

Time	Narration
00:01	به Spoken Tutorial در Java Interfaces خوش آمدید.
00:05	در این برنامه این موارد را یاد می گیریم: ایجاد کردن interface
00:10	ایجاد کردن Implementation classes و استفاده از interface
00:16	برای این برنامه من از Ubuntu 12.04 و JDK 1.7 Eclipse 4.3.1 استفاده می کنم.
00:28	برای این برنامه شما باید با Java و Eclipse IDE آشنایی داشته باشید.
00:36	همچنین باید با subclassing و Abstract classes در Java نیز آشنایی داشته باشید.
00:42	اگر نه برای برنامه های مربوطه به لینک نشان داده شده مراجعه کنید.
00:48	ابتدا می فهمیم که interface چیست.
00:52	یک Interface شامل گروهی از abstract methods و static data members می باشد.
00:58	این signatures گروهی از methods بدون body (بدنه) را تعریف می کند.
01:04	این استفاده از کلیدواژه interface را اعلام می کند.
01:08	حال به Eclipse می رویم و یک پروژه جدید با نام InterfaceDemo را ایجاد می کنیم.
01:15	در اینجا ما classes و interface لازم برای نشان دادن استفاده از interfaces را ایجاد می کنیم.
01:24	و src folder را کلیک راست کنید و New > Interface را کلیک کنید.
01:30	نام را Animal تایپ کنید و Enter را فشار دهید.
01:34	توجه کنید که کلیدواژه interface برای تعریف کردن interface استفاده می شود.
01:39	حالا کد را همانطور که در screen نمایش داده شده تایپ کنید.
01:43	در اینجا نام interface که Animal می باشد.
01:46	این شامل سه Abstract methods می باشد, see(), talk () و move()
01:52	همه این methods در یک interface که public و abstract می باشند.

01:59	یک interface همچنین می‌تواند شامل اعلام constant variable نیز باشد.
02:04	در اینجا مقدار(ثابت) constant string که Mammal است به متغیر type1 داده شده است.
02:12	و Reptiles به متغیر type2 داده شده است.
02:16	همه مقادیر constant که در یک interface تعریف شده‌اند public, static و final هستند.
02:25	سپس ما implementation class برای یک interface به همراه مثال را می‌بینیم.
02:32	در اینجا Human یک class است که Animal interface را اجرا می‌کند.
02:38	پس این باید اجرا را برای متدهای see(), talk() و move() فراهم کند.
02:45	یک class همچنین می‌تواند interface های چندگانه را پیاده‌سازی کند.
02:49	همانطور که در مثال نشان داده شده است class Human دو interface های Animal و Action را اجرا می‌کند.
02:57	توجه کنید که comma operator که در syntax (ساختار) استفاده شده است برای مشخص کردن interface های مختلف می‌باشد.
03:04	حالا این class باید پیاده‌سازی را برای همه abstract methods در هر دو interface های Animal و Action فراهم می‌کند.
03:13	تصویر در اینجا (رابطه پیاده سازی) implement relation را نشان می‌دهد.
03:18	Animal class یک interface است.
03:22	کلاسهای Human و Snake که دو implementation classes هستند.
03:28	Human class که پیاده‌سازی های مختلف خود را برای متدهای see(), talk () و move() فراهم می‌کند.
03:36	Snake class که پیاده‌سازی های مختلف خود را برای متدهای see(), talk () و move() فراهم می‌کند.
03:45	حالا استفاده از interfaces را با یک برنامه یاد می‌گیریم.
03:50	و default package را کلیک راست کنید و یک class با نام Human را ایجاد کنید.
03:56	حالا برای ساختن این بعنوان implementation class از Animal این را تایپ کنید: implement Animal
04:04	حالا یک اشتباه که با Eclipse IDE می‌آید را می‌بینیم.
04:09	این اشتباه نشان می‌دهد که ما باید یک implementation برای Animal interface را فراهم کنیم.
04:15	ببینیم چگونه این اشتباه را تصحیح کنیم.
04:19	حالا متدهای see(), talk () و move() را تعریف می‌کنیم.

04:23	پس () talk public void بین آکولاد System.out.println و بین quotes شما I am a human and I belong to را تایپ کنید.
04:37	ما حالا می‌توانیم مقادیر static, final variable type1 که در Animal interface اعلام شده است را استفاده کنیم.
04:45	پس + Animal.type1 + بین quotes شما family semicolon را تایپ کنید.
04:54	حالا see() method را پیاده‌سازی می‌کنیم.
04:57	پس () see public void بین آکولاد System.out.println بین quotes شما I can see all colors semicolon را تایپ کنید.
05:11	ما همچنین move() method را تعریف می‌کنیم.
05:14	پس () move public void بین آکولاد System.out.println بین quotes شما I move by walking semicolon را تایپ کنید.
05:29	توجه کنید وقتی که همه methods ها پیاده‌سازی شوند , اشتباه ناپدید می‌شود.
05:34	سپس ما می‌بینیم که چگونه snake class را تعریف کنیم.
05:38	من از قبل در project خود آن را ایجاد کرده‌ام.
05:42	لطفاً snake class را در project خود ایجاد کنید و این کد را همانطور که نمایش داده شده است تایپ کنید.
05:49	حالا کد را بررسی می‌کنیم.
05:52	ما می‌بینیم که همه methods های () see(), talk(), Animal interfce- و () move در این class پیاده‌سازی شده‌اند.
06:01	در اینجا talk() method که I am a snake and I belong to را پرینت می‌کند.
06:07	سپس مقدار Animal.type2 باید پرینت شود و سپس family
06:13	در اینجا see() metod که I can see only in black and white را پرینت می‌کند.
06:19	و move() method که I move by crawling را پرینت می‌کند.
06:23	توجه کنید که Human class پیاده‌سازی خود را برای متدهای () see(), talk() و () move دارد.
06:31	و Snake class پیاده‌سازی های خود را برای متدهای () see(), talk() و () move دارد.
06:39	حالا package پیش‌فرض را کلیک راست کنید و new > class را کلیک کنید و سپس نام را Demo تایپ کنید.
06:47	داخل این class ما main method را خواهیم داشت.
06:51	پس main را تایپ کنید و ctrl+space را برای ایجاد main method فشار دهید.

