Lucas Wenke

05252 973750
Lucas.Wenke@pet-rack.de
Am Schlehdorn 10 ♦ 33189 Schlangen

Projekt:

römische Buchhaltung

Erstüberlegungen

Stand: 30.10.2013

05252 973750
Lucas.Wenke@pet-rack.de
Am Schlehdorn 10 ♦ 33189 Schlangen

Inhaltsverzeichnis

05252 973750
Lucas.Wenke@pet-rack.de
Am Schlehdorn 10 ♦ 33189 Schlangen

Aufgabenstellung

Es soll ein Addierer zu Buchhaltungszwecken erstellt werden, der mehrere Zahlen in römischer Schreibweise entgegennimmt, diese addiert und das Ergebnis ebenfalls wieder in römischer Schreibweise ausgibt. Der Kunde hat bereits einige Regeln bezüglich der römischen Zahlnotation sowie Fragen zum Ablauf und Gestalt der Realisierung zusammengefasst. Diese finden sich im Anhang des Dokumentes.

Des weiteren deutete der Kunde im Laufe des Gesprächs an, das er beliebig große Zahlen verarbeiten möchte und infolgedessen eine möglichst kurze Zahlendarstellung wünscht.

Lösungsansätze

Rückführung auf eine bekannte Zahldarstellung

Das Rückführen eines ungelösten Problems auf ein bereits gelöstes gehört mit zu den Standartverfahren in der Mathematik. In diesem Fall müsste die römische Zahldarstellung in eine bekannte interne (binäre) Darstellung überführt werden, in der dann die Kalkulation durchgeführt wird. Deren Ergebnis ist dann wiederum in die römische Schreibweise zu transformieren, wie Abbildung 1 zeigt.

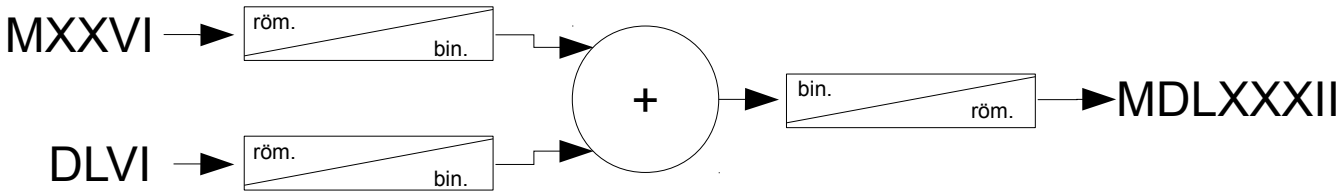

Abbildung 1: Transformation und Retransformation in die interne (binäre) Zahldarstellung, in der die Kalkulation durchgeführt wird.

Die eigentliche Kalkulation kann, da in einem bekannten Zahlensystem durchgeführt, als korrekt angesehen werden. Die Problematik verlagert sich in die Darstellungstransformationen. Deren Allgemeinheit lässt aber Grund zur Hoffnung, das diese bereits in den Standardbibliotheken der moderneren Programmiersprachen umgesetzt sind. In wie weit sich mit ihnen die Anforderungen des Kunden realisieren lassen ist durch eine Recherche zu prüfen.

Ein weiterer Vorteil dieses Vorgehen wäre, das die Kalkulation sich nicht auf die Addition beschränkt, wie es z.B. beim Ries Verfahren der Fall ist.

„addieren auf den Linien“ nach Adam Ries

Es bleibt zu prüfen, inwieweit dieses Rechenverfahren für einen automatisierte Verarbeitung geeignet ist. Der erhoffte Vorteil wäre der Wegfall der im obigen Ansatz nötigen Transformation zwischen den verschiedenen Zahldarstellungen. Das dazu nötige Wissen ist durch eine Recherche zu erwerben.

Lösung eines Drittanbieters

Es steht zu vermuten, das bereits fertige Lösungen von Drittanbietern existieren. Der Aufwand diese auf Korrektheit und Kundenbefriedigung hin zu untersuchen, sowie der Erwerb entsprechender Nutzungslizenzen dürften jedoch in den selben Größenordnungen wie eine Eigenentwicklung liegen, so das dieser Weg erst einmal nicht beschritten wird.

