

```

package com.sxt.prototype;

/**
 * 测试普通new方式创建对象和clone方式创建对象的效率差异
 * 如果需要短时间创建大量对象, 并且new的过程比较耗时。则可以考虑使用原型模式!
 * @author 江
 *
 */

public class Client04 {

    public static void testNew(int size) {
        long start=System.currentTimeMillis();
        for(int i=0;i<size;i++) {
            Laptop t=new Laptop();
        }
        long end=System.currentTimeMillis();
        System.err.println("new的方式创建耗时: "+(end-start));
    }

    public static void testClone(int size) throws CloneNotSupportedException
    {

        Laptop t=new Laptop();
        long start=System.currentTimeMillis();
        for(int i=0;i<size;i++) {
            Laptop temp=(Laptop) t.clone();
        }
        long end=System.currentTimeMillis();
        System.err.println("Clone的方式创建耗时: "+(end-start));
    }

    public static void main(String[] args) throws CloneNotSupportedException
    {

        testNew(1000);
        testClone(1000);
    }
}

```

```
}
```

```
class Laptop implements Cloneable{    //笔记本电脑
```

```
    public Laptop() {  
        try {  
            Thread.sleep(10);    //模拟创建对象耗时的过程!  
        } catch (InterruptedException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

```
@Override  
    protected Object clone() throws CloneNotSupportedException {  
        Object obj=super.clone();  
        return obj;  
    }  
}
```

```
}
```