06:58	این کد را همانطور که در صفحه نمایش داده شده است تایپ کنید.
07:01	در این خط ما Human class را با استفاده از Animal interface کشف می کنیم.
07:07	این بعنوان Animal h مساوی new Human(); نمایش داده می شود.
07:14	حالا ما می توانیم متدهای مختلف را با استفاده از این object بعنوان h.move(); h.see(); h.talk(); فراخوانی کنیم.
07:26	سپس ما Snake class را با استفاده از Animal interface کشف می کنیم.
07:31	حالا ما متدهای مختلف را با استفاده از این object همانطور که نشان داده شده است فراخوانی می کنیم.
07:38	حالا این برنامه Demo را run می کنیم.
07:41	پس class Demo را کلیک راست کنید و Run as > Java Application را انتخاب کنید.
07:48	ما می توانیم خروجی را ببینیم.
07:52	اینها توسط متدهای see(), talk(), و move() که با استفاده از h object human class فراخوانی شده اند، پرینت شده اند.
08:00	اینها توسط متدهای see(), talk(), و move() که با استفاده از s object Snake class فراخوانی شده اند، پرینت شده اند.
08:08	حال تفاوت های بین interface و یک abstract class را بررسی می کنیم.
08:14	همه متدها در یک interface باید abstract باشند.
08:18	داخل interface نباید هیچ main و constructors, concrete methods, static methods method باشد.
08:23	
08:28	اما یک abstract class می تواند همه اینها را در داخل خود داشته باشد.
08:32	متغیر ها در یک interface باید static و final باشند.
08:38	همچنین محدودیتی برای یک abstract class نمی باشد.
08:43	این ما را به پایان برنامه می آورد، خلاصه می کنیم.
08:48	در این برنامه این موارد را یاد گرفتیم: ایجاد کردن یک Interface
08:53	ایجاد کردن implementation class و
08:56	استفاده از interfaces
08:59	ارائه: یک interface Vehicle را که شامل متدهای brake() و run() می باشد، ایجاد کنید.
09:07	یک interface دیگر Fuel که شامل متدهای fill(String type,int

	(int quantity,int price), pay(int quantity), quantity) می‌باشد را ایجاد کنید.
09:19	یک subclass Car را که هر دو interfaces های Vehicle و Fuel را پیاده‌سازی می‌کند, ایجاد کنید.
09:26	اینجا brake method باید Car Applies Power brake را پرینت کند.
09:30	و run method باید Car is rrunning on 4 wheels را پرینت کند.
09:35	و fill() method می‌تواند نوع و مقدار fuel (سوخت) پر شده را پرینت کند.
09:41	به طور مثال: 10 لیتر بنزین
09:44	Pay() method برای پرینت کردن نرخى که باید پرداخت شود می‌تواند استفاده شود مثلاً Rs.640 پرداخت کنید.
09:53	یک subclass دیگر Bike که دوباره هر دو interfaces های Vehicle و Fuel را پیاده‌سازی می‌کند را ایجاد کنید.
10:00	در اینجا brake method می‌تواند Bike Applies hand brake را پرینت کند.
10:05	و run method می‌تواند Bike is running on 2 wheels را پرینت کند.
10:10	سپس متدهای fill() و pay() را همانطور که قبلاً توضیح داده شد, پیاده‌سازی کنید.
10:15	در آخر Demo class که شامل main method برای بررسی نتایج می‌باشد را ایجاد کنید.
10:21	ویدئو در لینک زیر خلاصه Spoken Tutorial Project می باشد, لطفاً دانلود و مشاهده کنید.
10:29	تیم این پروژه کارگاه آموزشی استفاده از Spoken Tutorial ارائه می دهد. و به کسانی که آزمون آنلاین را قبول
10:38	شوند, گواهینامه می‌دهد . برای جزئیات بیشتر لطفاً به ما ایمیل بفرستید .
10:41	بودجه Spoken Tutorial Project توسط NMEICT, MHRD دولت هند تأمین می شود.
10:48	اطلاعات بیشتر در لینک نشان داده شده در دسترس می باشد.
10:52	این script توسط Department of Information Technology, Amal Jyothi College of Engineering منتشر شده است.
11:01	ترجمه و صداگذاری شبنم اقبال خان. با تشکر از شما