RWTH Aachen **WS 2013/14**
Lehrgebiet Theoretische Informatik Blatt 11 Gruppe A
Frohn–Reidl–Rossmanith–Sánchez–Ströder

Minipräsenzübung zur Vorlesung Programmierung

Aufgabe 11 (11 Punkte)

Benutzen Sie im Folgenden für Ihre Lösung keine rekursiven Aufrufe, sondern lediglich eine der Methoden `foldl`,`filter` oder `map` mit einer von Ihnen bereitgestellten Funktion als erstes Argument.

Sei `xs` eine Liste vom Typ `[Int]`. Multiplizieren Sie alle Elemente der Liste `xs` miteinander.

Zur Erinnerung:

- `foldl :: (a -> b -> a) -> a -> [b] -> a`:
  Der Aufruf `foldl f a [b1,b2,...,bn]` entspricht dem Aufruf
  `f (... (f (f (f a b1) b2) b3 )... ) bn`, d.h. `f` wird sukzessive auf die Element der Liste Aufgerufen, wobei das Ergebnis des vorherigen Aufrufs als erster Parameter übergeben wird.
- `map :: (a -> b) -> [a] -> [b]`:
  Der Aufruf `map f [a1,a2,...]` ergibt die Liste `[f a1,f a2,...]`.
- `filter :: (a -> Bool) -> [a] -> [a]`:
  Der Aufruf `filter f [a1,a2,...]` liefert eine Liste derjenigen Elemente `ai`, für welche `f ai` wahr ist.

RWTH Aachen **WS 2013/14**
Lehrgebiet Theoretische Informatik Blatt 11 Gruppe B
Frohn–Reidl–Rossmanith–Sánchez–Ströder

Minipräsenzübung zur Vorlesung Programmierung

Aufgabe 11 (11 Punkte)

Benutzen Sie im Folgenden für Ihre Lösung keine rekursiven Aufrufe, sondern lediglich eine der Methoden `foldl`,`filter` oder `map` mit einer von Ihnen bereitgestellten Funktion als erstes Argument.

Sei `xs` eine Liste vom Typ `[Int]`. Erstellen Sie eine Liste, die nur die geraden Elemente der Liste `xs` enthält.

Zur Erinnerung:

- `foldl :: (a -> b -> a) -> a -> [b] -> a`:
  Der Aufruf `foldl f a [b1,b2,...,bn]` entspricht dem Aufruf
  `f (... (f (f (f a b1) b2) b3 )... ) bn`, d.h. `f` wird sukzessive auf die Element der Liste Aufgerufen, wobei das Ergebnis des vorherigen Aufrufs als erster Parameter übergeben wird.
- `map :: (a -> b) -> [a] -> [b]`:
  Der Aufruf `map f [a1,a2,...]` ergibt die Liste `[f a1,f a2,...]`.
- `filter :: (a -> Bool) -> [a] -> [a]`:
  Der Aufruf `filter f [a1,a2,...]` liefert eine Liste derjenigen Elemente `ai`, für welche `f ai` wahr ist.

RWTH Aachen
Lehrgebiet Theoretische Informatik
Frohn–Reidl–Rossmanith–Sánchez–Ströder

WS 2013/14
Blatt 11 Gruppe C

Minipräsenzübung zur Vorlesung Programmierung

Aufgabe 11 (11 Punkte)

Benutzen Sie im Folgenden für Ihre Lösung keine rekursiven Aufrufe, sondern lediglich eine der Methoden `foldl`,`filter` oder `map` mit einer von Ihnen bereitgestellten Funktion als erstes Argument.

Sei `xs` eine Liste vom Typ `[Int]`. Bestimmen Sie das Maximum der Liste `xs`. Sie dürfen davon ausgehen, dass die Liste nicht leer ist und nur positive Elemente enthält.