Lucas Wenke
05252 973750
Lucas.Wenke@pet-rack.de
Am Schlehdorn 10 ♦ 33189 Schlangen

Rechercheergebnisse

römische Zahldarstellung

Hauptquelle für dieses Kapitel ist der Wikipediaartikel „Römische Zahlschrift“, zusätzlich wurde noch http://www.diaware.de/html/roemzahl.html und http://www.marco-burmeister.de/helferlein/de_roemische_zahl.html hinzugezogen.

Das römische Zahlensystem ist ein sogenanntes Summationssystem. D.H. die einzelnen Ziffern, die durch Buchstaben repräsentiert werden, haben einen feste Wertigkeit unabhängig von ihrer Position innerhalb der Zahl. Die Wertigkeit wird in folgender Tabelle aufgeschlüsselt:

```
I = 1
V = 5
X = 10
L = 50
C = 100
D = 500
M = 1000
```

Der Zahlenwert ergibt sich aus der Summe der Wertigkeiten aller vorkommenden Ziffern. Diese dürfen auch mehrfach, verteilt über die ganze Zahl, vorkommen.

einfache Darstellung, Kompressionsregel

Um eine unnötige Länge der Zahl zu verhindern, darf jede Ziffer nur so oft vorkommen bis die Summe ihrer Wertigkeiten die Wertigkeit der nächst höheren Ziffer erreicht (**Kompressionsregel**). Ist das der Fall, sind diese durch die nächst höhere Ziffer zu ersetzen. So wird aus „VVV“ → „XV“ oder „IIIIIIIIIIIIIII“ → „VIIIIIIIIII“ → „“VVIIIIII“ → „VVVI“ → „XVI“.
Die nach diesen Regeln erstellten Zahldarstellung ist die sogenannte **einfach Darstellung**. Der Übersichtlichkeit halber ist es üblich die Ziffern von links nach rechts absteigen nach ihrer Wertigkeit zu sortieren.

Abbildung 2: Syntaxdiagramm einer römischen Zahl in einfacher Darstellung und sortierten Ziffern.

Subtraktionsregel

Die Länge einer römischen Zahl in einfacher Darstellung lässt sich durch die Subtraktionsregel weiter verkürzen. Sie macht aus vier Ziffern zwei Zeichen, indem, die vier Ziffern durch die nächst höhere Ziffer mit einer vorgestellten Ziffer ersetzt werden. Somit würde z.B. aus „XXXX“ → „XL“. In Kombination mit einer leicht geänderten Kompressionsregel würde „LXXXX“ → „LXL“ → „XC“. Die Kompressionsregel fasst also die beiden „L“ zu einem „C“ zusammen, muss aber nun die Positionen der „L“ berücksichtigen.

Zu der Subtraktionsregel gibt es zwei Unterarten, die auch miteinander kombiniert werden können.

05252 973750
Lucas.Wenke@pet-rack.de
Am Schlehdorn 10 ♦ 33189 Schlangen

Variante I

Die Subtraktionsregel beschränkt sich nicht nur auf die nächst höherwertige Ziffer, sondern lässt alle Kombinationen zu. So wäre es z.B. möglich für „999“ „IM“ anstelle von „CMXCIX“ schreiben. Der sehr verkürzten Schreibweise stünde aber ein erheblicher Mehraufwand gerade für die Retransformation gegenüber. Hier wäre mit dem Kunden abzustimmen, ob er diese Schreibweise in der Ausgabe wünscht und bereit ist den Mehraufwand zu tragen.

Variante II

die Subtraktionsregel lässt bis zu zwei Ziffern als Subtraktionsziffer zu. Im Falle der „VIII“ würde dann „IIX“ geschrieben.
Wie in Variante I wären auch hier extra Prüfungen von Nöten, die entsprechenden Mehraufwand verursachten. Dementsprechend sollte bei dieser Variante ebenfalls Rücksprache mit dem Kunden gehalten werden.

