

| Zeile | n | color | i | x | y | Zeile 7 | Zeile 24 | function |
|-------|---|---------|---|-----|-----|----------------------------|-------------------------|-----------------------------------|
| 8 | 5 | | | | | for - Schleife I | for - Schleife II | create box |
| 9 | | | | | 50 | i < n → true | (24) Klasse "big" | (39) create div |
| 10 | | | 0 | 170 | | | (25) create box | (34) Anhängen v. div an body |
| 12 | | #ff0000 | | | | | if (n = 4) → false | (35) Klasse "big" anhängen |
| 13 | | | | | | break → for - Schleife I | (24) Klasse "medium" | (36) HG #ff0000 |
| | 5 | | | | 100 | zurück 2.8. | (25) create box | (37, 38) links: 170px, top: 50px |
| 8 | | | | | | i < n → true | if → false | (33) create div |
| 9 | | | 1 | 340 | | | | (34) Anhängen an body |
| 10 | | | | | | (24) Klasse "small" | (25) create box | (35) Klasse "medium" |
| 16 | | #00ff00 | | | | if → false | | (36) HG #ff0000 |
| 17 | | | | | | break → for - Schleife I | - for - Schleife fertig | (37, 38) links: 170px, top: 50px |
| | 5 | | | | 120 | zurück 2.8. | | |
| 8 | | | | | | i < n → true | | (33) create div |
| 9 | | | 2 | 110 | | | (24) Klasse "big" | (34) Anhängen an body |
| 10 | | | | | | for - Schleife | (25) create box | (35) Klasse "medium" "small" |
| 21 | | #0000ff | | | | zurück 2.8. | if → false | (36, 37, 38) s.o. |
| | 5 | | | | 170 | i < n → true | | |
| 8 | | | 3 | 280 | | | (24) Klasse "medium" | (33) create div |
| 9 | | | | | | for - Schleife | (25) create box | (34) Anhängen an body |
| 10 | | | | | | ausführung abgebr. | if → false | (35) Klasse "big" |
| 19 | | | | | | zurück 2.8. i < n → true | | (36) HG #00ff00 |
| | 5 | | | | 220 | | (24) Klasse "small" | (37, 38) links: 340px, top: 100px |
| 8 | | | | | | for - Schleife | (25) create box | (33, 34) s.o. |
| 9 | | | 4 | 50 | | | if → false | (35) Klasse "medium" |
| 10 | | | | | | for - Schleife | | (36, 37, 38) s.o. |
| 16 | | #00ff00 | | | | fertig | | (33, 34) s.o. |
| 17 | | | | | | | (24) Klasse "small" | (35) Klasse "small" |
| | 5 | | | | | Programme | (25) create box | (36, 37, 38) s.o. |
| 7 | | | 5 | | | ende i < n → false | if → false | (33, 34) s.o. |
| | | | | | | | (24) Klasse "big" | (35) Klasse "big" |
| | | | | | | | (25) create box | (36) HG #0000ff |
| | | | | | | | if → false | (37) links: 110px |
| | | | | | | | | (38) top: 120px |

Zeile | n | color | l | x | y | for - schleife I

for - schleife II

(24) Klasse small
(25) create box
if - false
fertig

(24) Klasse big
(25) create box
if -> true
for - schleife fertig

createBox
(33, 34) s.o.
(35) Klasse medium
(36, 37, 38) s.o.

