鄙人的新书《Elasticsearch权威指南》正式出版发行，欢迎购买！本书由华为、中兴高级技术专家全面审读并撰序，助您挑战百万年薪 购书链接： 《Elasticsearch权威指南》

欢迎关注鄙人公众号，技术干货随时看！

  在实际工作中，我们很可能习惯性地选择Runnable或Thread之一直接使用，根本没在意二者的区别，但在面试中很多自以为是的菜货面试官会经常而且非常严肃的问出：请你解释下Runnable或Thread的区别？尤其是新手就容易上当，不知如何回答，就胡乱编一通。鄙人今天告诉你们这二者本身就没有本质区别，就是接口和类的区别。问出这个问题的面试官本身就是个二流子！如果非要说区别，请看如下：

Runnable的实现方式是实现其接口即可

Thread的实现方式是继承其类

Runnable接口支持多继承，但基本上用不到

Thread实现了Runnable接口并进行了扩展，而Thread和Runnable的实质是实现的关系，不是同类东西，所以Runnable或Thread本身没有可比性。

  网络上流传的最大的一个错误结论：Runnable更容易可以实现多个线程间的资源共享，而Thread不可以！ 这是一个二笔的结论！网络得出此结论的例子如下：

//program--Thread

public class Test {

public static void main(String[] args) {

// TODO Auto-generated method stub

new MyThread().start();

new MyThread().start();

}

static class MyThread extends Thread{

private int ticket = 5;

public void run(){

while(true){

System.out.println("Thread ticket = " + ticket--);

if(ticket < 0){

break;

}

}

}

}

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

运行结果如下：

Thread ticket = 5

Thread ticket = 5

Thread ticket = 4

Thread ticket = 3

Thread ticket = 2

Thread ticket = 1

Thread ticket = 0

Thread ticket = 4

Thread ticket = 3

Thread ticket = 2

Thread ticket = 1

Thread ticket = 0

Process finished with exit code 0

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

  很显然，总共5张票但卖了10张。这就像两个售票员再卖同一张票，原因稍后分析。现在看看使用runnable的结果：

//program--Runnable

public class Test2 {

public static void main(String[] args) {

// TODO Auto-generated method stub

MyThread2 mt=new MyThread2();

new Thread(mt).start();

new Thread(mt).start();

}

static class MyThread2 implements Runnable{

private int ticket = 5;

public void run(){

while(true){

System.out.println("Runnable ticket = " + ticket--);

if(ticket < 0){

break;

}

}

}

}

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

  运行结果如下：

Runnable ticket = 5

Runnable ticket = 4

Runnable ticket = 3

Runnable ticket = 1

Runnable ticket = 0

Runnable ticket = 2

Process finished with exit code 0

1

2

3

4

5

6

7

8

9

10

  嗯，嗯，大多数人都会认为结果正确了，而且会非常郑重的得出：Runnable更容易可以实现多个线程间的资源共享，而Thread不可以！ 真的是这样吗？大错特错！

  program–Thread这个例子结果多卖一倍票的原因根本不是因为Runnable和Thread的区别，看其中的如下两行代码：

new MyThread().start();

new MyThread().start();

1

2

  例子中，创建了两个MyThread对象，每个对象都有自己的ticket成员变量，当然会多卖1倍。如果把ticket定义为static类型，就离正确结果有近了一步（因为是多线程同时访问一个变量会有同步问题，加上锁才是最终正确的代码）。

现在看program–Runnable例子中，如下代码：

MyThread2 mt=new MyThread2();

new Thread(mt).start();

new Thread(mt).start();

1

2

3

  只创建了一个Runnable对象，肯定只卖一倍票（但也会有多线程同步问题，同样需要加锁），根本不是Runnable和Thread的区别造成的。再来看一个使用Thread方式的正确例子：

public class Test3 extends Thread {

private int ticket = 10;

public void run(){

for(int i =0;i<10;i++){

synchronized (this){

if(this.ticket>0){

try {

Thread.sleep(100);

System.out.println(Thread.currentThread().getName()+"卖票---->"+(this.ticket--));

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

}

}

public static void main(String[] arg){

Test3 t1 = new Test3();

new Thread(t1,"线程1").start();

new Thread(t1,"线程2").start();

}

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

运行结果如下：

线程1卖票---->10

线程1卖票---->9

线程1卖票---->8

线程1卖票---->7

线程1卖票---->6

线程1卖票---->5

线程1卖票---->4

线程1卖票---->3

线程1卖票---->2

线程1卖票---->1

Process finished with exit code 0

1

2

3

4

5

6

7

8

9

10

11

12

13

  上例中只创建了一个Thread对象（子类Test3）,效果和Runnable一样。synchronized这个关键字是必须的，否则会出现同步问题，篇幅太长本文不做讨论。

  上面讨论下来，Thread和Runnable没有根本的没区别，只是写法不同罢了，事实是Thread和Runnable没有本质的区别，这才是正确的结论，和自以为是的大神所说的Runnable更容易实现资源共享，没有半点关系！

  现在看下Thread源码：

public

class Thread implements Runnable {

/\* Make sure registerNatives is the first thing <clinit> does. \*/

private static native void registerNatives();

static {

registerNatives();

}

private volatile String name;

private int priority;

private Thread threadQ;

private long eetop;

1

2

3

4

5

6

7

8

9

10

11

12

  可以看出，Thread实现了Runnable接口，提供了更多的可用方法和成员而已。

  结论，Thread和Runnable的实质是继承关系，没有可比性。无论使用Runnable还是Thread，都会new Thread，然后执行run方法。用法上，如果有复杂的线程操作需求，那就选择继承Thread，如果只是简单的执行一个任务，那就实现runnable。

  再遇到二笔面试官问Thread和Runnable的区别，你可以直接鄙视了！

————————————————

版权声明：本文为CSDN博主「阿童木-atom」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/zhaojianting/article/details/97664370