



## INFORMATIK II

### PROGRAMMIERÜBUNGSBLATT 1

Ausgabe: Do, 17.04.2014 Abgabe: 17te Kalenderwoche

Für die Bearbeitung der Übungsaufgaben haben Sie 90 Minuten Zeit. Bitte geben Sie Ihre Lösungen der Aufgaben rechtzeitig im Vorlesungssystem unter dem Punkt Präsenzübungen ab. Dazu müssen Sie sich mit Ihrem Benutzernamen und Passwort am Vorlesungssystem anmelden. Sie können mehrmals innerhalb der Bearbeitungszeit Ihre Lösungen abgeben, nur die letzte Version Ihrer Abgabe wird gespeichert. Wenn Sie mit den Übungen fertig sind, bleiben Sie bitte an ihrem Platz sitzen und warten, bis ein Tutor ihre Lösungen bewertet. **Bei Fragen zur Rechnerbenutzung oder zu den Übungsaufgaben wenden Sie sich bitte jederzeit an einen der TutorInnen!**

### Aufgaben (insgesamt 22 Punkte)

**Notiz:** Halten Sie sich beim Programmieren in Scheme an die eingeführte Konstruktionsanleitung:

- a) Kurzbeschreibung als Kommentar (mittels: "; ...");
- b) Signatur (der "Vertrag" mittels: "(: ...)");
- c) Testfälle (mittels "(check-within ...)öder "(check-expect ...)öder ...)
- d) Prozedur Gerüst und Rumpf

Benutzen Sie Hilfsprozeduren, wenn Teilprobleme gelöst werden müssen!

#### 1.1 Pyramide-Kantenlänge (10 Punkte)

Schreiben Sie Prozeduren, die die Kantenlänge einer Pyramide berechnen! Von der Pyramide ist die Seitenlänge der Grundfläche  $a$  und die Höhe  $h$  gegeben.

Die Kantenlänge einer Pyramide ist die Summe der Kantenlänge der quadratischen Grundfläche (mit Seitenlänge  $a$ ) und der Länge der vier nach oben gehenden Kanten  $s$ . Schreiben Sie Prozeduren, die folgende Teilprobleme lösen:

- a) Schreiben Sie eine Prozedur `square-circumference`, die den Umfang der quadratischen Grundfläche berechnet.
- b) Um  $h_s$  und  $s$  zu berechnen, brauchen Sie jeweils den *Satz des Pythagoras*:  $a^2 + b^2 = c^2$   
Schreiben Sie eine Prozedur `pythagoras`, die das  $c$  der obigen Gleichung berechnet. Erkennen und abstrahieren Sie weitere Teilprobleme!
- c) Schreiben Sie eine Prozedur `pyramid-one-edge-length`, die die Länge  $s$  berechnet (dazu brauchen Sie die Länge  $h_s$ ).

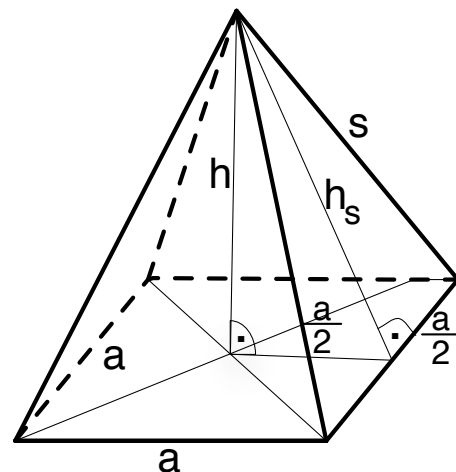


Abbildung 1: Pyramide

- d) Schreiben Sie schließlich eine Prozedur `pyramid-edge-length`, die die Kantenlänge der Pyramide berechnet. Benutzen Sie dafür die bisher geschriebenen Prozeduren.

**Hinweis:** Zur Lösung können Sie die eingebaute Scheme-Prozedur `sqrt`, die die Signatur `(number -> number)` besitzt, verwenden: `(sqrt x)` liefert die Quadratwurzel von  $x$ .

Abgabe: Programm `pyramid-edge.rkt`

## 1.2 Bussgelder (12 Punkte)

Schreiben Sie ein Programm, mit dem Sanktionen bei Verkehrsverstößen bestimmt werden.

- a) (3 Punkte) Programmieren Sie eine Prozedur `zu-langes-parken` für die Bewertung von zu langem Parken auf einem kostenpflichtigen Parkplatz. Die Prozedur bekommt eine Zeitspanne in Minuten übergeben und gibt das entsprechende Verwarngeld zurück.

Diese Verwarnungen sind wie folgt festgelegt:

- Überschreitung der Höchstparkdauer bis einschließlich 30 Minuten: €5
- bis zu einer Stunde: €10
- bis zu zwei Stunden: €15
- bis zu drei Stunden: €20
- länger als drei Stunden: €25

Test: `(check-expect (zu-langes-parken 55) 10)`

- b) (9 Punkte - jeweils 3) Das Überfahren einer roten Ampel kostet je nach Gefährdungslage mehr, gibt Punkte und Fahrverbot. Schreiben Sie eine Prozedur `rote-ampel-bussgeld`, die das Bußgeld berechnet und eine Prozedur `rote-ampel-punkte` für die Punkte in Flensburg. Schreiben Sie außerdem eine Prozedur `rote-ampel-fahrverbot`, die ausgibt, ob ein Fahrverbot erteilt wird. Übergeben Sie den Prozeduren, wie lange die Ampel schon rot war (in Sekunden) und ob eine Gefährdung oder Sachbeschädigung vorlag.

Die Sanktionen sind wie folgt definiert:

- Bei Rot über die Ampel innerhalb der ersten Sekunde: €50 und 3 Punkte
- Bei Rot über die Ampel innerhalb der ersten Sekunde mit Gefährdung oder Sachbeschädigung: €125, 4 Punkte und 1 Monat Fahrverbot
- Bei Rot über die Ampel nach der ersten Sekunde: €125, 4 Punkte und 1 Monat Fahrverbot
- Bei Rot über die Ampel nach der ersten Sekunde mit Gefährdung oder Sachbeschädigung: €200, 4 Punkte und 1 Monat Fahrverbot

Tests: `(check-expect (rote-ampel-bussgeld 1 #f #t) 125)`  
`(check-expect (rote-ampel-fahrverbot 15 #f #f) #t)`  
`(check-expect (rote-ampel-punkte 1 #f #f) 3)`

Befolgen Sie die Konstruktionsanleitung für Prozeduren und Fallunterscheidungen! Achten Sie bei den Testfällen auf eine vollständige Überdeckung!

**Hinweis:** Zur Lösung können Sie die eingebauten Racket-Prozeduren `and` und `or` verwenden.

Abgabe: Programm `bussgeld.rkt`