arrayList学习笔记

# 一. arrayList初始化（new ArrayList（）方法）

1. 容器默认容量为10 （private static final int DEFAULT\_CAPACITY = 10;）

2. private static final Object[] EMPTY\_ELEMENTDATA = {}

3. private static final Object[] DEFAULTCAPACITY\_EMPTY\_ELEMENTDATA = {};

4. transient Object[] elementData =null ;

5. private int size = 0;

6. 容器最大容量为 2147483639 (private static final int MAX\_ARRAY\_SIZE = Integer.MAX\_VALUE - 8;)

7. elementData = {}, elementData.length=0 (this.elementData = DEFAULTCAPACITY\_EMPTY\_ELEMENTDATA )

8. 初始化完成

# 二.往容器里存储对象 (调用 list.add(E e)方法)

1.确保容器的容量可以添加对象 (调用ensureCapacityInternal(size + 1) 方法)

1.1 判断 elementData 是否== DEFAULTCAPACITY\_EMPTY\_ELEMENTDATA

如果为true：

则比较初始容量DEFAULT\_CAPACITY和size+1,取其中较大的数为容器的最小容量

code: if (elementData == DEFAULTCAPACITY\_EMPTY\_ELEMENTDATA) {

minCapacity = Math.max(DEFAULT\_CAPACITY, minCapacity);

}

1.2 调用ensureExplicitCapacity(minCapacity)方法

code: ensureExplicitCapacity(minCapacity);

1.2.1 修改此列表的次数+1

code: modCount++; (modCount 属性继承自AbstractList类)

1.2.2 判断容器最小容量是否大于elementData数组的长度

如果为true：

调用grow(minCapacity)方法

code: if (minCapacity - elementData.length > 0)

grow(minCapacity);

1.2.2.1 获取旧的容器容量(为elementData的长度)

code: int oldCapacity = elementData.length;

1.2.2.2 设置新的容器容量(为旧的容器容量数值加上其数值二进制左移1位的数值)

code: int newCapacity = oldCapacity + (oldCapacity >> 1);

1.2.2.3 判断新容器容量是否小于最小容器的容量

如果为true

新容器容量的值等于最小容器的容量

code: if (newCapacity - minCapacity < 0)

newCapacity = minCapacity;

1.2.2.4 判断新容器容量是否大于最大容器的容量

1.如果为true

判断容器最小容量是否小于0

如果为true

抛出OutOfMemoryError错误

code: if (minCapacity < 0) // overflow

throw new OutOfMemoryError();

2.判断最小容器容量是否大于最大容量

如果为true

设置最新容量为2147483647 (Integer.MAX\_VALUE)

如果为fals

设置最新容量为2147483639 (MAX\_ARRAY\_SIZE)

code: newCapacity=(minCapacity > MAX\_ARRAY\_SIZE) ? Integer.MAX\_VALUE : MAX\_ARRAY\_SIZE;

1.2.2.5 elementData指向一个新的elementData对象，长度为最新的容器容量，将旧elementData对象的元素复制到新的elementData中

code： elementData = Arrays.copyOf(elementData, newCapacity);

2 给elementData对象添加一个object元素，并将size加1

code: elementData[size++] = e;

3 返回true表示容器添加对象成功

code: return true;