

Aufgabe 2)

```
/**
 * Diese Methode zeichnet einen Pythagors Baum auf ein Canvas mit verschiedenen
 * Einstellungsmoeglichkeiten.
 *
 * @param c          Das Canvas auf dem der Pythagoras Baum gezeichnet werden
 *                   soll.
 * @param level      Die Tiefe der Rekursion.
 * @param length     Die Groesse des ersten Rechtecks.
 * @param leftAngle  Die Groesse des linken Winkels.
 * @param rightAngle Die Groesse des rechten Winkels.
 * @param switchLength Die Laenge ab welcher alle groesseren Rechtecke Braun
 *                   seien sollen.
 */
private static void paintPythagorasTree(
    Canvas c,
    int level,
    double length,
    int leftAngle,
    int rightAngle,
    int switchLength) {

    // check for level
    if (level < 1) {
        return;
    }

    // set color
    if (length <= switchLength) {
        c.chooseColor(c.GREEN);
    } else {
        c.chooseColor(c.BROWN);
    }

    // draw Square
    c.square(length);

    // Right
    double rightLength = calculateLength(leftAngle, rightAngle, length);
    c.move(length / 2, -length / 2);
    c.rotate(rightAngle);
    c.move(-rightLength / 2, -rightLength / 2);

    paintPythagorasTree(c, level - 1, rightLength, leftAngle, rightAngle, switchLength);

    c.move(rightLength / 2, rightLength / 2);
    c.rotate(-rightAngle);
    c.move(-length / 2, length / 2);

    // Left
    double leftLength = calculateLength(rightAngle, leftAngle, length);
    c.move(-length / 2, -length / 2);
    c.rotate(-leftAngle);
    c.move(leftLength / 2, -leftLength / 2);

    paintPythagorasTree(c, level - 1, leftLength, leftAngle, rightAngle, switchLength);

    c.move(-leftLength / 2, leftLength / 2);
    c.rotate(leftAngle);
    c.move(length / 2, length / 2);
}

private static double calculateLength(int alpha, int beta, double length) {
    return (Math.sin(Math.toRadians(alpha)) * length) /
        (Math.sin(Math.toRadians(180 - alpha - beta)));
}
```

Aufgabe 4)

VPL Abgabe von Maximilian Petri