Java 核心技术

第六章　static、final和常量设计

第五节 不可变对象和字符串

华东师范大学 陈良育

不可变对象(1)

- 不可变对象(Immutable Object)
  - 一旦创建，这个对象（状态/值）不能被更改了
  - 其内在的成员变量的值就不能修改了。
  - 典型的不可变对象
    - 八个基本型别的包装类的对象
    - String，BigInteger和BigDecimal等的对象
- 可变对象(Mutable Object)
  - 普通对象

不可变对象(2)


```
String a = new String("abc");
String b = a;
System.out.println(b);
a = "def";
System.out.println(b);
```

```
public static void change(String b)
{
    b = "def";
}

 a=new String("abc");
 change(a);
 System.out.println(a);
```

- 不可变对象，也是传指针(引用)
- 由于不可变，临时变量指向新内存，外部实参的指针不改动

不可变对象(3)

- 如何创建不可变对象
  - immutable对象是不可改变，有改变，请clone/new一个对象进行修改
  - 所有的属性都是final和private的
  - 不提供setter方法
  - 类是final的，或者所有的方法都是final
  - 类中包含mutable对象，那么返回拷贝需要深度clone

不可变对象(4)

- 不可变对象(Immutable Object)优点
  - 只读，线程安全
  - 并发读，提高性能
  - 可以重复使用
- 缺点
  - 制造垃圾，浪费空间

Java 字符串(1)

- 字符串是Java使用最多的类，是一种典型的不可变对象
- String定义有2种
  - String a = “abc”; //常量赋值，栈分配内存
  - String b = new String(“abc”); //new对象，堆分配内存
- 字符串内容比较：equals方法
- 是否指向同一个对象：指针比较==

Java 字符串(2)

- Java 常量池(Constant Pool)
  - 保存在编译期间就已经确定的数据
  - 是一块特殊的内存
  - 相同的常量字符串只存储一份，节省内存，共享访问

Java 字符串(3)

- 字符串的加法
- String a=“abc”;
- a = a+“def”;//由于String不可修改，效率差
- 使用StringBuffer/StringBuilder类的append方法进行修改
- StringBuffer/StringBuilder 的对象都是可变对象
- StringBuffer(同步，线程安全，修改快速),
  StringBuilder(不同步，线程不安全，修改更快)

Java 字符串(4)

- 查看StringAppendTest.java 比较字符串操作时间性能
- 查看StringPassingTest.java和ArgumentPassing.java 理解可变对象传参

总结

- 不可变对象提高读效率
- 不可变对象设计的方法
- 字符串append操作速度：StringBuilder>StringBuffer>String

代码(1) ImmutableObjectTest.java


```
public class ImmutableObjectTest {
    public static void main(String[] args) {
        String a = new String("abc");
        String b = a;
        System.out.println(b);
        a = "def";
        System.out.println(b);

        a=new String("abc");
        change(a);
        System.out.println(a);
    }
    public static void change(String b)
    {
        b = "def";
    }
}
```

代码(2) StringAppendTest.java


```java
import java.util.Calendar;

public class StringAppendTest {
    public static void main(String[] args) {
        int n = 50000;
        Calendar t1 = Calendar.getInstance();
        String a = new String();
        for(int i=0;i<n;i++)
        {
            a = a + i + ",";
        }
        System.out.println(Calendar.getInstance().getTimeInMillis() - t1.getTimeInMillis());

        Calendar t2 = Calendar.getInstance();
        StringBuffer b = new StringBuffer("");
        for(int i=0;i<n;i++)
        {
            b.append(i);
            b.append(",");
        }
        System.out.println(Calendar.getInstance().getTimeInMillis() - t2.getTimeInMillis());

        Calendar t3 = Calendar.getInstance();
        StringBuilder c = new StringBuilder("");
        for(int i=0;i<n;i++)
        {
            b.append(i);
            b.append(",");
        }
        System.out.println(Calendar.getInstance().getTimeInMillis() - t3.getTimeInMillis());
    }
}
```

代码(3) StringPassingTest.java


```java
public class StringPassingTest {

    public static void main(String[] args) {
        String a = "abc";
        changeValue(a);
        System.out.println(a);

        a = "abc";
        String b = a;
        a = "def";
        System.out.println(b);
    }
    public static void changeValue(String b)
    {
        b = "def";
    }
}
```

代码(4) ArgumentPassing.java


```
public class ArgumentPassing {
     public static void changeValue(int a)
     {
          a = 10;
     }
     public static void changeValue(String s1)
     {
          s1 = "def";
     }
     public static void changeValue(StringBuffer s1)
     {
          s1.append("def");
     }
     public static void main(String[] args) {
          int a = 5;           //基本类型
          String b = "abc"; //不可变对象
          StringBuffer c = new StringBuffer("abc"); //可变对象
          changeValue(a);
          changeValue(b);
          changeValue(c);
          System.out.println(a);
          System.out.println(b);
          System.out.println(c);
     }
}
```

代码(5) PrimitiveTypePassingTest


```
public class PrimitiveTypePassingTest {

    public static void main(String[] args) {
        int a = 5;
        System.out.println(a);
        changeValue(a);
        System.out.println(a);

        a=5;
        int b = 6;
        swap(a,b);
        System.out.println(a);
        System.out.println(b);
    }

    public static void changeValue(int a){
        a = 10;
    }

    public static void swap(int a, int b){
        a = a+b;
        b = a-b;
        a = a-b;
    }
}
```

代码(6) ObjectPassingTest.java


```
class A
{
	private int id;

	public int getId() {
		return id;
	}

	public void setId(int id) {
		this.id = id;
	}
}

public class ObjectPassingTest {
	public static void main(String[] args) {
		A a = new A();
		a.setId(10);
		System.out.println(a.getId());
		changeValue(a);
		System.out.println(a.getId());
	}

	public static void changeValue(A b) {
		b.setId(20);
		//b = new A();
	}
}
```

谢 谢！