

Übungen zur Vorlesung
Deklarative Programmierung: Sommersemester 2018

Nr. 3, Abgabe bis 7.5.2018 11:59 Uhr

Aufgabe 3.1: Entwurfsrezept zur Funktionsdefinition - 1

4 Punkte

Betrachten Sie folgende Problemstellung:

„Wir möchten ein Programm erstellen, welches sowohl die Oberfläche, als auch das Volumen eines Zylinders berechnet. Das Volumen eines Zylinders ist das Produkt seiner Grundfläche und der Höhe. Seine Oberfläche ist die Summe der beiden Grundflächen, sowie der Fläche des Mantels. Die Fläche des Mantels ergibt sich wiederum aus dem Produkt des Grundflächenumfangs und der Höhe.“

Sie sollen nun schrittweise die beiden Funktionen (`cylinder-volume radius height`) und (`cylinder-area radius height`) entwickeln. Führen Sie dazu die Schritte 1 bis 3 des Entwurfsrezepts aus.

Aufgabe 3.2: Entwurfsrezept zur Funktionsdefinition - 2

4 Punkte

Implementieren Sie nun die beiden Funktionen aus Aufgabe 3.1. Achten Sie darauf, alle Teilprobleme in eigene Funktionen auszulagern und vermeiden Sie Redundanzen.

Hinweis: Verwenden Sie die vordefinierte Konstante `pi` als Annäherung der Kreiszahl.

Aufgabe 3.3: Batchprogramme

4 Punkte

In dieser Aufgabe sollen Sie ein lauffähiges Programm entwickeln, welches das aktuelle Alter in Jahren in Abhängigkeit des übergebenen Geburtsdatums berechnet und ausgibt. Passen Sie dazu das zur Verfügung gestellte Programmgerüst (`batch-skel.rkt`) an:

- Definieren Sie zunächst geeignete Testfälle für die `age`-Funktion.
- Implementieren Sie die `age`-Funktion
- Rufen Sie das Programm mit einem gültigen Datum von der Kommandozeile aus auf und fertigen Sie einen Screenshot der Ausgabe an.
- Übersetzen Sie das Programm in eine **ausführbare** Datei. Sehen Sie sich dazu die Dokumentation zu `raco` (<https://docs.racket-lang.org/raco/exe.html>) an. Rufen Sie nun auch dieses Programm mit einem gültigen Datum auf und fertigen Sie ebenfalls einen Screenshot an.

Hinweis: Gehen Sie bei der Implementierung der `age`-Funktion davon aus, dass Sie stets gültige und vergangene Daten erhalten.

Aufgabe 3.4: Rocket interactive

4 Punkte

In dieser Aufgabe sollen Sie die Animation der Rakete vom letzten Zettel ein wenig interaktiver gestalten. Bei einem Klick in die Animation, soll sich die Richtung der Rakete umkehren. Passen Sie dazu Ihre Lösung von Aufgabe 2.4 entsprechend an (eine Musterlösung finden Sie in der Datei `rocket-v8.rkt`):

- Der `WorldState` soll ein einfacher Zähler sein. Geben Sie die entsprechende Datendefinition an.
- Definieren Sie die Funktionen `on-tick-event`, `end-of-world` und `render`. Das Programm soll enden, falls die Rakete das obere bzw. untere Ende der Szene erreicht. Jeder Tick soll den `WorldState` um 1 erhöhen, die `render`-Funktion zeichnet – wie bisher – die Rakete abhängig vom übergebenen `WorldState`. Ist das Programm beendet, soll der Text „GAME OVER“ mittig in der Szene platziert werden.
- Definieren Sie einen Mouse-Handler (`on-mouse-event`), welcher den `WorldState` bei einem Klick so modifiziert, dass die Rakete ihre Richtung ändert.
- Verbinden Sie alle Funktionen durch einen geeigneten Aufruf der `big-bang`-Funktion. Achten Sie darauf, dass die Rakete in der Mitte der Szene startet.

Geben Sie zu jeder Funktion Signatur und Aufgabenbeschreibung in einem Kommentar an und definieren Sie geeignete Testfälle.