



FAKULTÄT
FÜR INFORMATIK
Faculty of Informatics

Programmiersprachen
LVA 185.208, VU, 3 ECTS, 2013 S



3. Übungsaufgabe

Complang
Puntigam
Über mich
Research
Lehre

Aufgabe:

Entwickeln Sie in einer stark typisierten funktionalen Sprache (ML oder Haskell) ein Programm zur Verwaltung von Platzreservierungen in Zügen. Jede Reservierung gilt für einen bestimmten Zug zwischen zwei benannten Stationen für eine bestimmte Anzahl an Personen. Muss man von einem Zug zu einem anderen umsteigen, ist für jeden verwendeten Zug eine eigene Reservierung nötig. Die Anzahl der reservierbaren Sitzplätze ist nicht nur durch die Anzahl der vorhandenen Plätze pro Waggon und die Anzahl der Waggons pro Zug begrenzt, sondern auch durch eine Mindestzahl von Plätzen pro Zug, die für Fahrgäste ohne Reservierung frei bleiben müssen. Natürlich darf kein Platz mehrfach belegt sein. Wird eine Reservierung für mehrere Personen gemacht, so müssen die dafür reservierten Plätze im selben Waggon liegen und eine fortlaufende Nummerierung aufweisen (z.B. Waggon 3, Plätze 5 bis 9). Neben Reservierungen werden verschiedene Abfragemöglichkeiten benötigt:

- Das Steckennetz mit allen bekannten Stationen, Zügen (inklusive Anzahl und Größe der Waggons) und Umsteigemöglichkeiten, wobei es mehrere sich kreuzende Strecken sowie Züge pro Strecke geben soll;
- Mindestanzahl der freien und maximale Anzahl der durch Reservierung belegten Plätze pro Zug und Waggon zwischen je zwei Stationen (wobei sich Minimum und Maximum darauf beziehen, dass Reservierungen möglicherweise nur auf Teilen der abgefragten Strecke existieren);
- Reservierungen für einen bestimmten Platz in einem Zug, wobei das Ergebnis die Stationen angibt, zwischen denen Reservierungen bestehen;
- Mindestanzahl der zwischen zwei Stationen freien und der noch reservierbaren Plätze sowie die maximale Gruppengröße (= maximale Zahl der Personen pro Reservierung) für einen Zug oder mehrere gegebene Züge (wenn Umsteigen nötig ist).

Daten sollen beim absichtlichen Beenden des Programms nicht verloren gehen, sondern persistent (in einer Datei) gespeichert werden. Es ist möglich, alle Daten beim Programmstart einzulesen und vor Beendigung neu zu schreiben.

Bei dieser Aufgabe kommt es darauf an, Daten zu verwalten. Dies scheint auf den ersten Blick in Widerspruch zu modernen funktionalen Sprachen zu stehen. Bei genauerer Betrachtung ergeben sich aber viele

Lösungsmöglichkeiten. Ein Blick in die mit den Programmiersystemen mitgelieferten Bibliotheken kann vielleicht bei der Auswahl des Lösungsansatzes helfen.

ML:

Zur Programmierung in ML verwenden Sie am besten OCaml. Dieses System erweitert ML unter anderem um objektorientierte Konzepte, die einen Programmierstil ähnlich dem von Java erlauben würden. Falls Sie Ihr Programm in OCaml schreiben, schränken Sie sich bitte auf die funktionalen Konzepte der Sprache ein, und verzichten Sie auf die objektorientierten Erweiterungen.

Haskell:

Allgemeine Informationen zu Haskell (beispielsweise die Sprachdefinition) finden Sie auf der Haskell-Seite. Sie haben im Wesentlichen die Wahl zwischen GHC und Hugs. GHC ist ein eher umfangreiches und effizientes System mit zahlreichen Sprach-Erweiterungen, das sich auch für größere Anwendungen eignet. Sie können gleich eine einigermaßen vollständige Entwicklungsplattform mit mehreren Werkzeugen installieren. Hugs ist ein einfaches, nur auf einem Interpreter basierendes System, das Sie möglicherweise schon kennen. Für die Lösung dieser Aufgabe reicht Hugs aus, bietet aber nicht die Effizienz und den Komfort von GHC.

Anfang | HTML 4.01 | letzte Änderung: 2013-05-17 (Puntigam)