```
1
      -- UebungSechs
 2
      -- Lilli Schuckert und Charlotte Seehagen
 3
4
      equalSums :: Int -> Int -> Int -> Bool
 5
      equalSums a b c d
 6
         |a + b == c + d = True -- Fallunterscheidung für mögliche Summenpaare aus 4
 7
                                -- Integer Eingaben. Falls die Summe der 2 Paare,
 8
                                -- welche jeweils aus der Addition von 2 Integern
9
                                -- bestehen, gleich sind, dann ist der
10
                                -- Ausgabeparameter True. Die Integer kommen ohne
                                -- Widerholung vor, weshalb wir 6 verschidene
11
12
                                -- Summenpaare haben und 3 Fälle um zu untersuchen, ob
13
                                -- die Summe der Summenpaare gleich ist.
14
        a + c == b + d = True
15
         |a + d == b + c = True
                                -- Ist dies nicht der Fall, ist der Ausgabeparameter
         |otherwise = False
16
17
                                -- False.
18
19
      numberOfPwd :: Int -> Int -> Int -> Int
20
      numberOfPwd a b c
21
         |a == b && b == c = 0 -- Sind die 3 Eingabeparametere gleich, dann gibt es
                                -- keine paarweise unterschiedliche Werte, weshalb der
22
23
                                -- Ausgabeparameter 0 ist.
         |a /= b && b /= c = 3 -- Sind alle 3 Eingabeparameter unterschiedlich, dann
24
25
                                -- ist der Ausgabeparameter 3, da es 3
26
                                -- paarweise unterschiedliche Werte gibt.
         otherwise = 2
                                -- Der dritte Fall, beschreibt die Möglichkeit, dass 2
27
28
                                -- der Eingabeparameter gleich sind und der dritte
                                -- einen anderen Wert hat. Es gibt daher 2 paarweise
29
                                -- verschiedene Werte und der Ausgabeparameter ist in
30
31
                                -- dem Fall 2.
32
33
      width :: Float -> Float -> Float -> Float
34
      width a b c
35
         |a >= b && b >= c = a - c -- Man muss bei der Funktion zwischen 6 Fällen
                                 -- unterscheiden. In diesem Fall ist der
36
37
                                 -- Eingabeparameter a größer als b und b ist größer
                                 -- als c. Um die Differenz des größten und kleinsten
38
39
                                 -- Wertes auszugeben, muss man nun c von a
                                 -- subtrahieren.
40
        |a\rangle = c \& c\rangle = b = a - b - In diesem Fall ist a größer als c und c ist größer
41
                                 -- als b, weshalb man b von a subtrahiert.
42
        |b >= a && a >= c = b - c -- In diesem Fall ist b größer als a und a ist größer
43
                                 -- als c, we shalb man c von b subtrahiert.
44
         |b >= c && c >= a = b - a -- In diesem Fall ist b größer als c und c ist größer
45
46
                                 -- als a, we shalb man a von b subtrahiert.
        c >= b && b >= a = c - a -- In diesem Fall ist c größer als b und b ist größer
47
48
                                 -- als a, weshalb man a von c subtrahiert.
         |c >= a && a >= b = c - b -- In diesem Fall ist c größer als a und a ist größer
49
50
                                 -- als b, weshalb man b von c subtrahiert.
51
52
      missing :: Int -> Int -> Int -> Int
53
      missing a b c
         | a /= b && b /= c = set1234 a b c
54
                                                     -- Die Eingabeparameter a,b und c
55
                                                     -- sind verschieden und aus der
56
                                                      -- Menge {1,2,3,4}.
57
           where
                                                      -- Ich definiere hier eine Hifs-
                                                      -- funktion.
58
```

```
59
                set1234 :: Int -> Int -> Int -> Int
60
                set1234 a b c
                    | a + b + c == 6 = 4
                                                      -- Sind a,b und c gleich 1, 2 und
61
                                                      -- 3 und addiert man diese
62
                                                      -- Integer erhält man 6 und da die
63
                                                      -- Addition von 1, 2 und 3 die
64
65
                                                      -- einzige Möglichkeit ist aus 3
                                                      -- Werten aus der Menge die Zahl 6
66
67
                                                      -- zu erhalten, kann man daraus
68
                                                      -- schließen, dass der fehlende
                                                      -- Wert, der ausgegeben werden
69
70
                                                      -- soll 4 ist.
