

خلاصه:

همیشه میتوانیم یک ارجاع از کلاس والد یا پدر بسازیم که به یک شیاز subclass خود اشاره کند.

این کار از آن جهت مفید می باشد که میتوان ارجاع های زیادی از کلاس پدر ساخت که به subclass های مختلف اشاره کند.

مثلا در این موضوع، ارجاعی از Shape که از کلاس پدر می باشد، میتواند به شی ای از subclass های خود مانند Rectangle, Triangle, Circle و ... اشاره کند. این کار از تکرار کردن کد جلوگیری، و استفاده مجدد از کد را افزایش می دهد.

در این قطعه کد در خط اول ما نمیتوانیم ارجاعی از یک subclass داشته باشیم که به یک شی ای از superclass اشاره کند! در واقع حالت عکس آن وجود دارد؛ یعنی ارجاعی از Shape میتواند به شی ای از زیر کلاس خود یعنی Circle اشاره نماید.

خط دو صحیح می باشد و ارجاعی از کلاس پدر یا والد به شی ای از subclass خود اشاره می کند.

در خط سوم ارجاعی از شی Rectangle نمیتواند به شی ای از کلاس Triangle اشاره کند و غلط می باشد.

خط چهارم صحیح می باشد. میتوانیم ارجاعی از Polygon داشته باشیم که به شی ای از زیر کلاس خود اشاره میکند.

خط پنج نیز به دلایلی که برای خط اول گفته شد اشتباه می باشد.

خط ششم صحیح می باشد. به دلایلی که برای خط 4 گفته شد.

خط هفتم و هشتم نیز صحیح می باشد.

خلاصه ی قسمت دوم:

هیچوقت نمیتوانیم ارجاعی از یک superclass را برابر با ارجاعی از یک subclass قرار بدهیم. برای انجام این کار نیاز به cast می باشد؛ یعنی قبل از ارجاعی از superclass نام subclass را که قصد داریم به آن تبدیل شود را می نویسیم.

ولی میتوانیم ارجاعی از subclass را برابر با superclass قرار دهیم.

در اینجا سه خط نهم، دهم و یازدهم نیاز به cast دارند؛ به دلیل اینکه هر سه تا خط، ارجاعی از superclass می باشد که میخواهیم آن ها را برابر با ارجاعی از subclass قرار دهیم.

اما دو خط بعد به cast نیاز نیست. به دلیل اینکه میتوانیم ارجاعی از subclass را برابر با superclass قرار دهیم.

```
public static void main (String... args) {
```

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

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

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

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

```
    Shape rect3 = new Rectangle(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 = (Circle)circle2;
```

```
    rect2 = (Rectangle)rect3;
```

```
    tri1 = (Triangle)tri3;
```

```
    circle2 = circle1;
```

```
tri3 = tri2;
```

```
rect3 = new Rectangle(2,3,2);
```

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

```
}
```