Zur Erinnerung:

- `foldl :: (a -> b -> a) -> a -> [b] -> a`:
  Der Aufruf `foldl f a [b1,b2,...,bn]` entspricht dem Aufruf
  `f (... (f (f (f a b1) b2) b3 )... ) bn`, d.h. `f` wird sukzessive auf die Element der Liste Aufgerufen, wobei das Ergebnis des vorherigen Aufrufs als erster Parameter übergeben wird.
- `map :: (a -> b) -> [a] -> [b]`:
  Der Aufruf `map f [a1,a2,...]` ergibt die Liste `[f a1,f a2,...]`.
- `filter :: (a -> Bool) -> [a] -> [a]`:
  Der Aufruf `filter f [a1,a2,...]` liefert eine Liste derjenigen Elemente `ai`, für welche `f ai` wahr ist.

RWTH Aachen **WS 2013/14**
Lehrgebiet Theoretische Informatik Blatt 11 Gruppe D
Frohn–Reidl–Rossmanith–Sánchez–Ströder

Minipräsenzübung zur Vorlesung Programmierung

Aufgabe 11 (11 Punkte)

Benutzen Sie im Folgenden für Ihre Lösung keine rekursiven Aufrufe, sondern lediglich eine der Methoden `foldl`,`filter` oder `map` mit einer von Ihnen bereitgestellten Funktion als erstes Argument.

Sei `xs` eine Liste vom Typ `[Int]`. Erstellen Sie eine Liste, welche die Quadrate der Zahlen in `xs` enthält.

Zur Erinnerung:

- `foldl :: (a -> b -> a) -> a -> [b] -> a`:
  Der Aufruf `foldl f a [b1,b2,...,bn]` entspricht dem Aufruf
  `f (... (f (f (f a b1) b2) b3 )... ) bn`, d.h. `f` wird sukzessive auf die Element der Liste Aufgerufen, wobei das Ergebnis des vorherigen Aufrufs als erster Parameter übergeben wird.
- `map :: (a -> b) -> [a] -> [b]`:
  Der Aufruf `map f [a1,a2,...]` ergibt die Liste `[f a1,f a2,...]`.
- `filter :: (a -> Bool) -> [a] -> [a]`:
  Der Aufruf `filter f [a1,a2,...]` liefert eine Liste derjenigen Elemente `ai`, für welche `f ai` wahr ist.

RWTH Aachen **WS 2013/14**
Lehrgebiet Theoretische Informatik Blatt 11 Gruppe E
Frohn–Reidl–Rossmanith–Sánchez–Ströder

Minipräsenzübung zur Vorlesung Programmierung

Aufgabe 11 (11 Punkte)

Benutzen Sie im Folgenden für Ihre Lösung keine rekursiven Aufrufe, sondern lediglich eine der Methoden `foldl`,`filter` oder `map` mit einer von Ihnen bereitgestellten Funktion als erstes Argument.

Sei `xs` eine Liste vom Typ `[Int]`. Erstellen Sie eine Liste, die diejenigen Elemente der Liste `xs` enthält, die kleiner als 5 sind.

Zur Erinnerung:

- `foldl :: (a -> b -> a) -> a -> [b] -> a`:
  Der Aufruf `foldl f a [b1,b2,...,bn]` entspricht dem Aufruf
  `f (... (f (f (f a b1) b2) b3 )... ) bn`, d.h. `f` wird sukzessive auf die Element der Liste Aufgerufen, wobei das Ergebnis des vorherigen Aufrufs als erster Parameter übergeben wird.
- `map :: (a -> b) -> [a] -> [b]`:
  Der Aufruf `map f [a1,a2,...]` ergibt die Liste `[f a1,f a2,...]`.
- `filter :: (a -> Bool) -> [a] -> [a]`:
  Der Aufruf `filter f [a1,a2,...]` liefert eine Liste derjenigen Elemente `ai`, für welche `f ai` wahr ist.

