题目：设计一个类，然后只能生成该类的一个实例

```
public class Test02{
 /*单例模式，饿汉式，线程安全*/
 public static class Singleton{
 private final static Singleton INSTANCE = new Singleton();
 private Singleton(){

 }
 public static Singleton getInstance(){
 return INSTANCE;
  }
  }

  /*懒汉式*/
  public static class Singleton2{
  private static Singleton2 INSTANCE = null;
  private Singleton2(){}
  public static Singleton2 getInstance()
  {
  if (INSTANCE == null){
  INSTANCE = new Singleton2();
  }
  return INSTANCE;
  }
  }
  /*懒汉式，线程安全，多线程环境下效率不高*/
  public static class Singleton3{
  private static Singleton3 INSTANCE = null;
  private Singleton3(){

  }
  public static synchronized Singleton3 getInstance(){
  if(instance == null){
  instance = new Singleton3();
  }
```

```
return instance;
}
}
/*懒汉式，变种*/
public static class Singleton4{
private static Singleton4 instance = null;
static{
instance = new Singleton4();
}
private Singleton4(){

}
public static Singleton4 getInstance(){
return instance;
}
}
/*使用静态内部类*/
public static class Singleton5{
private final static class SingletonHolder{
private static final Singleton5 INSTANCE = new Singleton5();
}
private Singleton5{}
public static Singleton5 getInstance(){
return SingletonHolder.INSTANCE;
}
}
/*静态内部类，采用双重校验锁，线程安全,两个if循环保证在多线程条件下只创建一个实例*/
public static class Singleton6{
private Singleton6(){ }
private volatile static Singleton6 instance = null;
public static Singleton6 getInstance(){
if(instance == null){
synchronized(Singleton6.class)
{
 if(instance == null){
 instance = new Singleton6();
 }
 }
 }
 return instance;
 }
 }

}
```