EECS2011 Fundamentals of Data Structures (Winter 2022)

Q&A - Lectures 8

Wednesday, March 16

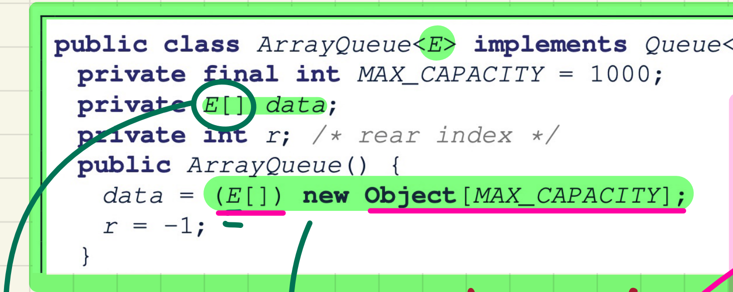
Announcements

- Lecture W9 to be released
    + Tree Traversals
    + Binary Search Trees (BST)
- Assignment 2 released
- Written Test 2 coming soon (guide released)

Hi. my question is, when exactly do we use Array.newInstance(this.getClass(), MAX_NUM_CHILDREN) and new object[x].

Or more precisely, How do we know which one to use?

thanks.

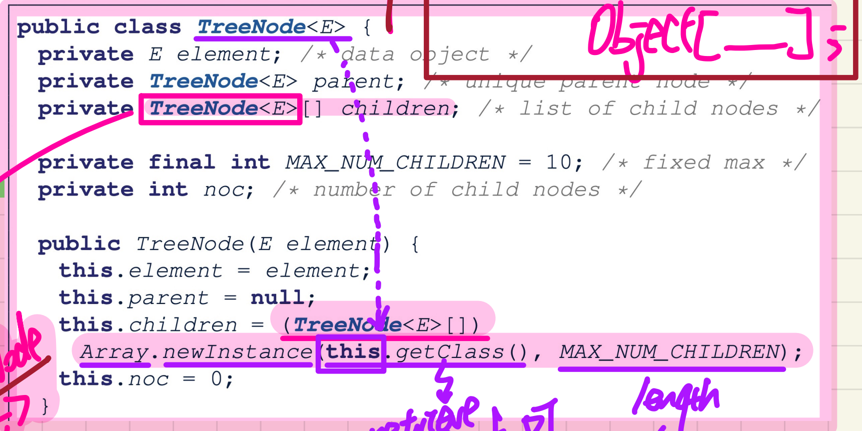


```
public class ArrayQueue<E> implements Queue<E> {
  private final int MAX_CAPACITY = 1000;
  private E[] data;
  private int r; /* rear index */
  public ArrayQueue() {
    data = (E[]) new Object[MAX_CAPACITY];
    r = -1;
  }
```

elements in array have type E

data = E[] X

data = new E[__]; X



```
public class TreeNode<E> {
  private E element; /* data object */
  private TreeNode<E> parent; /* unique parent node */
  private TreeNode<E>[] children; /* list of child nodes */

  private final int MAX_NUM_CHILDREN = 10; /* fixed max */
  private int noc; /* number of child nodes */

  public TreeNode(E element) {
    this.element = element;
    this.parent = null;
    this.children = (TreeNode<E>[])
    Array.newInstance(this.getClass(), MAX_NUM_CHILDREN);
    this.noc = 0;
  }
```

this.children = (TreeNode<E>[]) Object[__];

