

Till Mossakowski
Lutz Schröder

Software-Spezifikation in CASL

Übungsblatt 1

Abgabe 13.11.06, 10:00

Übung 1:

Es wird ein Programm gebraucht, das Listen sortiert, d.h. ihre Elemente in eine aufsteigende Reihenfolge bringt. Die Elemente der Liste sollen dabei erhalten bleiben, nicht jedoch ihre Anzahl.

- (a) Schreibe eine entsprechende Anforderungsspezifikation in CASL.
- (b) Schreibe eine Entwurfsspezifikation in CASL. Verwende einen effizienten Sortieralgorithmus, also einen mit Laufzeit $O(n \log n)$.

6 P.

Übung 2:

Identifiziere 5 Probleme der untenstehenden natürlichsprachlichen Spezifikation von Textformatierung. Für jedes Problem

- gib die entsprechende(n) Zeilennummer(n) des Spezifikationstextes an,
- erkläre, wo du das Problem siehst (ggf. mittels eines Beispiels), und
- klassifiziere das Problem entsprechend B. Meyer's 'sieben Sünden des Spezifizierers':

Noise (Rauschen): Überflüssiger Text, der den Leser ablenkt.

Remorse (Reue): Ein Bestandteil der Spezifikation wird an einer Stelle, an der er benutzt wird, nachträglich eingeschränkt, statt bereits bei seiner Definition.

Silence (Stille): Auslassung von Teilen der Spezifikation, die der Spezifizierer als "offensichtlich" ansieht.

Contradiction (Widersprüchlichkeit): In der Spezifikation werden widersprüchliche Aussagen gemacht.

Over-specification (Überspezifikation): Statt das Problem aufzuzeigen, wird eine Lösung beschrieben.

Ambiguity (Mehrdeutigkeit): Aussagen, die auf mehrere Weise interpretiert werden können.

Forward references (Vorwärtsverweise): Verwendung eines Konzepts, bevor es definiert wurde.

6 P.

Übung 3:

Spezifiziere untenstehendes Textformatierungsproblem in CASL.

8 P.

Specification of text formatting

- 1 The program's input is a stream of characters whose end is signalled
2 with a special end-of-text character, ET. There is exactly one ET
3 character in each input stream. Characters are classified as:
4
5 -> break characters - BL (blank) or NL (new line)
6 -> non-break characters - all others except ET;
7 -> the end-of-text indicator - ET.
8
9 A word is a nonempty sequence of non-break characters. A break is a
10 sequence of one or more break characters. Thus, the input can be
11 viewed as a sequence of words separated by breaks, with possibly
12 leading and trailing breaks, and ending with ET.
13
14 The program's output should be the same sequence of words as in the
15 input, with the exception that an oversize word (i.e. a word
16 containing more than MAXPOS characters, where MAXPOS is a positive
17 integer) should cause an error exit from the program (i.e. a variable,
18 Alarm, should have the value TRUE). Up to the point of an error, the
19 program's output should have the following properties:
20
21 a) A new line should start only between words and at the beginning
22 of the output text, if any.
23 b) A break in the input is reduced to a single break character in
24 the output.
25 c) As many words as possible should be placed on each line
26 (i.e. ~between successive NL characters).
27 d) No line may contain more than MAXPOS characters (words and BLs).

(J. Goodenough, S. Gerhart: *Towards a theory of test: Data selection criteria*, in *Current Trends in Programming Methodology* Vol. 2, T. Yeh (ed.), Prentice-Hall 1977, pp. 44-79.)