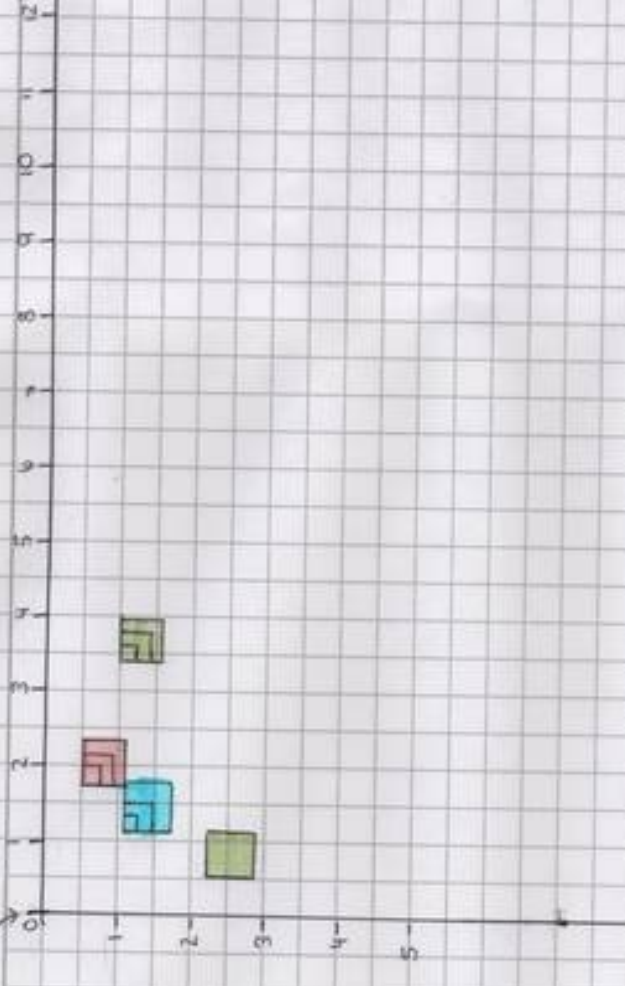
(33, 34) s.o.
(35) Klasse small
(36, 37, 38) s.o.

(33, 34) s.o.
(35) Klasse big
(36) HG + 00ff00
(37) links 50px
(38) top: 220px



linke,
obere
ecke

1cm = 100px



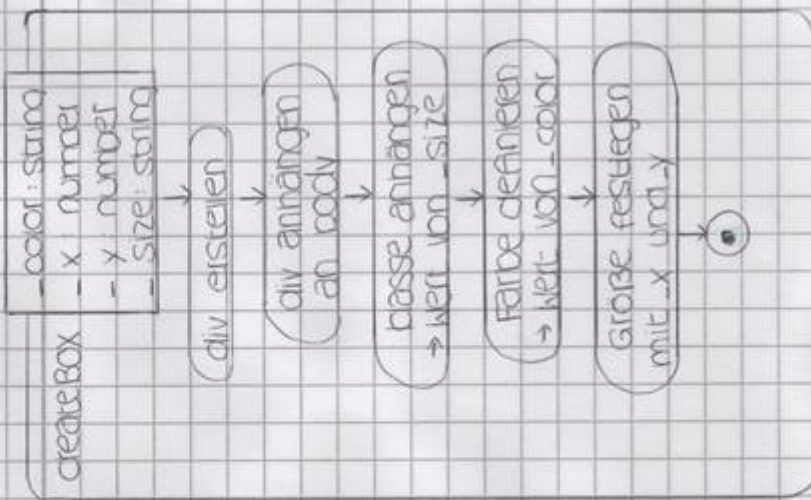
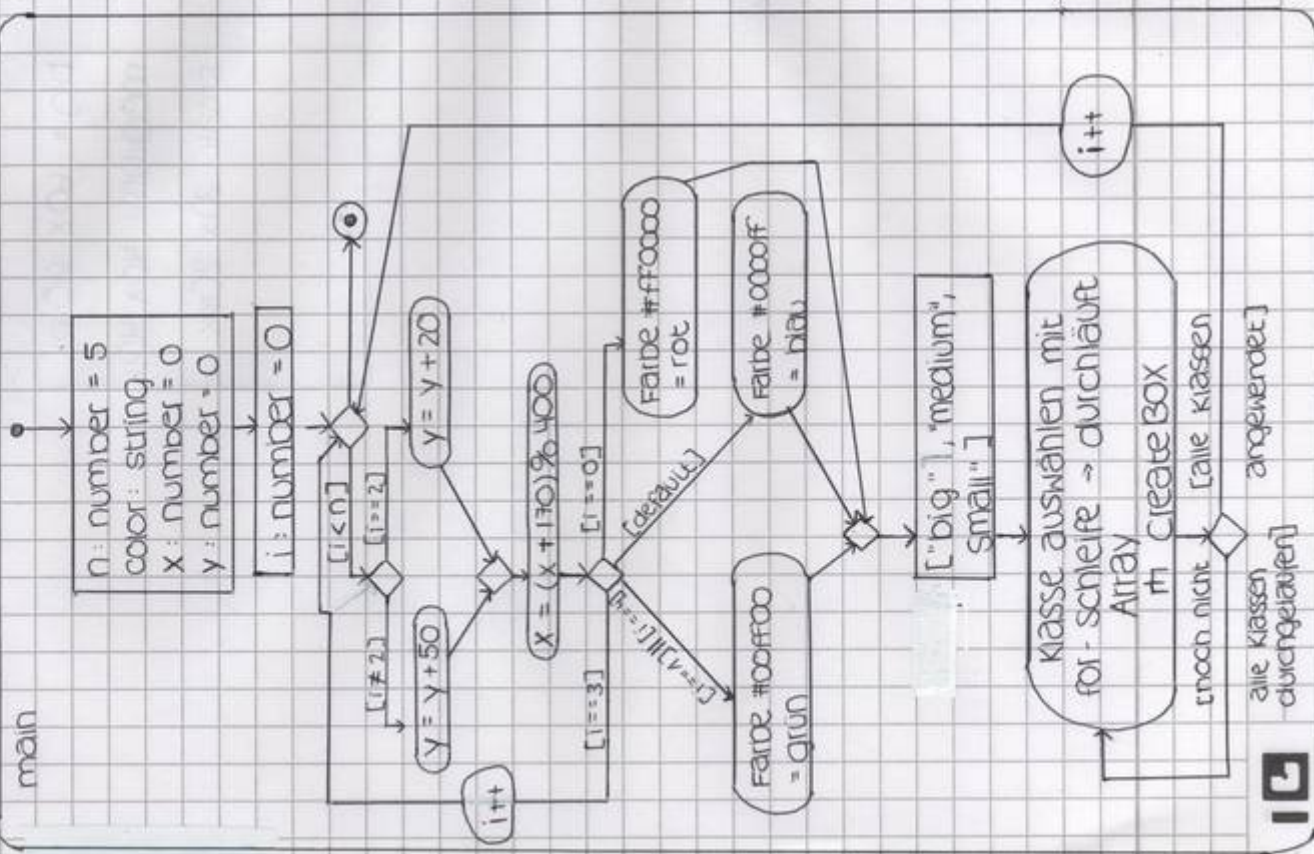
big = 60 x 60 px
medium = 40 x 40 px
small = 20 x 20 px

