```
--Thomas Alessandro Buse 192959 Gruppe 17 Übung 12
--Paul Rüssmann 196683
import Examples
import Coalg
import Data.Array

--Aufgabe 12.1
p1 = Point 5.0 8.0

--a)
getX :: State Point Float
getX = State $ \(Point x y) -> (x,(Point x y))

getY :: State Point Float
getY = State $ \(Point x y) -> (y,(Point x y))


--b)
setX :: Float -> State Point ()
setX f = State $ \(Point x y) -> ((), (Point f y))

setY :: Float -> State Point ()
setY f = State $ \(Point x y) -> ((), (Point x f))

--Aufgabe 12.3

type ID = Int
type Bank = Array ID Account
data Account = Account { balance :: Int, owner :: Client } deriving Show
data Client = Client
        { name :: String
        , surname :: String
```

```
        , address :: String
        } deriving Show

credit :: Int -> ID -> Bank -> Bank
credit amount id ls = (//) ls [(id, (newacc oldacc))] where
        oldacc = (!) ls id
        newacc (Account oldB oldO) = Account{balance = (oldB + amount), owner = oldO}

--Aufgabe 12.4

bincoeff :: (Int,Int) -> Int
bincoeff (n,k)
        | k == 0 ||k == n = 1
        | 0 < k, k < n = bincoeff(n-1,k-1) + bincoeff(n-1,k)
        | otherwise = 0

bincoeffDyn :: (Int,Int) -> Int
bincoeffDyn (n,k) = (!) (array ((0,0),(100,100)) binco) (n,k) where
                                        binco
                                                | k == 0 || k == n = [((n,k),1)]
                                                | 0 < k, k < n = [((n,k),(bincoeffDyn(n-1,k-1) +
bincoeffDyn(n-1,k)))]
                                                | otherwise = [((n,k),0)]
```