Darstellungsbereich

In dem Gespräch wünschte der Kunde unter anderem eine unbegrenzten Darstellungsbereich. Dem steht die dem Summensystem inhärente Schwäche bei der Darstellung großer Zahlen entgegen. Bei den römischen Zahlen liegt diese Grenze aufgrund der maximalen Wertigkeit 1000 „M“ ungefähr im Bereich von 4000 sprich „MMMM“. Währen der Recherche ist aufgefallen, das auffallend viele Drittanbieter ihre Lösungen künstlich auf diesen Bereich beschränken.

erweiterte Darstellungen

Das Problem der Darstellung von Zahlen über 1000 existierte bereits in der Antike. Seit dem wurden bis ins Mittelalter hinein verschiedene Schreibweisen entwickelt, die alle eine Art Multiplikator mit 1000 Darstellen.
Hier wäre mit dem Kunden zu klären, ob er eine solche Schreibweise kennt, welche er benutzt bzw. für das Projekt wünscht. Weitere Schritte, wie Aufwandsabschätzungen und Umsetzungskosten, in diese Richtung sollten erst nach Klärung dieser Frage erfolgen.

Prüfung auf Notwendigkeit

Anhand bereits vorhandener Buchhaltungen ließe sich eine objektive und statistisch belastbare Aussage treffen, in wie weit die Forderung nach der großen Zahl in der Vergangenheit berechtigt war. Entstammt die Forderung dem Hinblick auf zukünftige (geplante) Buchhaltungen mit entsprechendem Umfang, ist dieser Schritt hinfällig.

Umstellung auf ein Stellenwertsystem

Sollte beim Kunden nachweislich ein übermäßiger Bedarf an großen Zahlen herrschen, sollte ernsthaft die Einführung eines Stellensystems beim Kunden überlegt werden. Dies ist eine schlichte Abschätzung der Aufwände die einmalig für die Umstellung anfallen, gegen den latenten Mehraufwand, den große römische Zahlen gegenüber einem Stellenwertsystem bedeuten.
Als Einstieg könnte man sich die Tatsache zu nutze machen, das gängige Methoden zur Kalkulation mit römischen Zahlen (Fingerzahl, Rechenbrett) bereits auf Stellensysteme zurückgreifen.

05252 973750
Lucas.Wenke@pet-rack.de
Am Schlehdorn 10 ♦ 33189 Schlangen

Rückführung auf eine bekannte Zahldarstellung mittels Standarfunktion

Eine erste Recherche in den Standardbibliotheken der modernen Programmiersprachen wie C# oder Java erbrachte leider nicht den gewünschten Erfolg. In keiner der beiden Welten fanden sich Anzeichen für eine Transformation in und von der römischen Zahlschreibweise.

Stadtessen scheint dies eine beliebte Aufgabe für Studenten, die programmieren lernen, zu sein. Dementsprechend viele Treffer fanden sich in Foren, Universitätsseiten, etc.
Die dort vorgestellten und diskutierten Lösungen sind allerdings meistens pädagogisch motiviert und lass Sonderfälle wie die Varianten der Subtraktionsregel außen vor.
Hinzu kommen urheberrechtliche Probleme, so das diese Lösungen zwar als Anreiz für unser Projekt dienen können aber so nicht übernommen werden können. Die Transformation und Retransformation römischer Zahlen ist also als Eigenentwicklung umzusetzen.

„addieren auf den Linien“

Das von Adam Ries bekannt gemachte Verfahren, auch als Rechenbrett bekannt, lässt sich mit Zahlen einfacher Darstellung mittels reiner String Operationen (Abzählen und Ersetzen) umsetzen. Die zu addierenden Zahlen werden konkateniert, der Ergebnisstring mittel Kompressionsregel in die einfache Zahldarstellung überführt. Die Eingabe ließe sich auf die Subtraktionsdarstellung erweitern, indem diese in einem vorgeschalteten Schritt in die einfache Darstellung überführt wird.

Vorteile:

- Eine eventuell komplexe und damit fehleranfällige Transformation in eine interne Zahldarstellung fällt weg.
- Die Größe der Zahl ist nur durch die maximale Länge des Strings begrenzt.

Nachteil:

- Als Operation steht nur die Addition zu Verfügung.

