

Aufgabe T5.1: Code verstehen

(1+1=2 Punkte)

Sie finden online einen fertigen Programmcode, der wohl etwas hastig geschrieben wurde. Die sehr ineffizient gestaltete Funktion wurde wie folgt implementiert:

```
checkerFunc :: Float -> Float -> Bool
checkerFunc n m = if (n < m) then
    if (n * m) > 42
        then True
    else if (n * m) == 42
        then True
    else
        False
else if (n == m) then
    if (n * m) > 42
        then True
    else if (n * m) == 42
        then True
    else
        False
else
    if (n * m) < 42
        then False
    else if (n * m) == 42
        then False
    else
        False
```

- (a) Fassen Sie zusammen, welche Eigenschaften von n und m erfüllt sein müssen, damit die Funktion True zurückgibt.
- (b) Die Funktion lässt sich in einem einzigen booleschen Ausdruck darstellen. Geben Sie die entsprechende Funktion als Haskellcode (schriftlich) an.

Informatik I - Gruppe 1 - Übungsblatt 5

Ausgabe: 14.11.2022

Abgabe: 21.11.1022

Tutor: Tim Völker

Ali Kurt 528961

Thomas Kujawa 463620

Felix Hoff 366927

Aufgabe T5.2: Abgeleitete Klassen

(2 Punkte)

In der Vorlesung haben sie Standard-Typklassen wie Ord, Enum oder Num kennengelernt. Im folgenden Programmcode wird ein neuer Datentyp Season definiert:

```
data Season = Spring | Summer | Autumn | Winter

-- Testing if a season is cold
isCold :: Season -> Bool
isCold s
    | s == Winter = True

    | otherwise = False

-- Is there sill time until it is autumn?
beforeAutumn :: Season -> Bool
beforeAutumn s
    | s < Autumn = True
    | otherwise = False

-- Print first season
printFirstSeason :: Season
printFirstSeason = toEnum 0
```

Ausgehend von den gegebenen Funktionen, überprüfen Sie, ob für die neue Klasse alle notwendigen Standard-Typklassen in Form einer abgeleiteten Instanzdeklaration angegeben worden sind. Falls nicht, fügen Sie diese noch hinzu und begründen Sie Ihre Antwort.

Hinweis: Verwenden Sie nur die in aus der Vorlesung bekannten Standard-Typklassen.

Informatik I - Gruppe 1 - Übungsblatt 5

Ausgabe: 14.11.2022

Abgabe: 21.11.1022

Tutor: Tim Völker

Ali Kurt 528961

Thomas Kujawa 463620

Felix Hoff 366927

Aufgabe P5.3: Klassen (1 + 2 + 5 = 8 Punkte)

In der Vorlesung wurden die Datentypen Point und Vector umgesetzt. Nun soll eine Klasse Polygon implementiert werden. Dabei ist ein Polygon eine Menge von Punkten. Die Klasse Polygon definiert die Funktionen:

```
area      :: Polygon p => p -> Float
translate_poly :: Polygon p => p -> Vector -> p
scale_poly  :: Polygon p => p -> Float -> p
```

Dabei nimmt die Funktion area ein Polygon entgegen und gibt die Fläche des Polygons zurück. Die Funktion translate verschiebt jeden Punkt eines Polygons. Die Verschiebung wird durch einen Vektor angegeben. Die Funktion gibt danach das verschobene Polygon zurück. Die Funktion scale multipliziert jeden Punkt eines Polygons mit einer Gleitkommazahl s und gibt das skalierte Polygon zurück. Die Ergebnisse aller Funktionen sollen in der Konsole angezeigt werden können. Bearbeiten Sie nun folgende Aufgaben:

- (a) Implementieren Sie die Datentypen Point und Vector, die Sie in der Vorlesung kennengelernt haben. Dabei sollen die beiden Datentypen keine alternative Schreibweise für Tupel sein, sondern als eigene Datentypen definiert werden. Implementieren Sie die Funktionen translate und scale mit folgender Signatur:

translate :: Point → Vector → Point

scale :: Point → Float → Point

Dabei verschiebt die Funktion translate einen Punkt um einen Vektor und die Funktion scale multipliziert einen Punkt komponentenweise mit einer Gleitkommazahl.

```
1 {-
2   Rundet eine Gleitkommazahl n auf die k-te Nachkommastelle.
3   Beispiel: n = 3.4343232, k = 4 sollte 3.4343 ergeben.
4   Beispiel: n = 8.9385723, k = 2 sollte 8.94 ergeben.
5   Beispiel: n = 2.9238477, k = 6 sollte 2.923848 ergeben.
6 -}
7 roundDecimalPlace :: Double -> Integer -> Double
8 roundDecimalPlace n k = fromIntegral(round (n*10^k))/ fromIntegral(10^k)
```

