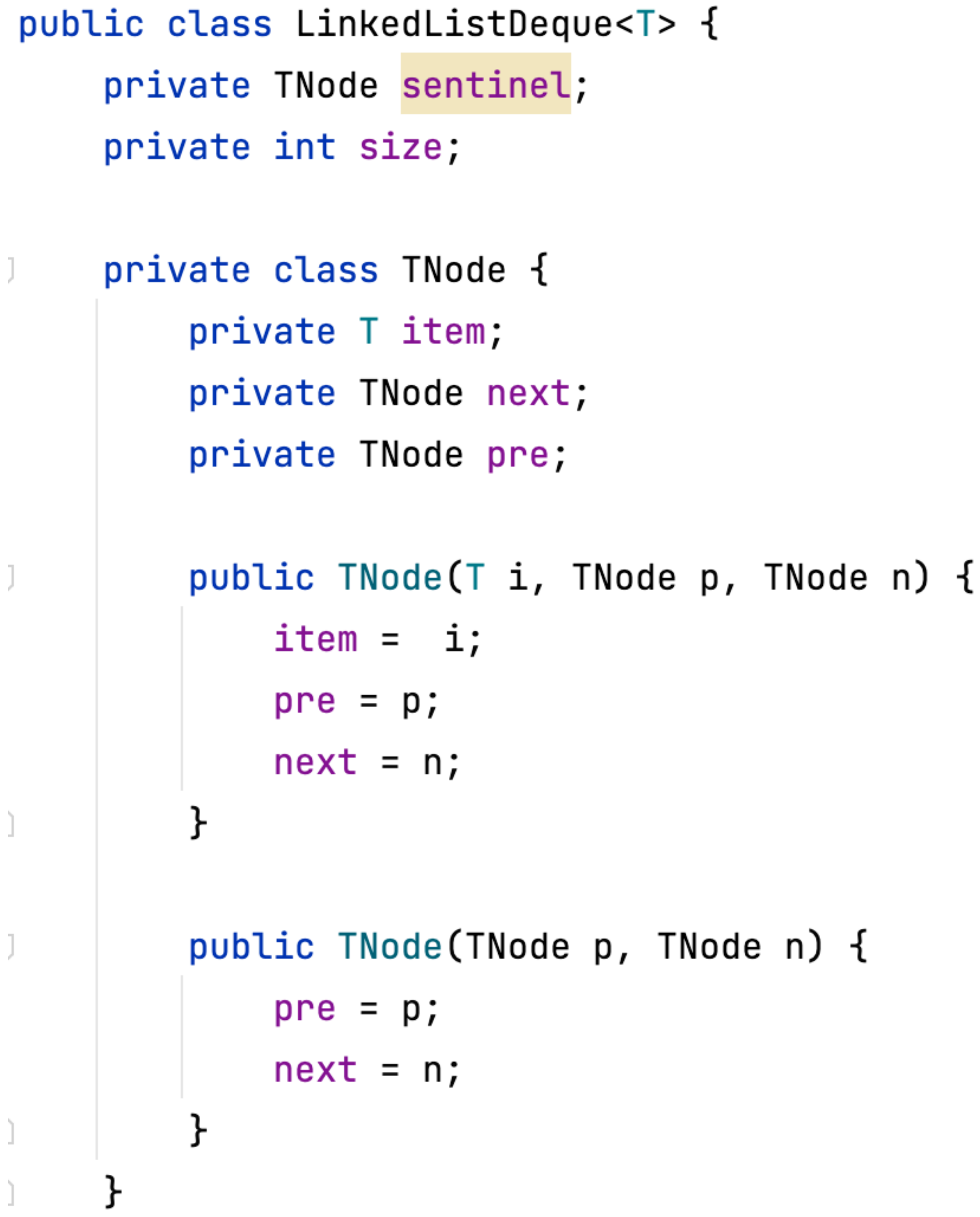
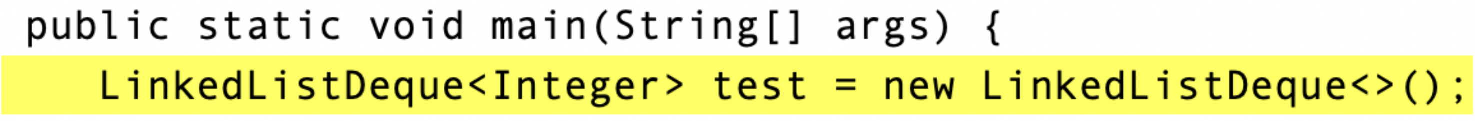
Table 1

