|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **[유헬스2\_9주\_실습과제]** | **학번:** |  | **이름:** |  |

* 프로그램 과제

1. 사각형과 삼각형, 사다리꼴 도형 중 사용자가 선택한 도형을 출력하는 프로그램을 작성하시오. 단 도형을 그리는데 사용하는 문자는 도형별로 다르게 사용한다.

* ShapeDraw 추상 클래스

- 추상 메소드 : draw(int x, char ch) : 도형을 그리는 행 크기와 출력할 문자를 매개변수 x, ch로 받는다, 반환 값은 없다

* Rectangle 클래스 : ShapeDraw 클래스 상속

- draw() : 메소드 오버라이딩, 사각형 출력

* Triangle 클래스 : ShapeDraw 클래스 상속

- draw() : 메소드 오버라이딩, 삼각형 출력

* Ladder 클래스 : ShapeDraw 클래스 상속

- draw() : 메소드 오버라이딩, 사다리꼴 출력

* TestDraw 클래스

- main() 메소드

메뉴항목에서 출력하고자 하는 도형과, 행크기, 문자를 입력 받는다

다형성을 적용할 것.

[출력 예시]

1. Rectangle 2. Triangle 3. Ladder 4. Finish

메뉴를 선택하세요: 2

행 크기를 입력하세요 : 5

문자를 입력하세요 : \*

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

**public** **abstract** **class** ShapeDraw {

**public** **abstract** **void** draw(**int** x, **char** ch);

}

**public** **class** Rectangle **extends** ShapeDraw {

**public** **void** draw(**int** x, **char** ch) {

**for** (**int** i = 0; i < x; i++) {

**for** (**int** j = 0; j < x; j++) {

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

}

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

}

}

}

**public** **class** Triangle **extends** ShapeDraw {

**public** **void** draw(**int** x, **char** ch) {

**for** (**int** i = 0; i < x; i++) {

**for** (**int** j = 0; j <= i; j++) {

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

}

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

}

}

}

**public** **class** Ladder **extends** ShapeDraw {

**public** **void** draw(**int** x, **char** ch) {

**for** (**int** i = 0; i < x; i++) {

**for** (**int** k = x; k != i; k--) {

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

}

**for** (**int** j = 0; j < x; j++) {

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

}

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

}

}

}

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

**public** **class** TestDraw {

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

// **TODO** Auto-generated method stub

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

ShapeDraw SD;

**while** (**true**) {

System.***out***.println("1.Rectangle 2.Triangle 3.Ladder 4.Finish");

System.***out***.print("메뉴를 입력하세요 ");

**int** n = scan.nextInt();

**if** (n== 4) {

System.***out***.println("종료");

**break**;

}

System.***out***.print("행 크기를 입력하세요 ");

**int** x = scan.nextInt();

System.***out***.print("문자를 입력하세요");

**char** ch = scan.next().charAt(0);

**switch** (n) {

**case** 1:

SD = **new** Rectangle();

SD.draw(x, ch);

**break**;

**case** 2:

SD = **new** Triangle();

SD.draw(x, ch);

**break**;

**case** 3:

SD = **new** Ladder();

SD.draw(x, ch);

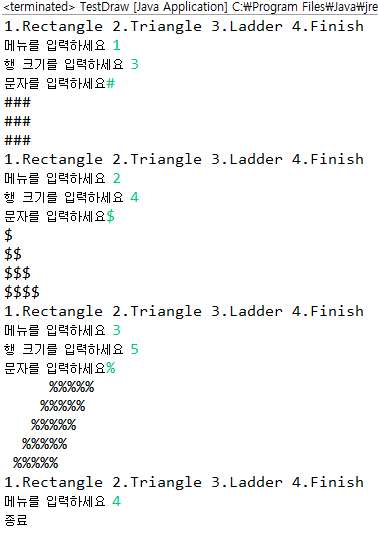
**break**;

