

«باسمه تعالی»

(پاسخ سوالات دستورکار جلسه پنجم)

در پایین یک متد main داده شده است. از شما می‌خواهیم با توجه به مفاهیم وراثت و چندریختی (بخش دوم پیاده‌سازی) هر خط را توجیه و در صورت وجود خطا آن را تصحیح نمایید (این کار را بدون پیاده‌سازی انجام دهید).

```
public static void main (String... args) {  
    Circle circle1 = new Shape(19);  
    Shape circle2 = new Circle(3);  
    Rectangle rect1 = new Triangle(1,4,1);  
    Polygon rect2 = new Rectangle(8,5,8,5);  
    Rectangle rec3 = new Shape(6,6,6,6);  
    Polygon tri1 = new Triangle(2,2,2);  
    Triangle tri2 = new Triangle(4,4,6);  
    Shape tri3 = new Triangle(2,2,2);  
  
    circle1 = circle2;  
    rect2 = rect3;  
    tri1 = tri3;  
    cricle2 = tri3;  
    tri3 = tri2;  
  
    rect3 = new Shape(2,3,2);  
    System.out.println(rect3.toString());  
}
```

### خط 1) `public static void main (String... args)`

همانطور که در دستور کار توضیح داده شد میتوان ورودی تابع را به صورت `varargs` داد . البته نوع تعریف دیگر ورودی به صورت `String [] args` نیز میباشد که برای تابع `main` متداول تر است.

### خط 2) `Circle circle1 = new Shape(19);`

نادرست میباشد. زیرا استاتیک تایپ `circle1` از جنس `Circle` میباشد ولی داینامیک تایپ آن از جنس `Shape` نادرست میباشد و در عین حال `Shape` یک `Superclass` برای `Circle` میباشد و بنابراین نمیتوان یک شیء از جنس `supertype` را داخل `subtype` قرار داد. برای مثال شکل درست آن :

```
Circle circle1 = new Circle(19);
```

```
Shape circle1 = new Circle(19);
```

### خط 3) `Shape circle2 = new Circle(3);`

درست میباشد. زیرا در اینجا `polymorphism` برقرار است و میتوان یک `subtype` را داخل یک `supertype` قرار داد.

### خط 4) `Rectangle rect1 = new Triangle(1,4,1);`

نادرست میباشد. زیرا `Triangle` و `Rectangle` دو شیء متمایز اند و روابط `subtype` و `supertype` بین آن ها نیست و در سلسله وراثت هر دو کلاس در یک سطح اند. برای مثال شکل های درست آن میتواند:

```
Shape tri1 = new Triangle(1,4,1);
```

```
Polygon tri1 = new Triangle(1,4,1);
```

```
Rectangle rect1 = new Rectangle(1,4,1,4);
```

### خط 5) `Polygon rect2 = new Rectangle(8,5,8,5);`

درست میباشد. زیرا در اینجا `polymorphism` برقرار است و میتوان یک `subtype` را داخل یک `supertype` قرار داد. و همانطور هم که مشخص است `Rectangle` یک `subclass` برای `Polygon` میباشد.

**خط 6) Rectangle rec3 = new Shape(6,6,6,6);**

نادرست می باشد. زیرا استاتیک تایپ rec3 از جنس Rectangle می باشد ولی داینامیک تایپ آن از جنس کلاس Shape می باشد و در عین حال Shape یک Superclass برای Rectangle می باشد و بنابراین نمیتوان یک شیء از جنس supertype را داخل subtype قرار داد. برای مثال شکل درست آن :

**Rectangle rec3 = new Rectangle(6,6,6,6);**

**Shape rec3 = new Rectangle(6,6,6,6);**

**خط 7) Polygon tri1 = new Triangle(2,2,2);**

درست می باشد. زیرا در اینجا polymorphism برقرار است و میتوان یک subtype را داخل یک متغیر از جنس supertype قرار داد. و همانطور هم که مشخص است Triangle یک subclass برای Polygon می باشد.

**خط 8) Triangle tri2 = new Triangle(4,4,6);**

درست می باشد. در این جا استاتیک تایپ و داینامیک تایپ tri2 از جنس Triangle می باشد و همه نکات رعایت شده است.

**خط 9) Shape tri3 = new Triangle(2,2,2);**

درست می باشد. زیرا در اینجا polymorphism برقرار است و میتوان یک subtype را داخل یک متغیر از جنس supertype قرار داد. و همانطور هم که مشخص است Triangle یک subclass برای Shape می باشد.

**خط 10) circle1 = circle2;**

نادرست می باشد. زیرا استاتیک تایپ circle2 از جنس Shape می باشد و استاتیک تایپ circle1 از جنس Circle می باشد و برای اینکه این عبارت درست باشد باید circle2 به Circle ، cast شود. درست آن به صورت زیر است:

**circle1 = (Circle)circle2;**

**خط 11) rect2 = rect3;**

درست میباشد. چون rect2 از جنس Polygon است و rect3 هم از نوع Rectangle میباشد و چون Rectangle یک subclass میباشد پس رد اینجا polymorphism رعایت شده و میتوان subtype را داخل supertype قرار داد. (با این فرض که غلط املائی موجود در rect3 را در نظر نگیریم)

**خط 12) tri1 = tri3;**

نادرست میباشد. زیرا استاتیک تایپ tri3 از جنس Shape میباشد و استاتیک تایپ tri1 از جنس Polygon میباشد و برای اینکه این عبارت درست باشد باید tri3 به Polygon ، cast شود. درست آن به صورت زیر است:

`tri1 = (Polygon)tri3;`

**خط 13) cricle2 = tri3;**

اگر غلط نوشتاری را در نظر نگیریم درست میباشد. زیرا استاتیک تایپ circle2 و tri3 هر دو shape میباشد.

**خط 14) tri3 = tri2;**

درست میباشد. زیرا یک subtype را داخل supertype قرار میدهیم و polymorphism رعایت شده است.

**خط 15) rect3 = new Shape(2,3,2);**

نادرست میباشد. اولاً که با دادن 3 ورودی به shape نمیتوان چیزی ساخت زیرا shape برای ما یک abstract class میباشد. ثانیاً از طرفی یک supertype را نمیتوان در subtype قرار داد.

**خط 16) System.out.println(rect3.toString());**

اگر غلط املائی موجود در rect3 را در نظر نگیریم درست میباشد.