

(1) اشباع: هر وقت یک کلاس یا متد Abstract ایجاد کنیم به این معناست که اکثر خواص

از یک کلاس برای ایجاد کنیم در abstract می نویسیم و داخل کلاس های child متد را overriding

می کنیم و متد ها رو override می کنیم و همیشه به این شکل باید و این اوریدرل این متد

abstract class order:

که از کلاس برای ایجاد متد مثال:

```
{
    private Date date;
```

```
}
```

```
}
```

```
abstract class Li;
```

```
}
```

کپسوله سازی: به معنی سازی اطلاعات از دید کاربر کپسوله سازی می گویند.

یعنی خصوصیات و متد ها درون کلاس ها و از دید کاربر مخفی کنیم تا قابل تغییر دلس نباشد.

اما همچنان از آن ها برای اجرای دستر استفاده می شود که با استفاده از access modifier

های مثل: public, private, protected این کار صورت می گیرد.

ارث پذیری: هرگاه یک کلاس تمام خصوصیات اعم از متد ها و ویژگی ها را از کلاس دیگر

بدرست پیروی کنیم که به کلاس اول کلاس فرزند و به معنی کلاس به ارث می نشیند و همچنین

و توان رفتار بیشتری در کلاس فرزند اضافه کرده

چندرسانی: از کلمات یونانی poly (چند) و ریشه morphism گرفته شده و هرگاه نام یک

متدی را برای انواع مختلف استفاده کنیم به کار برده می شود چندرسانی. Method overloading

Honar

operator overloading

class Animal :

(2)

def breathe(self):

pass

def Walk(self):

pass

class Dog (Animal):

super().__init__(Animal)

def run(self):

pass

class Sparrow (Animal):

super().__init__(Animal)

def fly(self):

pass

این تابع بازنویسی شده است
در کلاس Sparrow

(3)

در این سوال از مفهوم method overloading استفاده شده است که در پایتون ساپورت نمی‌شود

خوبی توابع طبق انتظار ما باید با توجه به ورودی‌ها این چنین باشد :

~~animal.Walk()~~ → walking...

~~animal.Walk(40)~~ → 40 minutes walking...

~~animal.Walk(40,60)~~ → 50 minutes

در اینجا مقادیر پس از فاصله انداز می‌شود

30 minutes and 59 seconds x walking...

40 به جای 30 در دست‌آورد می‌شود → 40 minutes and 59 seconds x walking...

حقیقت 50 و 60 جایگزین می‌شوند → 50 minutes and 60 seconds x walking...

مقادیر پس از فاصله می‌شوند

که در اینجا به ترتیب برای ورودی‌ها و مقادیر پس از فاصله این خطا چاپ می‌شوند