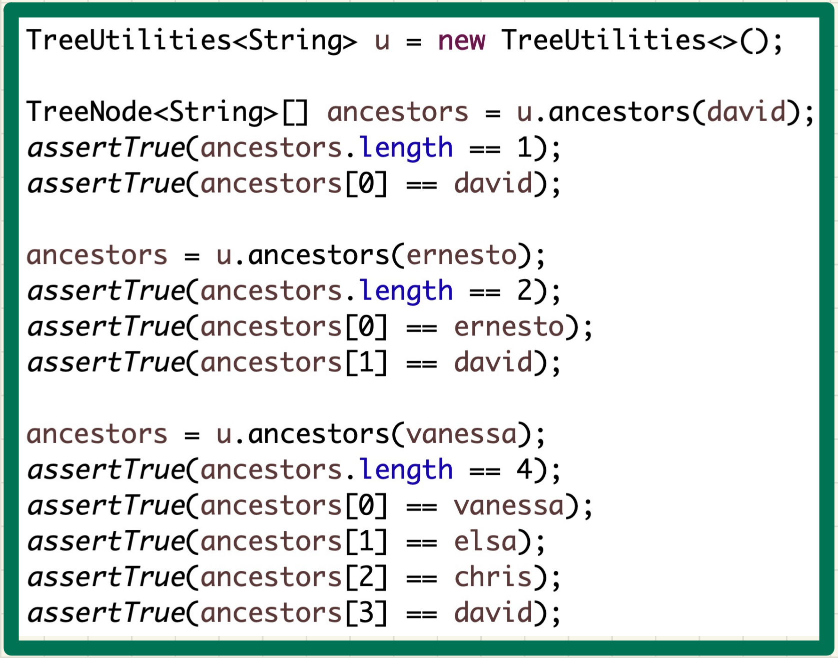
ClassCast Exception

each elem. in array has type ~~TreeNode<E>~~

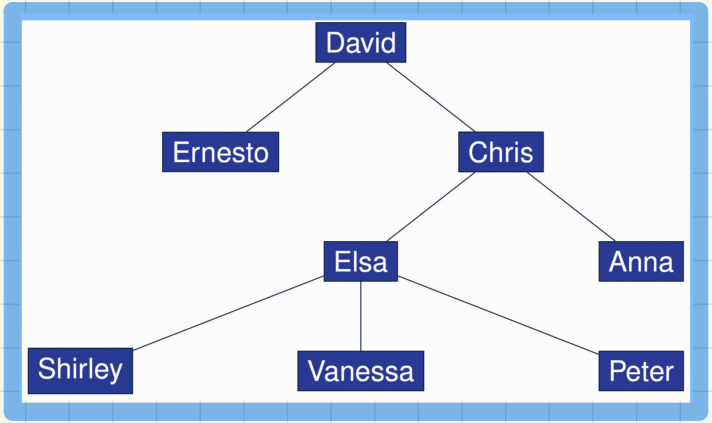
retrieve info of DT

length

Sketch: Computing a Node's Ancestors



```
TreeUtilities<String> u = new TreeUtilities<>();

TreeNode<String>[] ancestors = u.ancestors(david);
assertTrue(ancestors.length == 1);
assertTrue(ancestors[0] == david);

ancestors = u.ancestors(ernesto);
assertTrue(ancestors.length == 2);
assertTrue(ancestors[0] == ernesto);
assertTrue(ancestors[1] == david);

ancestors = u.ancestors(vanessa);
assertTrue(ancestors.length == 4);
assertTrue(ancestors[0] == vanessa);
assertTrue(ancestors[1] == elsa);
assertTrue(ancestors[2] == chris);
assertTrue(ancestors[3] == david);
```

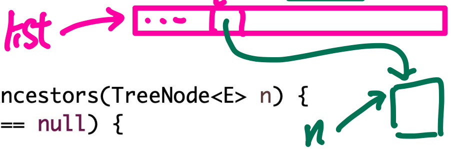


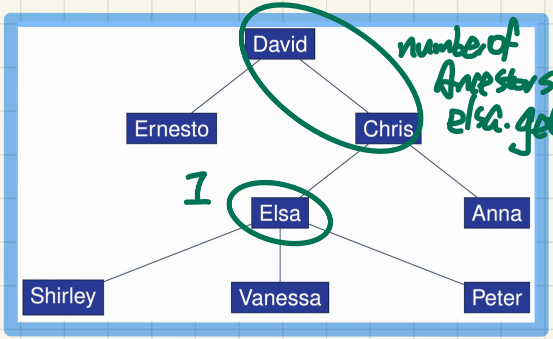
Tracing: Computing a Node's Ancestors

```java
public TreeNode<E>[] ancestors(TreeNode<E> n) {
    int size = this.numberOfAncestors(n);
    TreeNode<E>[] list = (TreeNode<E>[]) Array.newInstance(n.getClass(), size);
    this.ancestorsHelper(n, list, 0);
    return list;
}

private void ancestorsHelper(TreeNode<E> n, TreeNode<E>[] list, int i) {
    if(n != null) {
        list[i] = n;
        ancestorsHelper(n.getParent(), list, i + 1);
    }
}

private int numberOfAncestors(TreeNode<E> n) {
    if(n.getParent() == null) {
        return 1;
    }
    else {
        return 1 + numberOfAncestors(n.getParent());
    }
}
```

start index to store ancestors

this.getClass()
↳ TreeUtilities





numberOfAncestors(elsa.getP())

```java
TreeUtilities<String> u = new TreeUtilities<>();

TreeNode<String>[] ancestors = u.ancestors(david);
assertTrue(ancestors.length == 1);
assertTrue(ancestors[0] == david);

ancestors = u.ancestors(ernesto);
assertTrue(ancestors.length == 2);
assertTrue(ancestors[0] == ernesto);
assertTrue(ancestors[1] == david);

ancestors = u.ancestors(vanessa);
assertTrue(ancestors.length == 4);
assertTrue(ancestors[0] == vanessa);
assertTrue(ancestors[1] == elsa);
assertTrue(ancestors[2] == chris);
assertTrue(ancestors[3] == david);
```