

((به نام خدا))

محمد جواد زندیه 9831032

دستور کار شماره 5

سوال: تفاوت varargs با array چیست؟

در واقع تفاوت چندانی میان این دو وجود ندارد. فقط در ظاهر تفاوت دارند. یعنی اگر پارامتر ورودی به متد یک آرایه باشد باید ابتدا یک فضا برای آرایه در نظر بگیریم بعد با مقدار دهی کردن آرایه کل آرایه را به متد پاس کنیم اما در varargs نیاز به این حافظه گیری نیست و مقادیر را به طور مستقیم در ورودی قرار میدهیم. نحوه استفاده هم به این صورت است:

```
method(new String[]{"first", "second"});
```

```
is called like method("first", "second");
```

سوال: به نظر شما چگونه یک شی Integer را توانستیم به 2 تقسیم کنیم؟

در جاوا قابلیت auto boxing و un boxing وجود دارد. یعنی هر جایی که نیاز به یک شی داشتیم به صورت خودکار فرایند auto boxing رخ میدهد. یعنی یک شی ساخته میشود که مثلاً دارای فیلد int است و این فیلد با مقدار داده شده پر میشود. (مثلاً در array list) و در هر جا هم که نیاز به primitive ها داشته باشیم به صورت خودکار مثلاً شی Integer مقدار فیلد خود را در یک primitive یعنی int ریخته و از آن استفاده میکند. (مثل همین مثال که شی را به 2 تقسیم کردیم در واقع یک int درست شد و مقدار فیلد Integer در آن ریخته شد).

سوال: در باز طراحی ، paint فقط یک ArrayList از shape نگه میدارد. پس چگونه میتوان از متدهای مخصوص یک شکل خاص مانند دایره استفاده کرد؟

با استفاده از مفاهیم چند ریختی این امکان ایجاد میشود که یک تعداد متدهای مجازی (abstract) ایجاد کنیم و فرایند پیاده سازی آن را به subclass ها بسپاریم.

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

```

1. Circle circle1 = new Shape(19);

این عبارت به این علت ایراد دارد که کلاس Shape به صورت abstract تعریف شده است که در نتیجه نمیتوان از آن شی ایجاد کرد. همچنین با این کار در حال downCasting هستیم اما عبارت cast را پشت new Shape(19) نگذاشته ایم که منجر به compile error میشود.

Shape circle = new Circle(19); صورت درست:

2. Shape circle2 = new Circle(3);

این عبارت درست است در واقع در حال ایجاد شی هستیم که ارجاع آن از نوع super class است و شی ما از نوع sub class.

3. Rectangle rect1 = new Triangle(1,4,1);

بین Rectangle و Triangle هیچ رابطه وراثتی برقرار نیست که بتوان به این گونه شی ایجاد کرد.

4. Polygon rect2 = new Rectangle(8,5,8,5);

این نوع ایجاد شی هم درست است. به همان علت شماره 2

5. Rectangle rec3 = new Shape(6,6,6,6);

این خط غلط است چون Rectangle is a Shape یعنی رابطه ارث بری برعکس استفاده شده است.

Shape rec3 = new Rectangle(6,6,6,6) صورت درست:

6. Polygon tri1 = new Triangle(2,2,2);

درست است. به همان علت شماره 2

7. Triangle tri2 = new Triangle(4,4,6);

این خط نیز بدیهی است که شی از با ارجاع از نوع خودش ایجاد کنیم.

8. Shape tri3 = new Triangle(2,2,2);

همانند شماره 2

.....
حال اگر ایرادات بالا را رفع کنیم به بررسی موارد cast کردن می‌رسیم:

9. circle1 = circle2;

درست است. در این حالت تغییر ارجاع circle1 را انجام داده ایم. یعنی هر دو circle1 و circle2 به یک شی ارجاع می‌دهند.

10. rect2 = rect3;

همانند 9

11. tri1 = tri3;

همانند 9

12. cricle2 = tri3;

غلط است زیرا هیچ رابطه وراثتی میان این دو شی وجود ندارد.

13. tri3 = tri2;

همانند 9

14. rect3 = new Shape(2,3,2);

غلط است زیرا رابطه وراثت را برعکس استفاده کرده است. یعنی Rectangle فرزند Shape است نه برعکس.

صورت صحیح: `rect3 = new Rectangle(2,3,2)`

15. `System.out.println(rect3.toString());`

درست است. چون Rectangle در خود دارای متد `toString` است که `Override` شده این متد در کلاس Polygon است.

سوال: اکنون با توضیحات داده شده، طوری دستورات نوار ابزار را اجرا کنید که به داخل تابع رفته و تنها دستور اول آن را اجرا کند و سپس به دستور بعد از نقطه وقفه برود.

با استفاده از `shift+F8` زمانی که برنامه به خط مورد نظر رسید خط را اجرا کرده و از تابع خارج میشود و دستورات بعد از خط فراخوانی تابع را انجام میدهد.

سوال: آیا میدانید چرا به اشکالهای نرمافزاری اصطلاحاً `bug` گفته میشود؟

Bug به معنای یک فضای خالی یا خلع میباشد. که در واقع در اینجا به معنای خلع و ایراد در کد نویسی میباشد.