RWTH Aachen **WS 2013/14**
Lehrgebiet Theoretische Informatik Blatt 11 Gruppe A
Frohn–Reidl–Rossmanith–Sánchez–Ströder

Minipräsenzübung zur Vorlesung Programmierung

Aufgabe 11 (11 Punkte)

Benutzen Sie im Folgenden für Ihre Lösung keine rekursiven Aufrufe, sondern lediglich eine der Methoden `foldl`,`filter` oder `map` mit einer von Ihnen bereitgestellten Funktion als erstes Argument.

Sei `xs` eine Liste vom Typ `[Int]`. Multiplizieren Sie alle Elemente der Liste `xs` miteinander.

Zur Erinnerung:

- `foldl :: (a -> b -> a) -> a -> [b] -> a`:
  Der Aufruf `foldl f a [b1,b2,...,bn]` entspricht dem Aufruf
  `f (... (f (f (f a b1) b2) b3 )... ) bn`, d.h. `f` wird sukzessive auf die Element der Liste Aufgerufen, wobei das Ergebnis des vorherigen Aufrufs als erster Parameter übergeben wird.
- `map :: (a -> b) -> [a] -> [b]`:
  Der Aufruf `map f [a1,a2,...]` ergibt die Liste `[f a1,f a2,...]`.
- `filter :: (a -> Bool) -> [a] -> [a]`:
  Der Aufruf `filter f [a1,a2,...]` liefert eine Liste derjenigen Elemente `ai`, für welche `f ai` wahr ist.

Lösungsvorschlag:

RWTH Aachen **WS 2013/14**
Lehrgebiet Theoretische Informatik Blatt 11 Gruppe B
Frohn–Reidl–Rossmanith–Sánchez–Ströder

Minipräsenzübung zur Vorlesung Programmierung

Aufgabe 11 (11 Punkte)

Benutzen Sie im Folgenden für Ihre Lösung keine rekursiven Aufrufe, sondern lediglich eine der Methoden `foldl`,`filter` oder `map` mit einer von Ihnen bereitgestellten Funktion als erstes Argument.

Sei `xs` eine Liste vom Typ `[Int]`. Erstellen Sie eine Liste, die nur die geraden Elemente der Liste `xs` enthält.

Zur Erinnerung:

- `foldl :: (a -> b -> a) -> a -> [b] -> a`:
  Der Aufruf `foldl f a [b1,b2,...,bn]` entspricht dem Aufruf
  `f (... (f (f (f a b1) b2) b3 )... ) bn`, d.h. `f` wird sukzessive auf die Element der Liste Aufgerufen, wobei das Ergebnis des vorherigen Aufrufs als erster Parameter übergeben wird.
- `map :: (a -> b) -> [a] -> [b]`:
  Der Aufruf `map f [a1,a2,...]` ergibt die Liste `[f a1,f a2,...]`.
- `filter :: (a -> Bool) -> [a] -> [a]`:
  Der Aufruf `filter f [a1,a2,...]` liefert eine Liste derjenigen Elemente `ai`, für welche `f ai` wahr ist.

Lösungsvorschlag:

RWTH Aachen **WS 2013/14**
Lehrgebiet Theoretische Informatik Blatt 11 Gruppe C
Frohn–Reidl–Rossmanith–Sánchez–Ströder

Minipräsenzübung zur Vorlesung Programmierung

Aufgabe 11 (11 Punkte)

Benutzen Sie im Folgenden für Ihre Lösung keine rekursiven Aufrufe, sondern lediglich eine der Methoden `foldl`,`filter` oder `map` mit einer von Ihnen bereitgestellten Funktion als erstes Argument.

Sei `xs` eine Liste vom Typ `[Int]`. Bestimmen Sie das Maximum der Liste `xs`. Sie dürfen davon ausgehen, dass die Liste nicht leer ist und nur positive Elemente enthält.

