

ILV Informatik (NTG), Jgst. 10

Seminarsitzung 3 (Doppelstunde): Mögliche Inhalte einer Sitzung zum Thema Codierung

Es wird davon ausgegangen, dass sich die Kursteilnehmerinnen und -teilnehmer im Rahmen der vorangegangenen „Studierzeit“ durch Rechercheaufträge zu ausgewählten Themen aus dem Bereich Codierung wie z. B. Morse-Code, Blindenschrift, EAN/GTIN, ASCII-Code, Farbcode, QR-Code u. Ä. informiert haben.

Einstieg:

Ausgehend von unterschiedlichen Codierungsverfahren kann die prinzipielle Idee von Codierung (injektive Abbildung von Zeichen eines Zeichenvorrats in Zeichen oder Zeichenfolgen eines anderen Zeichenvorrats) herausgearbeitet werden.

Beispiele:

- ASCII-Code: Abbildung von Einzelzeichen in natürliche Zahlen
- Morse-Code: Abbildung von Buchstaben in Folgen von Kurz- und Lang-Tönen samt Pausen
- Farbcode: Farben werden durch Tripel natürlicher Zahlen codiert
- EAN bzw. ISBN: konkrete Artikel bzw. Buchtitel werden in Ziffernfolgen abgebildet
- ...

Konkrete Betrachtung der EAN-13/GTIN-13:



Mitteilung: Der eigentliche Zahlencode besteht aus den ersten zwölf Ziffern, die letzte Ziffer ist eine sogenannte Prüfziffer.

Vorteile dieser Codierung: In einzelnen Zifferngruppen lassen sich in einer für die maschinelle Verarbeitung günstigen Form Informationen über Herkunftsland, Hersteller und den Artikel selbst unterbringen.



Verfahren zur Ermittlung der Prüfziffer aus den ersten zwölf Ziffern:

Die Prüfziffer der EAN-13/GTIN-13-Nummern wird berechnet, indem man die ersten zwölf Ziffern abwechselnd mit 1 und 3 multipliziert (links mit 1 anfangen) und diese Produkte summiert. Die Prüfziffer erhält man, indem man die errechnete Summe vom nächstgrößeren Vielfachen von 10 subtrahiert. Falls die Summe durch 10 teilbar ist, ist die Prüfziffer die 0.

Hier konkret: $4 \cdot 1 + 0 \cdot 3 + 7 \cdot 1 + 0 \cdot 3 + 0 \cdot 1 + 7 \cdot 3 + 1 \cdot 1 + 9 \cdot 3 + 6 \cdot 1 + 7 \cdot 3 + 0 \cdot 1 + 7 \cdot 3 = 108$. Die Differenz zum nächstgrößeren Vielfachen von 10 (hier 110) ist 2, dies ist die Prüfziffer.

„EAN-13/GTIN-13 ist ein fehlererkennender Code.“

Mögliche Aufgaben:

Aufgabenidee 1: Berechnen Sie jeweils die Prüfziffer für vorgegebene EAN-13/GTIN-13 (12 Ziffern gegeben).

Aufgabe 2: Schreiben Sie eine Methode, die aus einer als Zeichenkette gegebenen zwölfstelligen Ziffernfolge die Prüfziffer berechnet oder setzen Sie das entsprechende Berechnungsverfahren in einem Rechenblatt eines Tabellenkalkulationssystems um.

Lösung zu Aufgabe 2 in Java:

```
public int pruefzahlBerechnen(String eingabe){
    int pruefsumme = 0;
    int pruefziffer, z1, z2;
    for (int i = 0; i < 12; i = i + 2){
        z1 = (int) eingabe.charAt(i) - 48;
        z2 = (int) eingabe.charAt(i + 1) - 48;
        pruefsumme = pruefsumme + z1 + 3 * z2;
    }
    pruefziffer = 10 - pruefsumme % 10;
    if (pruefziffer == 10){
        pruefziffer = 0;
    }
    return pruefziffer;
}
```

Kommentar: die zwölfstellige EAN-13/GTIN-13 ohne Prüfziffer wird als String übergeben; die Einzelzeichen werden von vorne beginnend in Zweierblöcken ausgelesen. Die Verwendung des Datentyps char für die Einzelzeichen bringt mit sich, dass man nach dem Casting zu einem Integer-Wert den ASCII-Code erhält, was in der nachfolgenden Berechnung berücksichtigt werden muss. Damit ergibt sich aber die Möglichkeit, den evtl. im Einstieg schon thematisierten ASCII-Code wieder aufzugreifen.

