

Aufgabe 1:

8 P.

a)

A	B	!(A && !B)
w	w	w
w	f	f
f	w	w
f	f	w

b)

 $!(A \ \&\& \ B) = !A \ || \ !B$ 

c)

1.) `kara.onLeaf() || !kara.treeFront()`2.) `!kara.treeLeft() && !kara.treeRight() && !kara.treeFront()`Aufgabe 2:

5 P.

```

if (!kara.treeFront()) {
    kara.move();
}
else {
    if (!kara.treeLeft()) {
        kara.turnLeft();
        kara.move();
    } // if //
} // else //

```

Aufgabe 3:

7 P.

```

import roboapp.javakara.JavaKaraProgram;
public class Fehlerhaft extends JavaKaraProgram {
    void umdenBaum() {
        kara.turnLeft();
        kara.move();
        kara.turnRight();
        kara.move();
        kara.move();
        kara.turnRight();
        kara.move();
        kara.turnLeft; ← Hier fehlt eine Klammer: kara.turnLeft();
    }; ← ; zuviel: }
    protected void myProgram() {
        while (!kara.onLeaf() { ← Hier fehlt Klammer zu: while (!kara.onLeaf()) {
            while (!kara.treeFront()) {
                kara.move();
            }
            this.umdeneBaumherum(); ← falscher Methodenaufruf: this.umdeneBaum();
        }
        removeLeaf(); ← kara.fehlt: kara.removeLeaf();
    } // myProgram //
} // Fehlerhaft //

```

Kara läuft um (einzelne) Bäume herum, bis er auf einem Kleeblatt steht, dann nimmt er es auf.

Aufgabe 4:

10 P.

```

import roboapp.javakara.JavaKaraProgram;
public class RechtsDanebenLegen extends JavaKaraProgram {
    void legeRechts() {
        kara.turnRight();
        kara.move();
        kara.putLeaf();
        kara.turnLeft();
        kara.turnLeft();
        kara.move();
        kara.turnRight();
    }
    protected void myProgram() {
        while (!kara.treeFront()) {
            if (kara.onLeaf()) {
                legeRechts();
            } // if //
            kara.move();
        } // kara.treeFront();
        if (kara.onLeaf()) { // wichtig, weil Kara am Ende auf einem Kleeblatt
            legeRechts(); // stehen könnte!
        } // if //
    } // myProgram //
} // RechtsDanebenLegen //

```