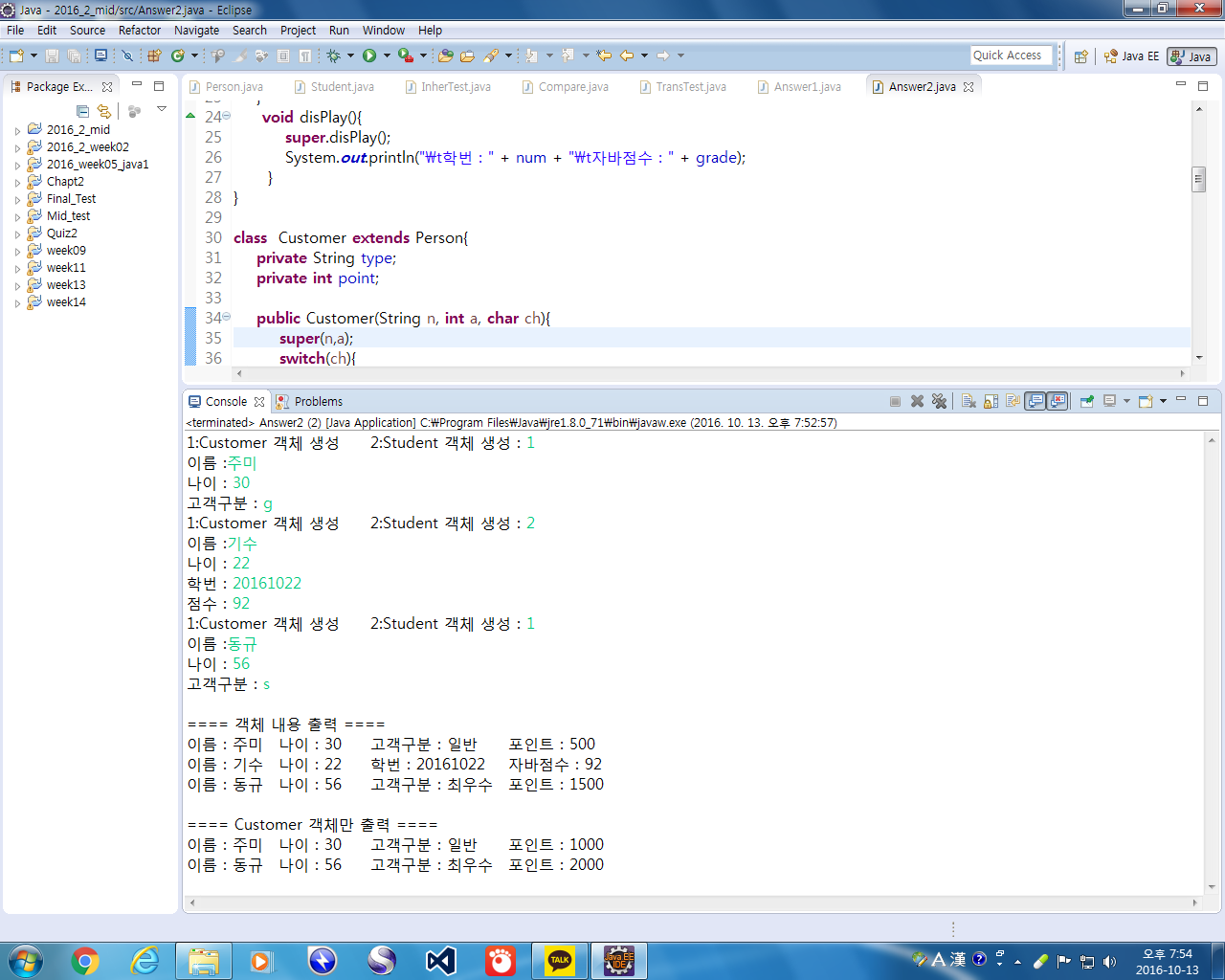
* addPoint(int) : 반환값 없고, 매개변수 값만큼 포인트를 증가한다.