Zur Erinnerung:

- `foldl :: (a -> b -> a) -> a -> [b] -> a`:
  Der Aufruf `foldl f a [b1,b2,...,bn]` entspricht dem Aufruf
  `f (... (f (f (f a b1) b2) b3 )... ) bn`, d.h. `f` wird sukzessive auf die Element der Liste Aufgerufen, wobei das Ergebnis des vorherigen Aufrufs als erster Parameter übergeben wird.
- `map :: (a -> b) -> [a] -> [b]`:
  Der Aufruf `map f [a1,a2,...]` ergibt die Liste `[f a1,f a2,...]`.
- `filter :: (a -> Bool) -> [a] -> [a]`:
  Der Aufruf `filter f [a1,a2,...]` liefert eine Liste derjenigen Elemente `ai`, für welche `f ai` wahr ist.

Lösungsvorschlag:

RWTH Aachen **WS 2013/14**
Lehrgebiet Theoretische Informatik Blatt 11 Gruppe D
Frohn–Reidl–Rossmanith–Sánchez–Ströder

Minipräsenzübung zur Vorlesung Programmierung

Aufgabe 11 (11 Punkte)

Benutzen Sie im Folgenden für Ihre Lösung keine rekursiven Aufrufe, sondern lediglich eine der Methoden `foldl`,`filter` oder `map` mit einer von Ihnen bereitgestellten Funktion als erstes Argument.

Sei `xs` eine Liste vom Typ `[Int]`. Erstellen Sie eine Liste, welche die Quadrate der Zahlen in `xs` enthält.

Zur Erinnerung:

- `foldl :: (a -> b -> a) -> a -> [b] -> a`:
  Der Aufruf `foldl f a [b1,b2,...,bn]` entspricht dem Aufruf
  `f (... (f (f (f a b1) b2) b3 )... ) bn`, d.h. `f` wird sukzessive auf die Element der Liste Aufgerufen, wobei das Ergebnis des vorherigen Aufrufs als erster Parameter übergeben wird.
- `map :: (a -> b) -> [a] -> [b]`:
  Der Aufruf `map f [a1,a2,...]` ergibt die Liste `[f a1,f a2,...]`.
- `filter :: (a -> Bool) -> [a] -> [a]`:
  Der Aufruf `filter f [a1,a2,...]` liefert eine Liste derjenigen Elemente `ai`, für welche `f ai` wahr ist.

Lösungsvorschlag:

RWTH Aachen **WS 2013/14**
Lehrgebiet Theoretische Informatik Blatt 11 Gruppe E
Frohn–Reidl–Rossmanith–Sánchez–Ströder

Minipräsenzübung zur Vorlesung Programmierung

Aufgabe 11 (11 Punkte)

Benutzen Sie im Folgenden für Ihre Lösung keine rekursiven Aufrufe, sondern lediglich eine der Methoden `foldl`,`filter` oder `map` mit einer von Ihnen bereitgestellten Funktion als erstes Argument.

Sei `xs` eine Liste vom Typ `[Int]`. Erstellen Sie eine Liste, die diejenigen Elemente der Liste `xs` enthält, die kleiner als 5 sind.

Zur Erinnerung:

- `foldl :: (a -> b -> a) -> a -> [b] -> a`:
  Der Aufruf `foldl f a [b1,b2,...,bn]` entspricht dem Aufruf
  `f (... (f (f (f a b1) b2) b3 )... ) bn`, d.h. `f` wird sukzessive auf die Element der Liste Aufgerufen, wobei das Ergebnis des vorherigen Aufrufs als erster Parameter übergeben wird.
- `map :: (a -> b) -> [a] -> [b]`:
  Der Aufruf `map f [a1,a2,...]` ergibt die Liste `[f a1,f a2,...]`.
- `filter :: (a -> Bool) -> [a] -> [a]`:
  Der Aufruf `filter f [a1,a2,...]` liefert eine Liste derjenigen Elemente `ai`, für welche `f ai` wahr ist.

Lösungsvorschlag: