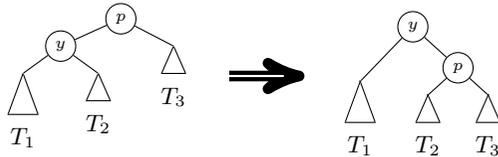


**Aufgabe 31**

(mündlich)

- (a) Implementieren Sie eine Prozedur `RightRotate(y)`, die den Teilbaum  $T(p)$  auf der linken Seite in den Teilbaum  $T(y)$  auf der rechten Seite überführt.



**Lösung:**

`RightRotate(y)`

---

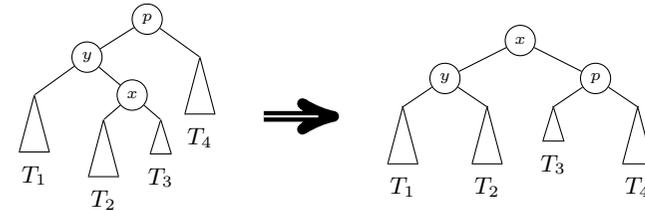
```

1  p := parent(y)
2  T1 := left(y)
3  T2 := right(y)
4  T3 := right(p)
5  // setze y an die Wurzel des Teilbaums
6  parent(y) := parent(p)
7  if parent(y) = nil then
8    root := y
9  else if left(parent(y)) = p then
10   left(parent(y)) := y
11  else
12   right(parent(y)) := y
13  // setze p als rechtes Kind von y
14  right(y) := p
15  parent(p) := y
16  // setze T2 als rechten Teilbaum von y
17  left(p) := T2
18  if T2 ≠ nil then parent(T2) := p
19  // aktualisiere die Balancen
20  bal(y) := 0
21  bal(p) := 0

```

---

- (b) Implementieren Sie eine Prozedur `LeftRightRotate(y)`, die den Teilbaum  $T(p)$  auf der linken Seite in den Teilbaum  $T(x)$  auf der rechten Seite überführt.



**Lösung:**

`LeftRightRotate(y)`

---

```

1  p := parent(y)
2  x := right(y)
3  T1 := left(y)
4  T2 := left(x)
5  T3 := right(x)
6  T4 := right(p)
7  // setze x an die Wurzel des Teilbaums
8  parent(x) := parent(p)
9  if parent(x) = nil then
10   self.root := x
11  else if left(parent(x)) = p then
12   left(parent(x)) := x
13  else
14   right(parent(x)) := x
15  // setze y als linkes Kind von x
16  left(x) := y
17  parent(y) := x
18  // setze p als rechtes Kind von x
19  right(x) := p
20  parent(p) := x
21  // setze T2 als rechten Teilbaum von y
22  right(y) := T2
23  if T2 ≠ nil then parent(T2) := y
24  // setze T3 als linken Teilbaum von p
25  left(p) := T3
26  if T3 ≠ nil then parent(T3) := p
27  // aktualisiere die Balancen
28  if bal(x) = 1 then bal(p) := -1 else bal(p) := 0
29  if bal(x) = -1 then bal(y) := 1 else bal(y) := 0
30  bal(x) := 0

```

---