Table 1

new, constructor, pointer, null的概念 - CS61b-prj1A - Jan26/2023



```java
public class LinkedListDeque<T> {
    private TNode sentinel;
    private int size;

    private class TNode {
        private T item;
        private TNode next;
        private TNode pre;

        public TNode(T i, TNode p, TNode n) {
            item =  i;
            pre = p;
            next = n;
        }

        public TNode(TNode p, TNode n) {
            pre = p;
            next = n;
        }
    }
```

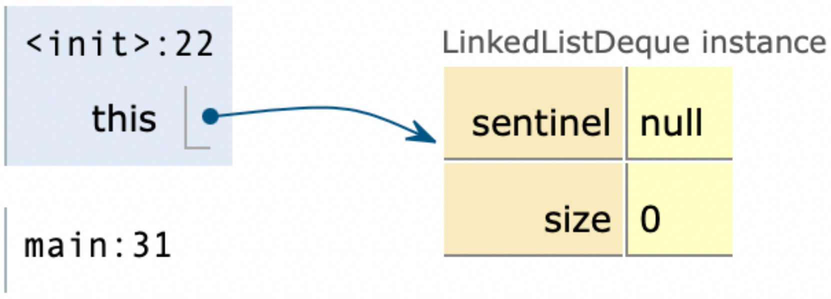


```java
public static void main(String[] args) {
    LinkedListDeque<Integer> test = new LinkedListDeque<>();
```

new 会create一个object，同时生成一个64-bits address 指向这个object；此时object配有所有的instance variable，每个variable 默认为null/0； （只要使用new，就会生成一个盒子，有可能是空盒子）

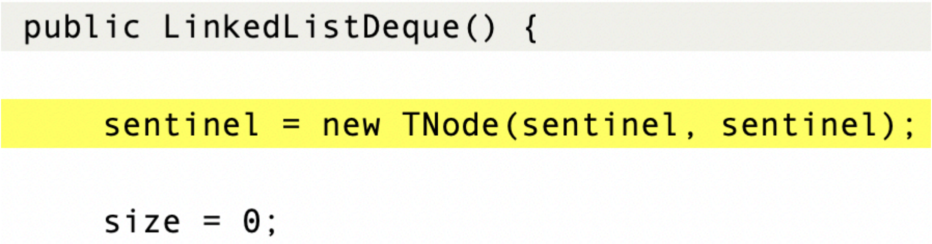
sentinel is a TNode, 所以sentinel 是一个variable，里面放置64-bits address，指向一个TNode。 （There are only nine types in java: 8 primitive types and references to anything , reference 都是64-bits address. 每个variable 必须指明type，sentinel是一个reference。 ）

参考： Java does not have uninitialized variables. Fields of classes and objects that do not have an explicit initializer and elements of arrays are automatically initialized with the default value for their type (false for boolean, 0 for all numerical types, null for all reference types). null 实际是64个0；



