Aufgabe 6.1

1 q:= *insert*(2,*insert*(4,*new*))
2 *min*(*insert*(4,*new*)) = 4
3 2<4 = *delete_min*(*insert*(2,*insert*(4,*new*)))=*insert*(4,*new*)
4 *delete_min*(*insert*(4,*new*)) = *new*

Aufgabe 6.2

(1) Type Option[T]

(2) **Functions:**

none: Option[T] → zero

some(t): Option[T] → t

the: Option[T] → t

(3) **Preconditions:** -

(4) **Axioms:**

t : T, t •

none(option(t)) = zero

some(t)(option(t)) = t

this(some(t)(option(y))) = t

Aufgabe 6.3

```java
// Nat.java
public interface Nat<T> {
    Nat<T> succ();
    Nat<T> pred();
    boolean less(Nat<T> x);
    Nat<T> add(Nat<T> x);
    Nat<T> mult(Nat<T> x);
}
```

```java
// MyNat.java
class MyNat<T> implements Nat<T> {
    public int x;
    MyNat(int x) {
        this.x = x;
    }
    public MyNat<T> succ() {
        return new MyNat(++this.x);
    }
    public MyNat<T> pred() {
        int z = this.x - 1;
        if (z >= 0) {
            return new MyNat(z);
        } else {
            return new MyNat(0);
        }
    }
    public boolean less(Nat<T> x) {
        MyNat<T> t = (MyNat<T>) x;
        return this.x < t.x;
    }
    public MyNat<T> add(Nat<T> x){
        MyNat<T> t = (MyNat<T>) x;
        return new MyNat(this.x + t.x);
    }
    public MyNat<T> mult(Nat<T> x){
        MyNat<T> t = (MyNat<T>) x;
        return new MyNat(this.x * t.x);
    }
    public String toString() {
        return "x: "+this.x;
    }
}
```