}

}

}

}



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

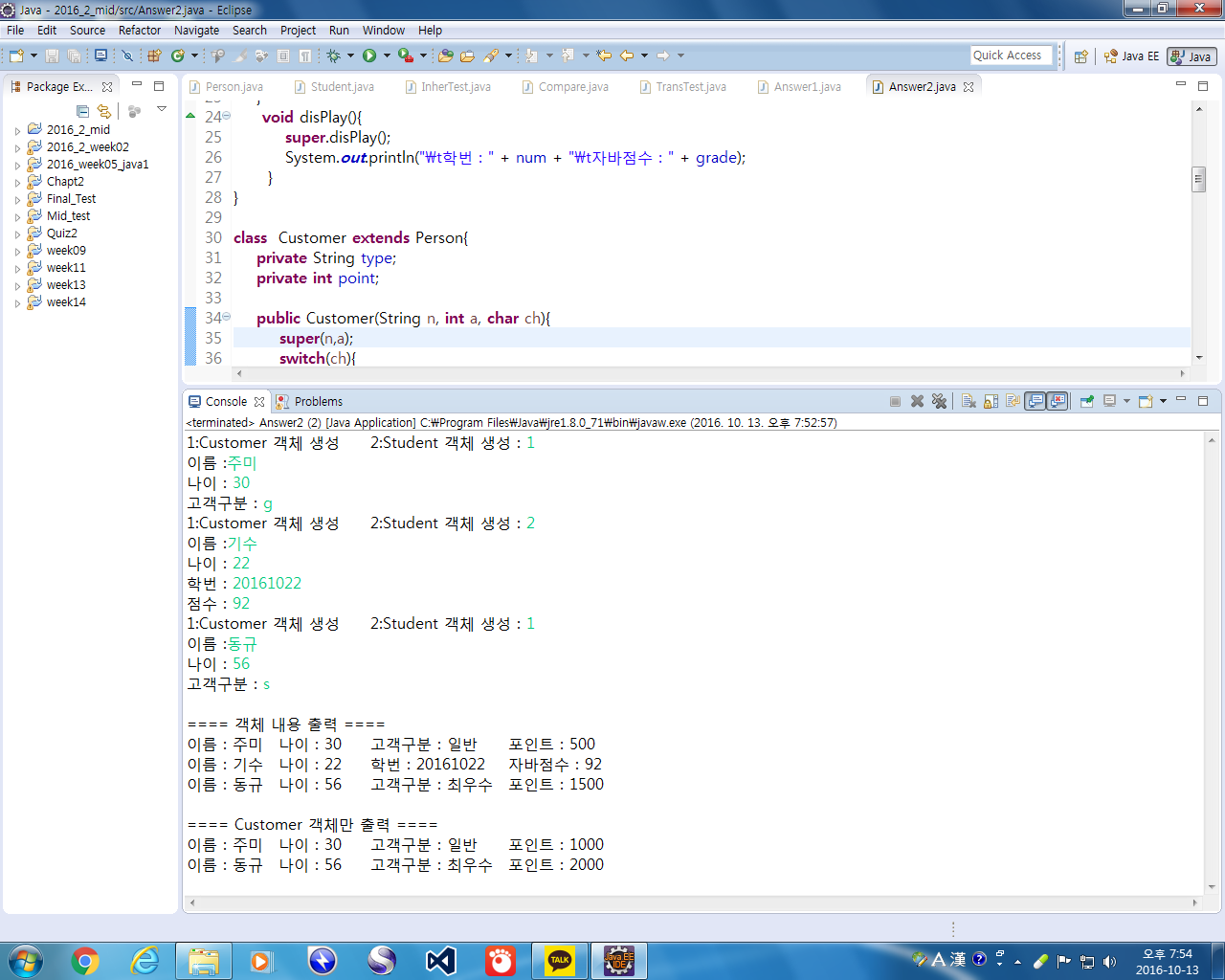
* addPoint(int) : 반환값 없고, 매개변수 값만큼 포인트를 증가한다.

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();

}

}

프로그램소스 -------------------------------------------------

**public** **class** Person {

**protected** String name;

**protected** **int** age;

Person(String name, **int** age) {

**this**.name = name;

**this**.age = age;

}

Person() {

}

**public** **void** disPlay() {

System.***out***.print("이름 : " + name);

System.***out***.print("\t나이 : " + age);

}

}

**public** **class** Customer **extends** Person {

**private** String type;

**private** **int** point;

Customer(String name, **int** age, **char** ch) {

**super**(name, age);

**if** (ch == 'G' || ch == 'g') {

type = "일반";

point = 500;

} **else** **if** (ch == 'v' || ch == 'V') {

type = "우수";

point = 1000;

} **else** **if** (ch == 's' || ch == 'S') {

type = "최우수";

point = 1500;

}

}

Customer(){}

**public** **int** getPoint() {

**return** point;

}

**public** **void** disPlay(){

**super**.disPlay();

System.***out***.print("\t고객구분 : "+type);

System.***out***.print("\t포인트 : "+point);

}

**public** **void** addPoint(**int** a){

point+=a;

}

}

**public** **class** Student **extends** Person {

**private** String num;

**private** **int** grade;

Student(String name, **int** age, String num, **int** grade) {

**super**(name, age);

**this**.num = num;

**this**.grade = grade;

}

Student() {

}

**public** **void** disPlay() {

**super**.disPlay();

System.***out***.print("\t학번 : " + num);

System.***out***.print("\t자바점수 : " + grade);

}

}

**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

Person[] P = **new** Person[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.

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

System.***out***.print("1.Customer 객체 생성\t2.Student 객체생성");

**switch** (key.nextInt()) {

**case** 1:

P[i] = **new** Customer(*input*("이름 : ", key), *digit*("나이: ", key),

*input*("고객구분 : ", key).charAt(0));

**break**;

**case** 2:

P[i] = **new** Student(*input*("이름 : ", key), *digit*("나이 : ", key),

*input*("학번 : ", key), *digit*("점수 : ", key));

**break**;

}

}

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

// 배열에 저장된 객체내용을 출력하면서 Customer 객체의 포인트를 500씩 //증가, 객체 내용 출력 시 foreach

// 구문을 사용할 것

**for** (Person temp : P) {

temp.disPlay();

**if** (temp **instanceof** Customer)

((Customer) temp).addPoint(500);

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

}

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

// 배열에 저장 된 객체 중 Customer 객체만 출력

**for** (Person temp : P) {

**if** (temp **instanceof** Customer) {

temp.disPlay();

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

}

}

}

**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();

}

}