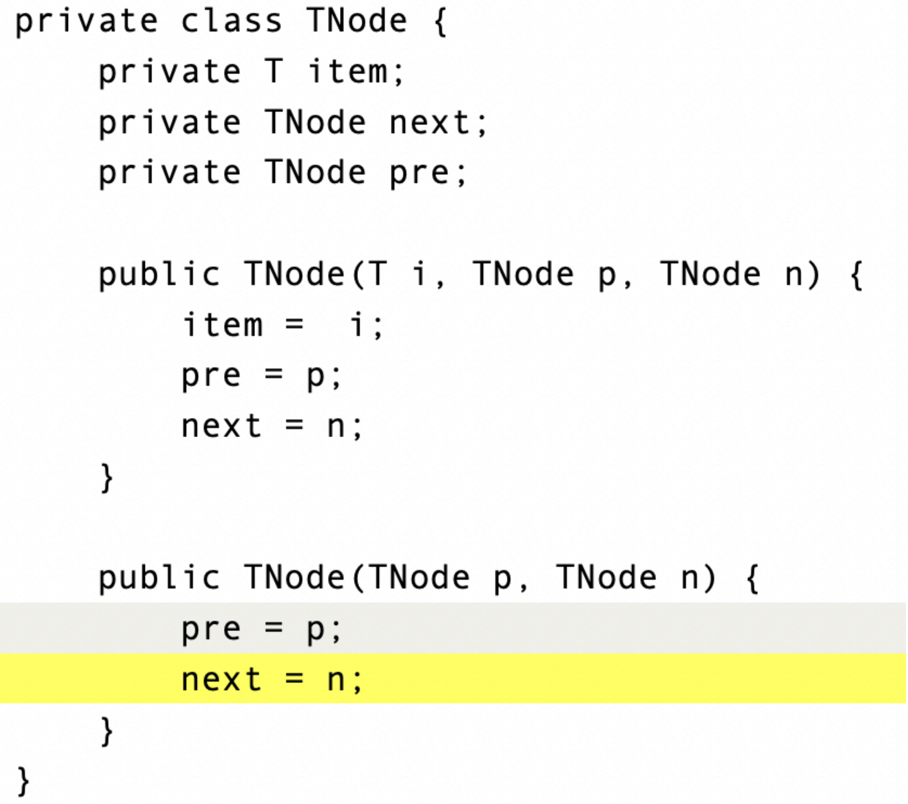
经过construtor，把这些variable 填充为初始值；

注意此时等号右边的sentinel == null

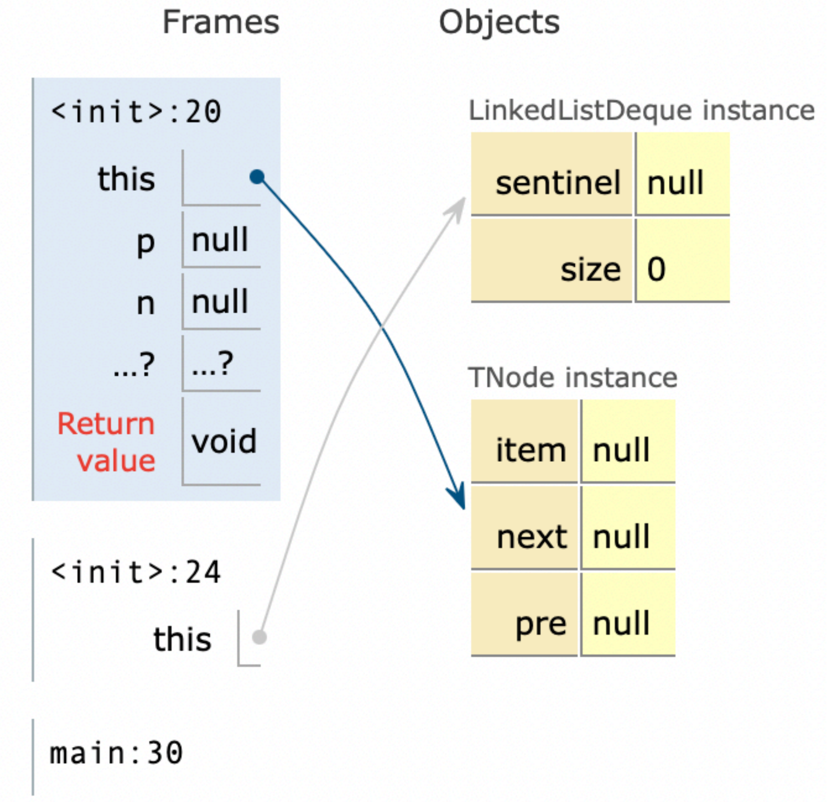


```
public LinkedListDeque() {

    sentinel = new TNode(sentinel, sentinel);

    size = 0;
```