Beispiel für ein entsprechendes Rechenblatt:

	A	B	C	D	E	F	G	H
1	EAN ohne Prüfziffer:						Prüfsumme:	
2								
3	407007196707		407007196707	7	21		108	
4			40700719670	0	0			
5			4070071967	7	21		Prüfziffer:	
6			407007196	6	6			
7			40700719	9	27		2	
8			4070071	1	1			
9			407007	7	21			
10			40700	0	0			
11			4070	0	0			
12			407	7	7			
13			40	0	0			
14			4	4	4			
15								
16								
17								
18								

Kommentar: Verwenden der ganzzahligen Division (Spalte C) und Rest bei Division (Spalte D) sowie der VRunden-Funktion in G7 (= VRUNDEN(G3;10))

Aufgabe 3: Entscheiden Sie, ob bei einer EAN-13/GTIN-13 mit genau einer fehlerhaft eingegebenen Ziffer die fehlerhafte Stelle gefunden und korrigiert werden kann, und begründen Sie Ihre Entscheidung.



Lösungsbeispiel:

4070071967072 → 4071071967072 (EAN-13/GTIN-13 mit Eingabefehler)

Prüfziffer der EAN-13/GTIN-13 mit Fehler müsste 9 sein; d. h. es kann erkannt werden, dass eine fehlerhafte Eingabe vorliegt. Es kann aber nicht erkannt werden, welche Ziffer falsch eingelesen wurde, da sich verschiedene Stellen so abändern ließen, dass sich die korrekte Prüfziffer ergeben würde, beispielsweise wäre auch ein Abändern der fehlerhaften EAN-13/GTIN-13 zu 4071078967072 denkbar.

Aufgabe 4: Ändern Sie die aus 13 Ziffern bestehende EAN-13/GTIN-13 eines Artikels an zwei der ersten zwölf Stellen so ab, dass die dann entstehende (nicht mehr zum ursprünglichen Artikel passende) EAN-13/GTIN-13 trotzdem die ursprüngliche Prüfziffer hat.

Mögliche Lösung: 4070071967072 → 3080071967072, d. h. bei zwei fehlerhaften Eingaben ist eine Fehlererkennung nicht mehr garantiert.

Bemerkung: Die folgende Aufgabe soll die Bedeutung der Gewichte in der Prüfsummenberechnung hervorheben.

Aufgabe 5: Erläutern Sie, unter welcher Bedingung durch das gegebene Verfahren erkannt werden kann, dass in einer fehlerhaften EAN-13/GTIN-13 unter den ersten zwölf Ziffern zwei benachbarte verschiedene Ziffern vertauscht sind.

Lösung: Zwei benachbarte Ziffern haben in der Prüfsummenberechnung stets die Gewichte 1 und 3. Bezeichnet man diese Ziffern mit x und y , so ändert sich durch Vertauschen dieser beiden Ziffern die Prüfsumme um $(3x + y) - (x + 3y) = 2x - 2y = 2(x - y)$. Der Fehler kann genau dann nicht erkannt werden, wenn $2(x - y)$ ein Vielfaches von 10 ist, d. h. wenn sich x und y um genau 5 unterscheiden.

Aufgabe 6: Recherchieren Sie die Prüfzifferberechnung für eine 10-stellige ISBN und überprüfen Sie das Verfahren anhand der ISBN eines Ihrer Schulbücher (ist dort eine 13-stellige Nummer angegeben, lassen Sie die ersten drei Ziffern weg; dann erhält man allerdings auch eine andere Prüfziffer).

Auf die Möglichkeit, hier auch Fehler zu korrigieren, soll aus Zeitgründen in der Regel nicht eingegangen werden. Bei sehr leistungsstarken Schülerinnen und Schülern kann evtl. eine entsprechende Aufgabe für die Studierzeit gestellt werden; Problem ist dabei, dass nur die älteren Bücher neben der EAN-13 auch noch die alte 10-stellige ISBN ausweisen, bei der sich allerdings eine andere Prüfziffer ergibt.