Informatik I - Gruppe 1 - Übungsblatt 5

Ausgabe: 14.11.2022

Abgabe: 21.11.1022

Tutor: Tim Völker

Ali Kurt 528961

Thomas Kujawa 463620

Felix Hoff 366927

-
- (b) Definieren Sie die Klasse Polygon und die Datentypen Triangle (allgemeines Dreieck) und Quad (allgemeines Viereck).

```
1 {-  
2   Rundet eine Gleitkommazahl n auf die k-te Nachkommastelle.  
3   Beispiel: n = 3.4343232, k = 4 sollte 3.4343 ergeben.  
4   Beispiel: n = 8.9385723, k = 2 sollte 8.94 ergeben.  
5   Beispiel: n = 2.9238477, k = 6 sollte 2.923848 ergeben.  
6 -}  
7 roundDecimalPlace :: Double -> Integer -> Double  
8 roundDecimalPlace n k = fromIntegral(round (n*10^k))/ fromIntegral(10^k)
```

Informatik I - Gruppe 1 - Übungsblatt 5

Ausgabe: 14.11.2022

Abgabe: 21.11.1022

Tutor: Tim Völker

Ali Kurt 528961

Thomas Kujawa 463620

Felix Hoff 366927

-
- (c) Implementieren Sie die Datentypen Triangle und Quad als Instanzen von der Klasse Polygon und implementieren Sie dabei die von der Klasse Polygon definierten Funktionen.

Hinweis: Falls notwendig, recherchieren Sie die Formeln für die benötigten Flächenberechnungen.

```
1 {-  
2   Rundet eine Gleitkommazahl n auf die k-te Nachkommastelle.  
3   Beispiel: n = 3.4343232, k = 4 sollte 3.4343 ergeben.  
4   Beispiel: n = 8.9385723, k = 2 sollte 8.94 ergeben.  
5   Beispiel: n = 2.9238477, k = 6 sollte 2.923848 ergeben.  
6 -}  
7 roundDecimalPlace :: Double -> Integer -> Double  
8 roundDecimalPlace n k = fromIntegral(round (n*10^k))/ fromIntegral(10^k)
```

Informatik I - Gruppe 1 - Übungsblatt 5

Ausgabe: 14.11.2022

Abgabe: 21.11.1022

Tutor: Tim Völker

Ali Kurt 528961

Thomas Kujawa 463620

Felix Hoff 366927

Aufgabe P5.4: Aufzählung (1 + 2 + 3 = 6 Punkte)

Es soll eine Klasse Card für Spielkarten implementiert werden. Eine Spielkarte besteht aus einem Rank (Wert) {Seven, Eight, Nine, Ten, Jack, Queen, King, Ace} in aufsteigender Reihenfolge und einem Suit (Farbe) {Diamond, Heart, Spade, Club }.

- (a) Implementieren Sie die Datentypen Rank, Suit und Card.

```
1 {-
2   Rundet eine Gleitkommazahl n auf die k-te Nachkommastelle.
3   Beispiel: n = 3.4343232, k = 4 sollte 3.4343 ergeben.
4   Beispiel: n = 8.9385723, k = 2 sollte 8.94 ergeben.
5   Beispiel: n = 2.9238477, k = 6 sollte 2.923848 ergeben.
6 -}
7 roundDecimalPlace :: Double -> Integer -> Double
8 roundDecimalPlace n k = fromIntegral(round (n*10^k)) / fromIntegral(10^k)
```

- (b) Implementieren Sie Card als eine Instanz der Klasse Ord. Dabei entscheidet der Wert (Rank) welche Karte größer ist, bei zwei Karten gleichen Wertes entscheidet die Farbe (Suit), welche Karte größer ist. Bei gleichem Wert und gleicher Farbe sind beide Karten gleich groß.

```
1 {-
2   Rundet eine Gleitkommazahl n auf die k-te Nachkommastelle.
3   Beispiel: n = 3.4343232, k = 4 sollte 3.4343 ergeben.
4   Beispiel: n = 8.9385723, k = 2 sollte 8.94 ergeben.
5   Beispiel: n = 2.9238477, k = 6 sollte 2.923848 ergeben.
6 -}
7 roundDecimalPlace :: Double -> Integer -> Double
8 roundDecimalPlace n k = fromIntegral(round (n*10^k)) / fromIntegral(10^k)
```

- (c) Implementieren Sie den Datentypen Hand, der aus drei Karten besteht. Implementieren Sie anschließend eine Funktion value :: Hand → Integer, welchen den Wert der Hand zurückgibt. Dabei seien folgende Handkombinationen definiert:

Hand	Beispiel	Value
nichts	7♣ 8♦ 10♠	0
Paar (2 Karten selben Wertes)	8♦ 8♠ 8♣	1
Drilling (3 Karten selben Wertes)	B♣ B♥ B♠	2
Flush (3 Karten selber Farbe)	K♥ 9♥ 7♥	3

```
1 {-
2   Rundet eine Gleitkommazahl n auf die k-te Nachkommastelle.
3   Beispiel: n = 3.4343232, k = 4 sollte 3.4343 ergeben.
4   Beispiel: n = 8.9385723, k = 2 sollte 8.94 ergeben.
5   Beispiel: n = 2.9238477, k = 6 sollte 2.923848 ergeben.
```