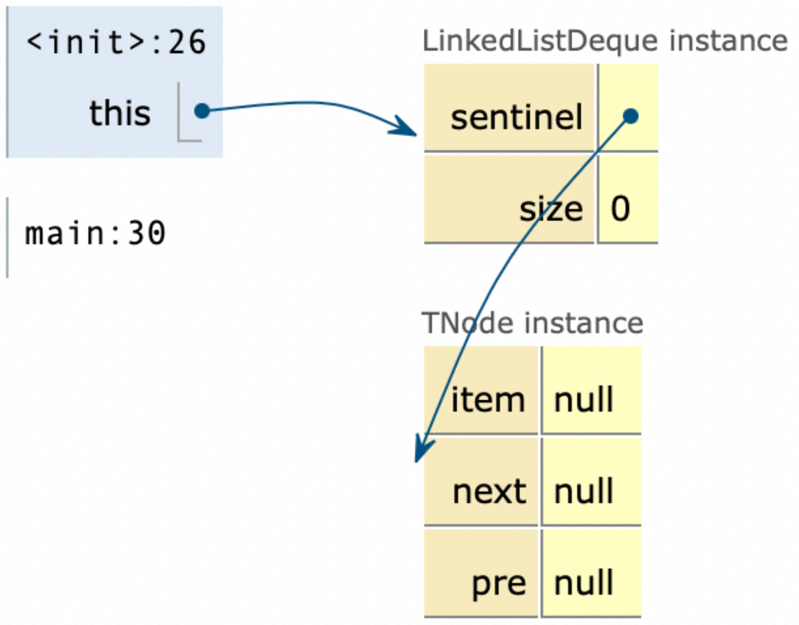
此时需要生成一个新的TNode object



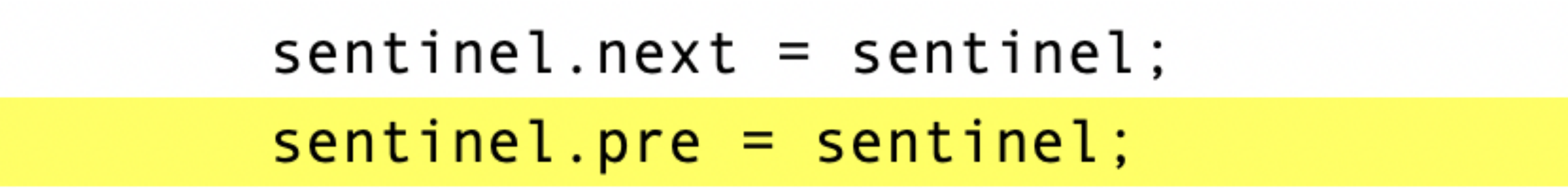
```
private class TNode {
    private T item;
    private TNode next;
    private TNode pre;

    public TNode(T i, TNode p, TNode n) {
        item =  i;
        pre = p;
        next = n;
    }

    public TNode(TNode p, TNode n) {
        pre = p;
        next = n;
    }
}
```



new返回一个64-bits address 指向一个null， 把这个地址copy 到左边的sentinel

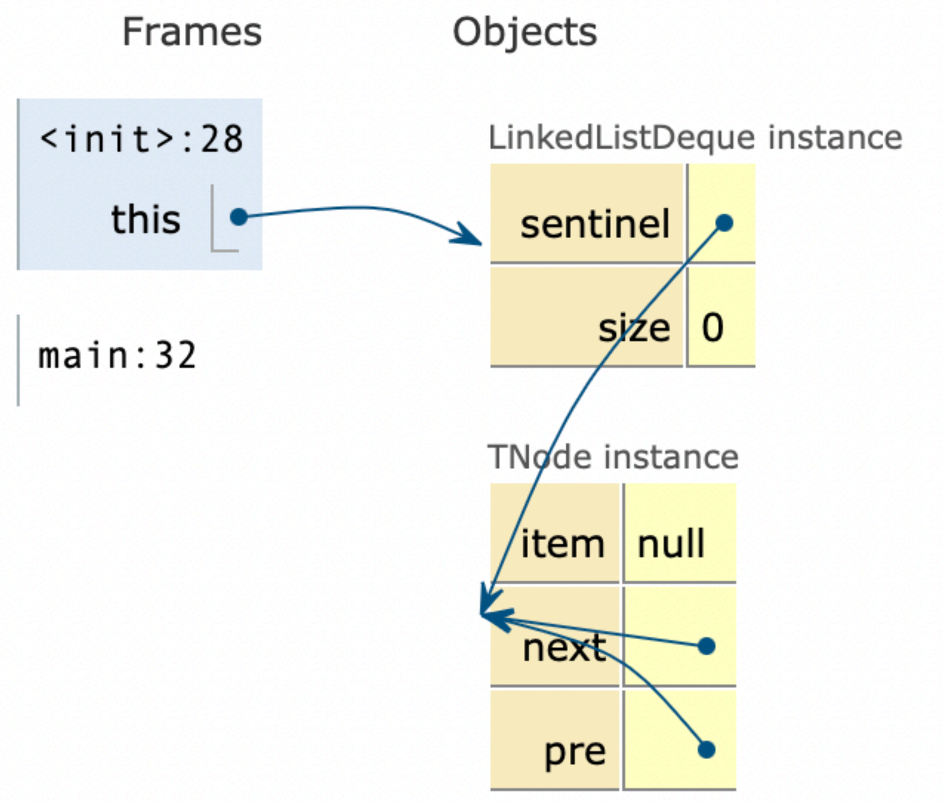


添加下面两行，注意此时sentinel 不是null了， sentinel 里面是指向TNode instance 的地址(pointer) , next 和pre 里面要存放的是地址；



```
sentinel.next = sentinel;
sentinel.pre = sentinel;
```

circular sentinel，initialize时需要sentinel 的next 和pre 都存放着sentinel自己的地址；



all the fields of the object are null, 但是这个object 本身不是null，相当于空盒子，不是null