1. Student 클래스 : Person 클래스 상속

* 필드:
* 학번(num, String), 자바성적(grade, int), private
* 메소드 :
* 생성자 : 이름, 나이(부모클래스 생성자 사용), 학번, 자바성적을 매개변수로 받은 값으로 초기화
* disPlay() : 부모 클래스 disPlay() 재정의

이름과 나이(부모클래스 메소드 사용) 학번, 자바성적 출력

1. 위의 문제에서 정의한 부모클래스와 자식클래스를 사용하여 제시된 결과처럼 출력되는 프로그램을 완성하시오. 주석에 해당하는 문장을 작성하면 됨



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

**public** **class** Answer2 {

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

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

//부모 클래스 타입으로 객체 배열을 선언하고 생성한다. 배열크기는 5

//1번을 입력하면 Customer 객체를 2번을 입력하면 Student객체를 //생성하여 배열에 저장한다. 단, 필요한 값은 입력을 받으며

//이름과 나이, 학번, 점수를 입력할 때는 제시된 메소드를 사용한다.

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

//배열에 저장된 객체내용을 출력하면서 Customer 객체의

포인트를 500씩 //증가, 객체 내용 출력 시 foreach 구문을 사용할 것

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

//배열에 저장 된 객체 중 Customer 객체만 출력

**static** String input(String str, Scanner key){

System.***out***.print(str);

**return** key.next();

}

**static** **int** digit(String str, Scanner key){

System.***out***.print(str);

**return** key.nextInt();

}

}

|  |
| --- |
| **[프로그램 소스]**  **public** **class** Person {  **protected** String name;  **protected** **int** age;    **public** Person(){}    **public** Person(String name, **int** age) {  **this**.name = name;  **this**.age = age;  }    **public** **void** disPlay() {  System.***out***.print("이름 : " + name + "\t나이 : " + age);  }  }  **public** **class** Customer **extends** Person {  **private** String type;  **private** **int** point;    **public** Customer(**char** ch, String name, **int** age) {  **super**(name,age);    **switch**(ch) {    **case** 'g' :  **case** 'G' :  type = "일반";  point = 500;  **break**;    **case** 'v' :  **case** 'V' :  type = "우수";  point = 1000;  **break**;    **case** 's' :  **case** 'S' :  type = "최우수";  point = 1500;  **break**;  }  }  **public** **int** getPoint() {  **return** point;  }      @Override  **public** **void** disPlay() {  **super**.disPlay();  System.***out***.print("\t고객구분 : " + type + "\t포인트 : " + point + "\n");  }    **public** **void** addPoint(**int** add) {  point += add;  }  }  **public** **class** Student **extends** Person {  **private** String num;  **private** **int** grade;    **public** Student(String name, **int** age, String num, **int** grade) {  **super**(name,age);  **this**.num = num;  **this**.grade = grade;  }    @Override  **public** **void** disPlay() {  **super**.disPlay();  System.***out***.print("\t학번 : " + num + "\t자바점수 : " + grade + "\n");  }  }  **import** java.util.\*;  **public** **class** Answer2 {  **public** **static** **void** main(String[] args) {  Scanner key = **new** Scanner(System.***in***);  Person[] per = **new** Person[5];    **for**(**int** i=0; i<per.length; i++) {  System.***out***.print("1:Customer 객체 생성\t2:Student 객체 생성 : ");  **int** menu = key.nextInt();    **if**(menu == 1) {  System.***out***.print("고객구분 : ");  per[i] = **new** Customer(key.next().charAt(0), *input*("이름 : ",key), *digit*("나이 : ",key));  }  **else** **if**(menu == 2) {  per[i] = **new** Student(*input*("이름 : ",key), *digit*("나이 : ",key), *input*("학번 : ",key), *digit*("점수 : ",key));  }  }    System.***out***.println("\n==== 객체 내용 출력 ====");  **for**(Person data : per) {  data.disPlay();  **if**(data **instanceof** Customer){  ((Customer)data).addPoint(500);  }  }    System.***out***.println("\n==== Customer 객체만 출력 ====");  **for**(Person data : per) {  **if**(data **instanceof** Customer){  ((Customer)data).disPlay();  }  }  }    **public** **static** String input(String str, Scanner key) {  System.***out***.print(str);  **return** key.next();  }    **public** **static** **int** digit(String str, Scanner key) {  System.***out***.print(str);  **return** key.nextInt();  }  } |
| **[실행 결과]** |