Binäre Codierung und Dualzahlen:

Barcodes und QR-Codes als Beispiele für eine binäre Codierung; die Verarbeitung erfolgt auf Rechner Ebene als Folgen von 0 und 1. Dies führt zu der Notwendigkeit, Zahlen aus der gewohnten Darstellung im Dezimalsystem in Dualzahlen (d. h. in das Binärsystem) umzuwandeln.

Hier ist nicht daran gedacht, die konkrete Codierung zu decodieren bzw. nachzuvollziehen; die Schülerinnen und Schüler sollen lediglich eine Vorstellung von dem Begriff „binäre Codierung“ entwickeln. Ein spielerischer Zugang, der diese Entwicklung fördern kann, findet sich auf der Seite <http://www.mal-den-code.de>.

Erste einfache Aufgaben zum Umwandeln von Dezimalzahlen in Dualzahlen und umgekehrt; ebenso von Dezimalzahlen in Hexadezimalzahlen und umgekehrt.

Konkrete Betrachtung von achtstelligen Bitfolgen unter folgenden Gesichtspunkten:

- Begriffe Bit und Byte
- Welcher Zahlenraum lässt sich damit abdecken?
- Möglichkeit, eine solche achtstellige Folge kompakt als zweistellige Hexadezimalzahl zu schreiben (*kann aber auch in den Unterpunkt „Adressierung von Geräten“ im Komplex „Kommunikation in Netzwerken, Internet“ verschoben werden*)
- Erläuterungen zu Größenangaben von Speichern: Kilobyte, Megabyte, Gigabyte, Terabyte,...
- Evtl. einfache Berechnungen, z. B. Download-Dauer bei gegebener Verbindungsgeschwindigkeit, Speicherbelegung von Fotos bei gegebener Auflösung, ...

Übungen:

Anknüpfung an den Einstieg: Insbesondere nähere Betrachtung von ASCII-Code oder Farbcodes aus Sicht der maschinellen Verarbeitbarkeit.

Mögliche Aufgaben:

Aufgabe 1: Berechnen Sie die Anzahl unterschiedlicher Zeichen, die mit einem Byte codiert werden können.

Lösung: $2^8 = 256$



Aufgabe 2: Berechnen Sie die Anzahl der möglichen Farben, wenn die Intensität des Rot-, Grün- und Blauanteils jeweils durch ein Byte codiert wird.

Lösung: $256^3 = 16.777.216$

Aufgabe 3: Decodieren Sie den folgenden durch ASCII codierten Text: 01000011 01001111 01000100 01000101

bzw. hexadezimal 43 4F 44 45

Lösung: CODE

Nachbereitung:

Nachbereitung der vergangenen Seminarsitzung in der Studierzeit:

- Paritätsbit als Möglichkeit zur Fehlererkennung in Binärfolgen

Hier analog wie bei der EAN-13/GTIN-13 auf die Möglichkeit der Fehlererkennung eingehen; Möglichkeiten zur Fehlerkorrektur könnten hier durchaus auch von den Schülerinnen und Schülern selbständig erarbeitet werden. Evtl. entsprechende Seite zum „parity magic trick“ bei CS Unplugged (<https://csunplugged.org/en/topics/error-detection-and-correction/unit-plan/parity-magic/>) zum Selbststudium bzw. in ähnlicher Form in LOGIN in Heft 168 Jahrgang 2011 vorgeben.

- Recherche zum UTF-8-Code

Abhängig davon, wie ausführlich auf den ASCII-Code eingegangen wurde

- Konkrete Aufgaben zur Umwandlung von Zahldarstellungen
- Aufbau von MAC-Adressen in Hexadezimalform zur eindeutigen Identifizierung von Geräten
- Anzahl möglicher verschiedener MAC-Adressen

Vorbereitung der folgenden Seminarsitzung:

Auftrag: Erarbeiten Sie anhand eigener Recherchen die einzelnen Schichten bei einem Schichtenmodell des Internets sowie deren jeweilige Aufgaben.