

Time	Narration
00:01	به Spoken Tutorial در Subclassing و Method overriding خوش آمدید.
00:06	در این برنامه ما subclassing کلیدواژه extends و method overriding را یاد می گیریم.
00:15	در اینجا ما از Ubuntu Linux version 12.04 JDK 1.7 Eclipse 4.3.1 استفاده می کنیم.
00:25	برای این برنامه شما باید با اصول Java و Eclipse IDE آشنایی داشته باشید.
00:32	اگر نه برای برنامه‌های مربوطه به وب سایت ما مراجعه کنید.
00:37	ابتدا یاد می‌گیریم که subclassing چیست
00:41	این یک روش برای ایجاد class جدید از class موجود می باشد.
00:46	Class جدید ایجاد شده subclass یا derived class یا child class می باشد.
00:53	Class که از قبل موجود است super class یا base class یا parent class نامیده می شود.
01:00	حالا نشان می‌دهم که چگونه subclass را ایجاد کنید. من از قبل project با نام MyProject را ایجاد کرده ام.
01:10	من یک class با نام Employee را ایجاد کرده ام.
01:15	این شامل variables, name و email_address می باشد.
01:19	این همچنین شامل setter و getter methods برای class می باشد.
01:24	این "getDetails ()" method را دارد. این method که name و email_address را می دهد.
01:31	حالا به manager class می آیم.
01:35	این شامل variables, name, email_address و department می باشد.
01:40	می‌بینیم که بعضی از متغیرها برای هر دو Employee و Manager class مشترک هستند.

01:47	و name و email_adress نیز موجود هستند. در Employee class می بینیم که Manager class نیز موجود است.
01:57	پس Manager class را می توانیم بعنوان subclass از Employee class در نظر بگیریم.
02:03	برای آن ما باید تغییراتی در Manager class بدهیم.
02:08	بعد از public class Manager شما extends employee را تایپ کنید.
02:14	ما از کلیدواژه extends برای ایجاد subclass از class موجود استفاده می کنیم.
02:21	Variables های تکراری مشترک در هر دو classes را حذف کنید.
02:26	پس name و email_address از Manager class را حذف کنید.
02:32	همچنین setter methods و getter methods از همان را نیز حذف کنید.
02:37	در class Manager ما فقط یک متغیر department را خواهیم داشت.
02:43	ما همچنین setter و getter methods برای department را داریم.
02:59	با این روش manager class اعضای Employee class را می گیرد.
02:55	این روش از گسترش یک class از دیگری single inheritance نامیده می شود.
03:02	من همچنین یک class دیگر با نام TestEmployee را ایجاد کرده ام.
03:08	در main method ما object از Manager class را ایجاد خواهیم کرد.
03:14	در main method شما Manager manager مساوی new Manager پراترها را تایپ کنید.
03:23	سپس ما setter method از Manager class را می خوانیم.
03:28	پس manager dot setName بین کروشه و double quotes شما Nikkita Dinesh را تایپ کنید.
03:38	سپس manager dot setEmail بین کروشه و double quotes شما abc at gmail dot come را تایپ کنید.
03:49	سپس manager dot setDepartment بین کروشه و double quotes شما Accounts را تایپ کنید.
03:57	شما می توانید هر name, email address و department را استفاده کنید.
04:02	حالا method getDetatails() را با استفاده از Manager object می خوانیم.
04:08	پس System.out.println بین کروشه manager dot getDetails را تایپ کنید.
04:17	حالا برنامه را save و run می کنیم.
04:21	ما خروجی

	<p>Name: Nikkita Dinesh</p> <p>Email: abc@gmail.com را بدست می آوریم.</p>
04:30	در اینجا object از Manager class که method <code>getDetails()</code> را می خواند.
04:36	حالا به Manager class بیاييد.
04:39	می بینیم که در اینجا هیچ <code>getDetails()</code> method نمی باشد.
04:43	اما ما هنوز خروجی را بدست می آوریم. چون Manager class که Employee class را گسترش (extend) می دهد.
04:52	Manager class بطور خودکار متغیرها و methods از Employee class را می گیرد.
04:59	پس این parent class که Employee است را بررسی می کند.
05:04	به Employee class می آییم. این <code>getDetails()</code> method را اینجا پیدا می کند.
05:11	توجه کنید که ما department را نداده ایم در نتیجه این department را در خروجی چاپ نمی کند.
05:20	حالا <code>getDetails</code> method را به <code>private</code> تغییر می دهیم. فایل را <code>save</code> کنید.
05:27	می بینیم که ما اشتباه کامپایلشن در <code>TestEmployee</code> class را داریم.
05:34	که می گوید: <code>The method getDetails() from the type Employee is not visible</code>
05:40	یعنی نمی توانیم به <code>getDetails()</code> method دسترسی داشته باشیم.
05:45	چون ما <code>getDetails()</code> method را بعنوان <code>private</code> اعلام کرده ایم.
05:52	<code>Subclass</code> اعضای <code>private</code> را از <code>superclass</code> نمی گیرد.
05:58	<code>Subclass</code> نمی تواند مستقیماً به اعضای <code>private</code> از <code>superclass</code> دسترسی داشته باشد.
06:04	<code>Superclass</code> می تواند متدهای <code>public</code> یا <code>protected</code> را داشته باشد.
06:09	این <code>methods</code> می توانند به <code>private fields</code> هایشان دسترسی داشته باشند.
06:13	<code>Subclass</code> همچنین می تواند به <code>private fields</code> از طریق این <code>methods</code> دسترسی داشته باشد.
06:18	پس آن را دوباره به <code>public</code> تغییر می دهیم.
06:21	حالا <code>getDetails</code> method را به Manager class اضافه می کنیم.
06:27	این <code>method</code> که <code>name</code> , <code>email_address</code> و <code>department</code> را می دهد.
06:33	پس <code>getDetails</code> <code>public String</code> پراپرتی ها را تایپ کنید.

