Aufgabe 5.1

(1) Aufgabe

```
1  iff r = lcm(n, m) => r = (n * m) / gcd(n, m)
```

(2) Aufgabe

```
1  var n, m, r : Integer
2  pre n ≥ 0 ∧ m ≥ 0
3  post r = lcm(n, m)
4  reads −
5  changes n, m, r
6  mem −
```

(3) Aufgabe

```
1  module lcm(n, m : Integer)
2  var r : Integer;
3  r := (n * m) / gcd(n, m)
4  endmodule
```

(4) Aufgabe

```
1  public int gcd(int x, int y) {
2    int tmp;
3      while (y != 0) {
4        tmp = x % y;
5          x = y;
6          y = tmp;
7      }
8      return x;
9  }
10
11 public int lcm(int n, int m) {
12   return (n * m) / gcd(n , m);
13 }
```

Aufgabe 5.2

(1) Create(-2, 2)

(2) Put(Create(-2, 2), 0, false)
Put(Put(Create(-2, 2), 0, false), -2, true)
Put(Put(Put(Create(-2, 2), 0, false), -2, true), 2, true)

(3) Get(Put(Put(Put(Create(-2, 2), 0, false), -2, true), 2, true), -2)

(4) Lower(Put(Put(Put(Create(-2, 2), 0, false), -2, true), 2, true))

Aufgabe 5.3

(1) halve (division by 2)

Functions:

halve: Nat $\times$ Nat $\rightarrow$ Nat

Preconditions:

$\forall$i : Nat $\bullet$ pre(halve(i)) $\iff$ $\neg$ less(zero, i)

Axiom:

halve(zero) = zero

halve(succ(zero)) = zero

halve(i) = succ(halve(i-2))

(2) odd (predicate for imparity)

Functions:

Halve: odd $\rightarrow$ Bool

Preconditions:

-

Axiom:

odd(zero) = false

odd(succ(zero)) = true