

## Aufgabe 1

Ein leerer Knoten ist ein BinaryTree-Objekt, dessen Referenzen alle auf null zeigen.

Sobald man einen Inhalt einfügt, werden zwei neue leere Knoten (BinaryTree-Objekte) erzeugt und mit den Referenzen left und right verknüpft. Der Inhalt wird mit der Referenz content verknüpft.

## Aufgabe 2

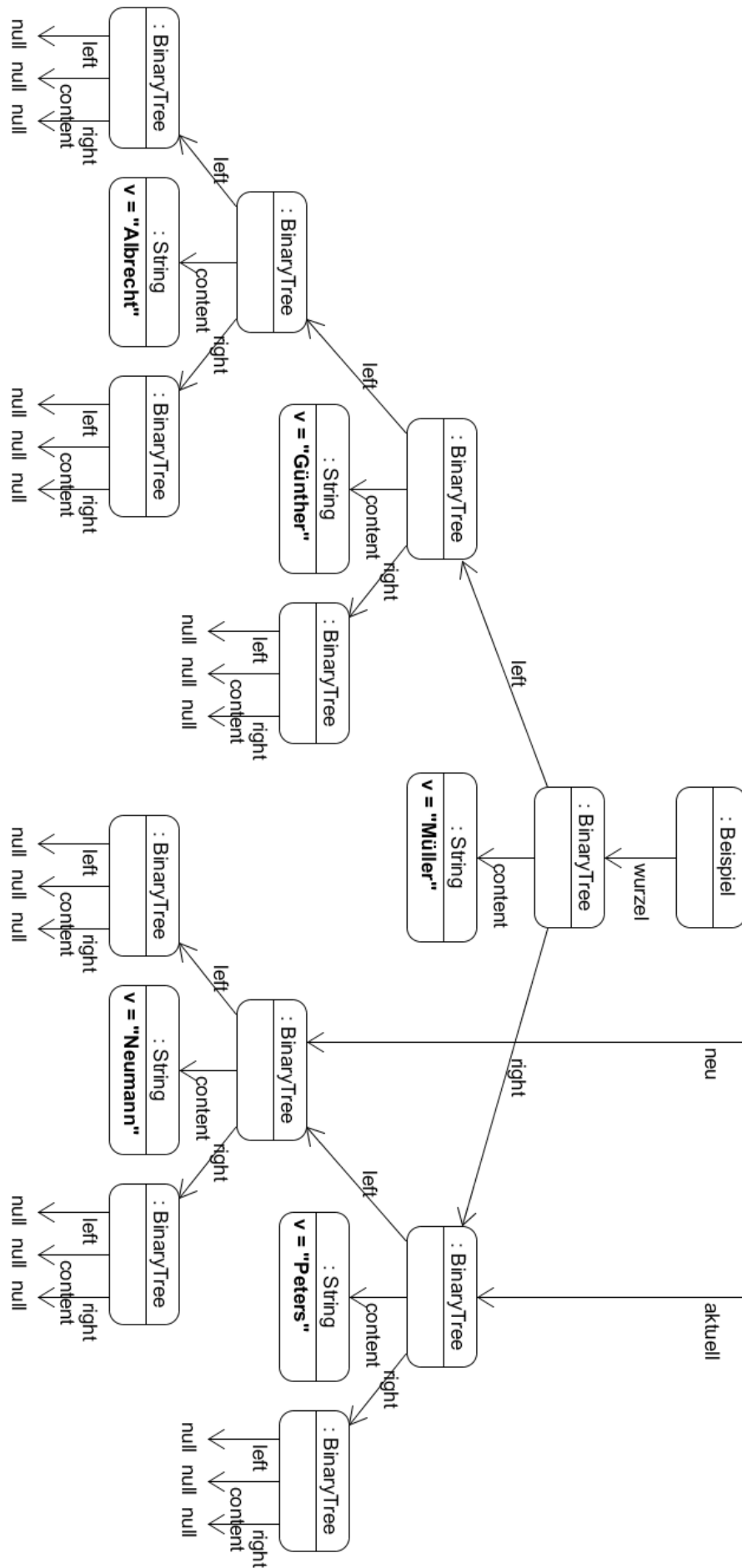
a) (Siehe nächste Seite)

b)

Die Referenz wurzel zeigt immer auf den Wurzelknoten. Die Referenz aktuell wird benutzt, um einen Knoten innerhalb des Baums auszuwählen, und dann neue Knoten daran zu hängen.

Die Referenz neu wird zur Erzeugung neuer Knoten verwendet.

Wenn die Referenz wurzel lokal deklariert wäre, würde sie am Ende der main-Methode gelöscht. Dann würde keine Referenz mehr auf das Objekt an der Wurzel des Binärbaums zeigen, und es würde vom Garbage Collector gelöscht. Damit würden dann auch alle anderen Knoten des Binärbaums samt Inhaltsobjekten nacheinander gelöscht, weil keine Referenzen mehr auf sie zeigen.



```
public void main()
{
    Kontakt k;
    BinaryTree<Kontakt> neu, aktuell;

    // Kontakt an der Wurzel einfügen
    k = new Kontakt("Müller", "", "", "");
    wurzel.setContent(k);

    // Kontakte links / rechts von der Wurzel einfügen
    k = new Kontakt("Günther", "", "", "");
    neu = new BinaryTree(k);
    wurzel.setLeftTree(neu);

    k = new Kontakt("Peters", "", "", "");
    neu = new BinaryTree(k);
    wurzel.setRightTree(neu);

    // Kontakte unter dem linken Nachfolger der Wurzel
    aktuell = wurzel.getLeftTree();
    k = new Kontakt("Albrecht", "", "", "");
    neu = new BinaryTree(k);
    aktuell.setLeftTree(neu);

    k = new Kontakt("Becker", "", "", "");
    neu = new BinaryTree(k);
    aktuell.setRightTree(neu);

    // Kontakte rechts unter Becker
    aktuell = aktuell.getRightTree();
    k = new Kontakt("Frantz", "", "", "");
    neu = new BinaryTree(k);
    aktuell.setRightTree(neu);

    aktuell = aktuell.getRightTree();
    k = new Kontakt("Engel", "", "", "");
    neu = new BinaryTree(k);
    aktuell.setLeftTree(neu);
}
```