

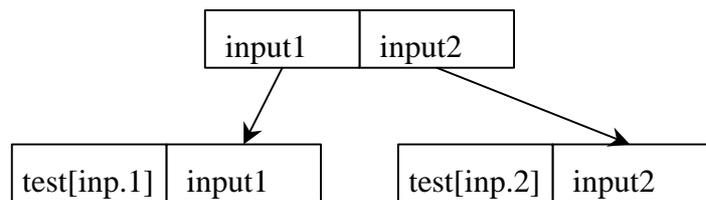
Übung 6: Codierfilter [24]

Schreiben Sie einen FilterOutputStream, einen FilterInputStream und ein Testprogramm nach folgender Spezifikation (es wird dabei ein sogenannter Hamming-Code¹ realisiert):

EncodeOutputStream

Ein Byte wird halbiert, beiden Teilen werden vier Testbits vorangestellt und das Ergebnis geschrieben. Newline-Zeichen sollen erkannt und nicht codiert werden. Die Testbits haben folgendes Aussehen:

```
byte test[] = {
    32, -16, -32, 48,
    -80, 96, 112, -96,
    112, -96, -80, 96,
    -32, 48, 32, -16
};
```

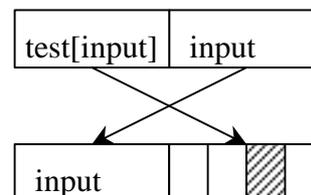


Dabei sind sie bereits geshiftet. Überlegen Sie weiters, ob beim Codieren eine Pufferung notwendig ist.

DecodeInputStream

Beim Decodieren werden Daten- und Testbits vertauscht, die Decodierung selbst funktioniert folgendermaßen: Man überprüfe das dritte Testbit (muss eins sein) und entferne es. Den übrigen Bitvektor aus Daten- und restlichen Testbits multipliziert man mit der Matrix

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$



Damit erhält man das Bitmuster der Stelle eines Fehlers. Bei Auftreten von Fehlern bei drittem Testbit oder Bitvektor sollen Ausnahmen ausgeworfen werden. Ist es beim Decodieren günstig zu puffern?

Testprogramm

Testen Sie obige Filter, wobei Sie auch fehlerbehaftete Eingaben decodieren sollten.

Literatur

MacWilliams, F. J., Sloane, N. J. A.: *The Theory of Error-Correcting Codes I, II*. North-Holland, Amsterdam, 1977.

¹ siehe Literatur