آموزش زبان برنامه نویسی جاوا

کلمه کلیدی final

جلسه بیست و یکم

نویسنده : رحمان زارعی

جاوا را ساده،آسان و شیرین بنوشید!!!!

کلمه کلیدی final برای محدود کردن کاربر استفاده می شود.کلمه کلیدی final جاوا را می توان در همه زمینه ها استفاده کرد.

کلمه کلیدی final در موارد زیر استفاده می شود:

۱. کلاس
۲. متدها
۳. متغیرها

تعریف یک متغیر از نوع final

نحوه نوشتن یک متغیر از نوع فاینال بصورت زیر است:

```
final int variable;
```

همان طور که گفتیم کلمه کلیدی final برای متغیر ها بکار می رود،وقتی یک متغیر از نوع final تعریف و مقدار دهی اولیه می شود دیگر در تمام بخش های برنامه مقدار این متغیر ثابت و غیر قابل تغیر خواهد بود.

```
package javalike;

public class Rahman {

	final int a = 5;

	public static void main(String[] args) {

		Rahman r = new Rahman();
```

```
            r.a = 6;
            System.out.println(r.a);

        }

}
```

خروجی:

```
خطای کامپایل!!!!!!
```

- همان طور که گفتیم یک متغیر وقتی از نوع final تعریف و مقداردهی اولیه می شود در تمام بخش های برنامه اجازه تغییر مقدار متغیر را نخواهیم داشت. در این کد برنامه چون متغیر a از نوع final و مقدار اولیه ان ۵ بوده و در متد main قصد تغییر مقدار این متغیر را داشته ایم برنامه خطای کامپایل داده.

پس اگر ما یک متغیر را از نوع final تعریف کردیم ،دیگه در بخش های برنامه نمی توانیم مقدار متغیر را تغییر دهیم. به این متغیری که نوع final تعریف بشه به اصلاح به ان متغیر ،متغیر ثابت یا constant می گویند.

تعریف یک متد از نوع final

نحوه نوشتن یک متد از نوع فاینال بصورت زیر است:

```
public final void myMethod(){

}
```

اگر شما یک متد را از نوع final تعریف کردید،دیگر این متد را نمی توان override کرد!!!

میدونم این مفهوم ناآشناست ، صرفا جهت مقدمه چینی گفتم، در جلسه بعد مفصل در مورد override صحبت خواهیم کرد پس مثال و توضیح را به جلسه ای که کاملا مربوط به override هستش واگذار میکنیم.

این override خوب یادتون باشه پس ☺

وقتی یک کلاس را final تعریف میکنیم!

اگر شما یک کلاس را final تعریف کنید ،دیگر نمی توانید آن را به ارث ببرید!

```
package javalike;

final class Bike {
}

class Honda1 extends Bike {
	void run() {
		System.out.println("running safely with 100kmph");
	}

	public static void main(String args[]) {
		Honda1 honda = new Honda();
		honda.run();
	}
}
```

خروجی:

```
خطای کامپایل!!!!!!!!
```

- در اینجا چون کلاس Bike از نوع final هستش دیگه نمی توان این کلاس توسط کلاس دیگر به ارث برده شود.

آیا متدی که از نوع final تعریف شود را می توان به ارث برد؟

پاسخ: بله،متد final را می توان به ارث برد اما نمی توان آن را override کرد.

```
package javalike;

class Bike {
	final void run() {
		System.out.println("running...");
	}
}

class Honda2 extends Bike {
	public static void main(String args[]) {
		new Honda2().run();
	}
}
```

- در اینجا کلاس Honda2 کلاس Bike را به ارث برده است.

- متد run درون کلاس Bike از نوع final هستش. همان طور که میدونید کلاس Honda2 کلاس Bike را به ارث برده است یعنی به تمام متدها و متغیرهای آن دسرسی دارد.و در این جا با ساختن شی از کلاس Honda2 متد run را با وجود final بودن صدا زدیم و هیچ مشکلی پیش نیامد.

آیا می شود یک متغیر از نوع final را هنگام تعریف خالی یا بدون مقدار ایجاد کنیم؟

بله.تنها در یک صورت می شود یک متغیر از نوع final بدون مقدار دهی اولیه ایجاد کرد،وقتی متغیر خود را از نوع final تعریف کردید در سازنده کلاس خود آن متغیر را مقدار دهی کنید.

```
package javalike;

public class Rahman {

	final int a;

	Rahman() {
		a = 5;
	}

	public static void main(String[] args) {

		Rahman r = new Rahman();

		System.out.println(r.a);

	}

}
```

خروجی:

```
5
```

- اینجا متغیر a را از نوع final بدون مقدار دهی اولیه تعریف کرده ایم.برای این که برنامه ما دچار خطای کامپایل نشود تنها راه ما برای مقدار دهی به متغیر از نوع final درون سازنده کلاس هستش که در اینجا متغیر a در سازنده کلاس مقدار ۵ را بهش نسبت داده ایم.
- این نکته را در نظر بگیرید تنها در صورتی میشود یک متغیر از نوع final را درون سازنده کلاس مقداردهی کرد که مقدار اولیه متغیر final ما خالی باشد.همان طور که می بینید متغیر a هنگام تعریف به ان هیچ مقداری داده نشده و به اصلاح خالی می باشد.

وقتی یک متغیر هم از نوع final باشد هم از نوع sataic :

بله.اگر متغیر ما از نوع final و static و همچنین هنگام تعریف آن را مقدار دهی اولیه نکردیم و به اصلاح خالی بود دیگر نمی توان در سازنده کلاس آن را مقدار دهی کرد و تنها راه مقدار دهی آنها این هست که در یک بلوک استاتیک آن متغیر را مقدار دهی کنیم!!!

- دوستان اینو در نظر بگیرید بعضی از مفاهیم کاربرد و استفاده زیاد نداره مثل همین عنوان اما حالا امکان داره در برنامه خود به ان برخورد کنید گفتمش وگرنه من خودم هم هیچ علاقه ای بهش ندارم ☺

یادآوری:

نحوه نوشتن بلوک استاتیک بصورت زیر است:

```
static {

}
```

ابتدا کلمه استاتیک بعد دو بلوک یا آکولاد {} را باز و بسته می کنیم و می توانیم دستورات خود را درون بدنه و دربین دو آکولاد بنویسیم.

```
package javalike;

public class Rahman {

	final static  int a;

	static {
		a = 5;
	}

	public static void main(String[] args) {

		Rahman r = new Rahman();

		System.out.println(r.a);

	}

}
```

خروجی:

5

- در اینجا متغیر a هم از نوع final هستش و هم از نوع static ،چون هنگام تعریف مقدار دهی اولیه نشده و خالی می باشد،تنها راه مقدار دهی آن در بخشی از برنامه این است که درون یک بلوک استاتیک آن را مقداردهی کنیم.

میخواستم فقط ی سوالی کرده باشم و این که آیا میشه یک سازنده را از نوع final تعریف کنیم؟! 😀 خیر

پیروز و موفق باشید