                                                      -- Sind a, b und c gleich 1,
                    | a + b + c == 7 = 3
 71
72
                                                      -- 2 und 4 erhält man bei der
                                                      -- Addition der 3 Zahlen 7 und
 73
74
                                                      -- kann daraus schließen, dass der
75
                                                      -- fehlende Wert 3 ist.
 76
                    a + b + c == 8 = 2
                                                      -- Sind a, b und c gleich 1,
                                                      -- 3 und 4 erhält man bei der
77
 78
                                                      -- Addition der 3 Integer 8 und
79
                                                      -- kann daraus schließen, dass
                                                      -- der fehlende Wert 2 ist.
80
81
                    | a + b + c == 9 = 1
                                                      -- Sind a, b und c gleich 2, 3
                                                      -- und 4 erhält man bei der
82
83
                                                      -- Addition der 3 Integer 9 und
                                                      -- kann daraus schließen, dass
84
                                                      -- der fehlende Wert 1 ist.
85
                    otherwise = 0
                                                      -- Trifft keiner der 4 Fälle zu,
86
87
                                                      -- hat die Funktion den Ausgabe-
88
                                                      -- parameter 0.
89
       valueAt :: Float -> Float -> Float
90
       valueAt x y z = (y / (-x)) * z + y -- Die allgemeine Gleichung einer Gerade ist
91
92
                                            -- y = m*x + n. Die Steigung m errechnen wir
93
                                            -- indem wir Die Differenz von y2 und y1
94
                                            -- ausrechnen und diese Differenz durch die
                                            -- Differenz von x2 und x1 teilen. Y2 ist
95
                                            -- ein Eingabeparameter, Y1 ist 0, da ein
96
97
                                            -- Schnitt mit der X-Achse vorliegt. X2 ist
98
                                            -- 0, da die X-Achse geschnitten wird und X1
                                            -- ist ein Eingabeparameter. n ist in
99
                                            -- diesem Fall Y2, da n den Schnitt mit der
100
                                            -- Y-Achse darstellt. Gibt man nun einen
101
                                            -- weiteren Wert x ein, erhält man den Wert
102
103
                                            -- y, so dass der Punkt (x,y) auf g liegt.
104
105
       testParallel :: Float -> Float -> Float -> Float -> Bool
106
       testParallel x y a b
         (y / (-x)) == (b / (-a)) = True -- Damit eine 2 Geraden Parallel zueinander
107
108
                                           -- sind, müssen die Steigungen gleich sein.
109
                                           -- Die Berechnung der Steigung ist bereits
                                           -- in 2a) beschrieben worden. Sind die
110
111
                                           -- Steigung der Geraden g und h gleich, dann
112
                                           -- ist der Ausgabeparameter True.
         otherwise = False
113
                                           -- Sind die Steigungen unterschiedlich, ist
114
                                           -- der Ausgabeparameter False.
115
```

parallelThroughX :: Float -> Float -> Float

116

```
parallelThroughX x y a = (-a) * (y / (-x)) -- Damit die Geraden parallel zu-
117
118
                                                   -- einander sind, müssen beide
119
                                                   -- Steigungen wieder gleich sein.
                                                   -- Aus den Eigabeparametern
120
121
                                                   -- errechnet man die Steigung für
122
                                                   -- die Gerade g. Des Weiteren haben
                                                   -- wir noch den Eingabeparamter a,
123
124
                                                   -- welcher die Differenz zwischen
                                                   -- X2 und X1 darstellt. Aus den 3
125
                                                   -- Eingabeparametern soll die
126
127
                                                   -- Funktion den Wert y so bestimmen,
                                                   -- dass die Steigungen beider Geraden
128
129
                                                   -- gleich sind. Man stellt die
130
                                                   -- Gleichung nun so um, dass die
                                                   -- Funktion parallelThroughX den
131
132
                                                   -- Wert y berechnet.
133
134
       crossingAt :: Float -> Float -> Float -> Float -> Float
135
       crossingAt xg yg xh yh
136
         yg / (-xg) /= yh / (-xh) = (yh - yg) / (yg/(-xg) - yh/(-xh)) -- Die Steigung
137
                                                            -- mg und mh sollen nicht
138
                                                            -- gleich sein. Man setzt
                                                            -- beide Geradengleichungen
139
                                                            -- gleich, und stellt die
140
141
                                                            -- Gleichung nach dem
142
                                                            -- gesuchten x-Wert um, der
143
                                                            -- dann die x-Koordinate
144
                                                            -- ausgibt.
         | otherwise = error "error"
                                                            -- Sind die Steigungen
145
                                                            -- gleich, gibt die Funktion
146
147
                                                            -- error aus.
148
```