Algorithmen

Ziel der Algoritmen ist ei eine vollständie Kette der Darstellungarten aufzubauen:

sub → rom → int → rom → sub

wobei sub für Subtraktionsdarstellung, rom für die einfache Darstellungund int für die interne Zahldarstellung steht.

einfache Darstellung → interne Darstellung (rom2int)

$$int.\,Darstellung = \sum_{i=0}^{i-1} Wertigkeit\left(a_i\right)$$

a_i *Ziffern der röm. Zahl* in *einfacher Darstellung*

n *Länge der röm Zahl*

interne Darstellung → römische Ziffer (int2rom)

Wird nach Rücksprache mit dem Kunden erstellt.

Subtraktionsdarstellung → einfache Darstellung (sub2rom)

Ersetzung von Zeichenmustern nach folgender Tabelle:

05252 973750
Lucas.Wenke@pet-rack.de
Am Schlehdorn 10 ♦ 33189 Schlangen

„IV“ → „IIII“
„IX“ → „VIIII“
„XL“ → „XXXX“
„XC“ → „LXXXX“
„CD“ → „CCCC“
„CM“ → „DCCCC“

Weitere Variationen die durch Afnahme der Variante I oder Variante II entstehen werden nach Absprache mit dem Kunden aufgenommen.

einfache Darstellung → Subtraktionsdarstellung (rom2sub)

Wird nach Rücksprache mit dem Kunden erstellt.

Sortierung nach Wertigkeiten (sort)

Hinweis:
diese Funktion darf nur auf römische Ziffer in einfacher Darstellung angewendet werden. Ziffern in Subtraktionsschreibweise würden verfälscht!

beliebiges bekanntes Sortierverfahren mit der Ordnung:
„M“ > „D“ > „C“ > „L“ > „X“ > „V“ > „I“

wobei die höchste Wertigkeit „M“ links steht.

Durch die Beschränkung auf einfache Darstellungen entfallen auch Überlegungen zu erweiterten Darstellungen bei großen Zahlen wie der Multiplikationsschreibweise.

Kompressionssionsregel (kompress)

1. Fange mit der kleinsten Wertigkeit „I“ an und ersetze solange fünf dieser Ziffern durch die nächst höhere Ziffer wie möglich.
2. führe Schritt 1 mit der nächst höherwertigen Ziffer aus, solange man nicht bei „M“ angelangt ist.

Hinweis:

Durch Sortieren und geschicktes Vorgehen kann das Ziffernzählen durch schnelleres Patternmatching und Musterersetzung ersetzt werden.

Technologie

Aufgrund der in diesem Dokument gewonnenen Erkenntnisse habe ich mich für eine Realisierung als Browserapplikation mittels Javascript entschieden, die das Rechenbrett nach Adam Ries umsetzt.
Damit lässt sich der Prototyp ohne weiteren Aufwand auf jedem Endgerät mit JavaScript fähigem Bowser ausführen.

Die Umsetzung erfolgt mittels „Komposer“ einem HTML- WYSIWYG Editor mit integriertem Texteditor.

Die Programmierung erfolgt prozedural. Ein objektorientierter Ansatz dürfte bei der Größe des Prototypen nur Mehraufwand und keine Vorteile bringen.

05252 973750
Lucas.Wenke@pet-rack.de
Am Schlehdorn 10 ♦ 33189 Schlangen

Qualitätssicherung

Das Hauptproblem wurde gemäß dem Vorgehen „divide and conquer“, Teile und herrsche, in kleine überschaubare Probleme aufgeteilt. Diese wurden exakt spezifiziert, und gegen diese Spezifizierung verifiziert. Sie können somit als korrekt umgesetzt angenommen werden.

Die Zusammenführung der Teillösungen zur Gesamtlösung erfolgt wieder nach einer Spezifikation, gegen die wiederum verifiziert wird. Somit ist die Gesamtlösung ebenfalls verifiziert.

In der von mir verwendeten Methode wird die Funktionsspezifikation als Kommentar im Code hinterlegt, wobei unter Funktion bereits das Aneinanderfügen mehrerer Anweisungen zu verstehen ist.
Die Verifikation erfolgt dann durch Durchlaufen des Codes z.B. im Debugger, wobei händisch der aktuelle Status mit den Erwartungen abgeglichen wird. Stimmen diese in jedem Zweig einer Methode, der Prozedur überein, gilt diese als verifiziert.

App

die App kann zur Zeit unter der URL http://www.digidact.de/ abgerufen werden.