Informatik I - Gruppe 1 - Übungblatt 5

Ausgabe: 14.11.2022

Abgabe: 21.11.1022

Tutor: Tim Völker

Ali Kurt 528961

Thomas Kujawa 463620

Felix Hoff 366927

```
6  -}  
7  roundDecimalPlace :: Double -> Integer -> Double  
8  roundDecimalPlace n k = fromIntegral(round (n*10^k))/ fromIntegral(10^k)
```

Informatik I - Gruppe 1 - Übungsblatt 5

Ausgabe: 14.11.2022

Abgabe: 21.11.1022

Tutor: Tim Völker

Ali Kurt 528961

Thomas Kujawa 463620

Felix Hoff 366927

Aufgabe P5.5: Eigene Datentypen (1 + 2 + 2 + 2 = 7 Punkte)

In dieser Aufgabe sollen Sie einen Datentyp Currency erstellen. Dabei sei eine Currency entweder Euro, Dollar oder Yen. Der Wechselkurs zwischen den verschiedenen Currencies sei dabei:

1 Dollar = 0,90 Euro

1 Yen = 0,0083 Euro

1 Dollar = 108,59 Yen

Bearbeiten Sie hierfür folgende Schritte:

- (a) Implementieren Sie den Datentyp Currency, der entweder Euro, Dollar oder Yen repräsentiert. Dabei bestehen Euro und Dollar aus zwei Integern für Euro und Cents bzw. Dollar und Cents und Yen besteht nur aus einem Integer.

```
1 {-
2   Rundet eine Gleitkommazahl n auf die k-te Nachkommastelle.
3   Beispiel: n = 3.4343232, k = 4 sollte 3.4343 ergeben.
4   Beispiel: n = 8.9385723, k = 2 sollte 8.94 ergeben.
5   Beispiel: n = 2.9238477, k = 6 sollte 2.923848 ergeben.
6 -}
7 roundDecimalPlace :: Double -> Integer -> Double
8 roundDecimalPlace n k = fromIntegral(round (n*10^k))/ fromIntegral(10^k)
```

- (b) Leiten Sie den Datentyp Currency von der Klasse Show ab und geben Sie Euro und Dollar in der Form 12,03€ bzw. 14,60\$ an und Yen in der Form 120¥. Verwenden Sie hierbei nicht deriving (Show).

```
1 {-
2   Rundet eine Gleitkommazahl n auf die k-te Nachkommastelle.
3   Beispiel: n = 3.4343232, k = 4 sollte 3.4343 ergeben.
4   Beispiel: n = 8.9385723, k = 2 sollte 8.94 ergeben.
5   Beispiel: n = 2.9238477, k = 6 sollte 2.923848 ergeben.
6 -}
7 roundDecimalPlace :: Double -> Integer -> Double
8 roundDecimalPlace n k = fromIntegral(round (n*10^k))/ fromIntegral(10^k)
```

- (c) Schreiben Sie eine Funktion toEuro, die eine Currency entgegennimmt, den entsprechenden Betrag in Euro umrechnet und diesen dann als Currency zurück gibt.

```
1 {-
2   Rundet eine Gleitkommazahl n auf die k-te Nachkommastelle.
3   Beispiel: n = 3.4343232, k = 4 sollte 3.4343 ergeben.
```

Informatik I - Gruppe 1 - Übungsblatt 5

Ausgabe: 14.11.2022

Abgabe: 21.11.1022

Tutor: Tim Völker

Ali Kurt 528961

Thomas Kujawa 463620

Felix Hoff 366927

```
4 Beispiel: n = 8.9385723, k = 2 sollte 8.94 ergeben.
5 Beispiel: n = 2.9238477, k = 6 sollte 2.923848 ergeben.
6 -}
7 roundDecimalPlace :: Double -> Integer -> Double
8 roundDecimalPlace n k = fromIntegral(round (n*10^k))/ fromIntegral(10^k)
```

- (d) Leiten Sie den Datentyp Currency von den Klassen Eq und Ord ab. Beachten Sie dabei, dass verschiedene Currencies verglichen werden können. Verwenden Sie hierbei nicht deriving (Eq, Ord). Geben Sie ein Beispiel als Kommentar an.

```
1 {-
2 Rundet eine Gleitkommazahl n auf die k-te Nachkommastelle.
3 Beispiel: n = 3.4343232, k = 4 sollte 3.4343 ergeben.
4 Beispiel: n = 8.9385723, k = 2 sollte 8.94 ergeben.
5 Beispiel: n = 2.9238477, k = 6 sollte 2.923848 ergeben.
6 -}
7 roundDecimalPlace :: Double -> Integer -> Double
8 roundDecimalPlace n k = fromIntegral(round (n*10^k))/ fromIntegral(10^k)
```