BOXES

→ Activity Diagram



main



2. Stelle die zu erwartende Ausgabe grafisch auf Papier dar.
3. Bringe schließlich den Code zum Laufen und überprüfe das Ergebnis.
4. Verfolge den Verlauf im Debugger. Bestimme Abweichungen zu deiner Erwartung und erkläre diese.
5. Versuche ein Aktivitätsdiagramm für den Programmablauf zu konstruieren. Nutze dafür das EIA2-Booklet sowie das Beispiel aus der ersten Aufgabe.

TypeScript

```

namespace Boxes {
    let n: number = 5;
    let color: string;
    let x: number = 0;
    let y: number = 0;

    for (let i: number = 0; i < n; i++) {
        y += (i == 2) ? 20 : 50; // y = y + 20 ODER y = y + 50
        x = (x + 170) % 400; // i = 2 i = 50
        switch (i) {
            case 0:
                color = "#ff0000";
                break;
            case 1:
            case 4:
                color = "#00ff00";
                break;
            case 3:
                continue;
            default:
                color = "#0000ff";
        }
    }

    for (let size of ["big", "medium", "small"]) {
        createBox(color, x, y, size);
        if (i == 4)
            break;
    }
}

```

Handwritten annotations in the image include arrows pointing from the switch cases to the color assignments, and a note "MODULO" next to the modulo operation in the x calculation. The loop variables i and size are also annotated with their respective values.

```

}

```

```

function createBox(_color: string, _x: number, _y: number, _size: string): void {
    let div: HTMLDivElement = document.createElement("div");
    document.body.appendChild(div);
    div.classList.add(_size);
    div.style.backgroundColor = _color;
    div.style.left = _x + "px";
    div.style.top = _y + "px";
}

```