06:39	داخل method شما return بین گروهه Name به اضافه () getName() به اضافه slash n به اضافه Email به اضافه () getEmail() به اضافه slash n به اضافه Manager of به اضافه () getDepartment و semicolon را تایپ کنید. فایل را save کنید.
07:07	توجه کنید که حالا ما method getDetails را در هر دو Employee class و Manager داریم.
07:15	و name, return type و argument list از method برای هر دو classes یکی می باشد.
07:22	یک method در subclass گفته می شود که method در parent class را override کرده است اگر name return type argument list کاملاً تطبیق داده شوند.
07:33	به Manager class برگردید.
07:36	قبل از method getDetails() شما @Override را تایپ کنید.
07:43	این یک override annotation است. این نشان می دهد که method برای override کردن method در superclass در نظر گرفته شده است.
07:53	حالا می بینیم که annotation چیست.
07:57	:Annotation با علامت کاراکتر at (@) شروع می شود. Data در مورد برنامه را فراهم می کند. هیچ تأثیر مستقیمی روی عملکرد کد ندارد.
08:10	اگر method با @Override annotated شده باشد، کامپایلر error (اشتباه) می دهد اگر این method یک method که در superclass اعلام شده است را override کند.
08:23	Method signature در superclass آن متفاوت باشد.
08:28	حالا به IDE بر می گردیم. به Manager class برگردید.
08:34	علامت کاراکتر at (@) نشان می دهد که کامپایلر چیزی را که دنبال می کند annotation است.

08:42	اینجا این نشان می‌دهد که method gtDetails() که overridden شده است.
08:48	به TestEmployee class برمی‌گردیم.
08:51	فایل را save و برنامه را run کنید.
08:55	ما output (خروجی) را Name: Nikkita Dinesh Email: abc@gmail.com Manager of Accounts بدست می‌آوریم.
09:05	اینجا object از Manager class که method gtDetails() را می‌خواند.
09:11	اما این دفعه این method از خود Manager class را می‌خواند.
09:16	با این روش ما method از parent class را با subclass خود override کردیم.
09:23	خلاصه می‌کنیم. در این برنامه ما subclassing و Method Overriding را یاد گرفتیم.
09:31	ارائه: Vehicle class که شامل method run که The Vehicle is running را چاپ می‌کند می‌باشد را ایجاد کنید.
09:40	همچنین یک class Bike که method run را دارد و The Bike is running safely را چاپ می‌کند، ایجاد کنید.
09:48	خروجی باید The Bike is running safely باشد.
09:52	برای دانستن بیشتر Spoken Tutorial Project ویدئو را در لینک زیر ببینید. این خلاصه پروژه می‌باشد. اگر پهنای باند خوبی ندارید ابتدا دانلود و سپس مشاهده کنید.
10:06	تیم این پروژه کارگاه آموزشی استفاده از Spoken Tutorial ارائه می‌دهد. و به کسانی که آزمون آنلاین را قبول شوند، گواهینامه می‌دهد. برای جزئیات بیشتر لطفاً به contact AT spoken HYPHEN tutorial DOT org ایمیل بفرستید.
10:21	Spoken Tutorial Project بخشی از پروژه Talk To a Teacher می‌باشد. که تحت پشتیبانی National Mission on Education از طریق ICT توسط MHRD دولت هند می‌باشد.

	اطلاعات بیشتر در لینک spoken hyphen tutorial dot org Slash NMEICT hyphen Intro دسترس می باشد:
10:42	ترجمه و صداگذاری شبیم اقبال خان. با تشکر از شما