

## 6 Testat Paritätsprüfung

**Übung 20** Die Zeichenkette *Friedberg* soll übertragen werden. Zur Sicherung gegen Übertragungsfehler werden Paritätsinformationen eingebaut. Ergänzen Sie in der Tabelle die Bits für Querparität sowie das zusätzliche Byte für die Längsparität. Verwenden Sie in beiden Fällen gerade Parität.

Querparität	Binär-Darstellung	Zeichen
	100 0110	F
	111 0010	r
	110 1001	i
	110 0101	e
	110 0100	d
	110 0010	b
	110 0101	e
	111 0010	r
	110 0111	g
		Längsparität

**Übung 21** Berechnen Sie für das Divisor-Polynom  $x^3 + x^2 + 1$  die Prüfzeichen zu der Nachricht 101011011.

**Übung 22** Paritätsberechnung mit XOR

Die Parität lässt sich effizient mit der logischen Verknüpfung des exklusiven ODER (XOR, Symbol  $\oplus$ ) realisieren. Berechnen Sie als Beispiel damit die Längsparität der Zeichenfolge *Friedberg* (ohne Querparität). Verwenden Sie dazu folgende Tabelle. Im ersten Schritt wird das Muster 0000 0000 mit dem Bitmuster für *F* verrechnet. Tragen Sie das Ergebnis in der nächsten Zeile ein. Dieser neue Wert wird dann mit dem Bitmuster für *r* verrechnet. Dieses Verfahren wird bis zum letzten Buchstaben fortgesetzt.

	Binär-Darstellung	Zeichen
	0000 0000	
$\oplus$	0100 0110	<i>F</i>
$\oplus$	0111 0010	<i>r</i>
$\oplus$	0110 1001	<i>i</i>
$\oplus$	0110 0101	<i>e</i>
$\oplus$	0110 0100	<i>d</i>
$\oplus$	0110 0010	<i>b</i>
$\oplus$	0110 0101	<i>e</i>
$\oplus$	0111 0010	<i>r</i>
$\oplus$	0110 0111	<i>g</i>
		Längsparität

**Übung 23** Die Java-Klasse *Kanal* stellt eine einfache Simulation für eine fehlerhafte Übertragung dar. Der Ablauf ist folgender:

1. Eingangs- und Ausgangsdatei werden geöffnet
2. Ein Kopf mit standardmäßig 1000 Bytes wird unverändert kopiert. Dies ist eine unrealistische Vorsichtsmaßnahme, um sicherzustellen, dass der Dateityp nach der Übertragung noch erkannt wird.
3. Anschließend wird jeweils ein Block gelesen, nach der vorgegebenen Wahrscheinlichkeit verfälscht und wieder ausgegeben.

Intern wird an jeden Datenblock

Daten
-------

eine Paritätsinformation angehängt (Methode `bildeRahmen`):

Daten	⇒	P
-------	---	---

Der gesamte Block wird übertragen (Methode `uebertrage`). Beim „Empfänger“ wird die gleiche Paritätsberechnung ausgeführt und überprüft, ob das Ergebnis 0 ist (Methode `rahmenFalsch`). Falls nicht, erkennt der Empfänger auf Übertragungsfehler und fordert das Paket erneut an.

Daten	$\stackrel{?}{\Rightarrow}$	P
-------	-----------------------------	---

Ansonsten wird die Paritätsinformation wieder entfernt (Methode `leseRahmen`) und der Block ausgegeben.

Daten
-------

In der vorgegebenen Version gibt die Methode `paritaet` einfach den Wert 0 zurück.

1. Laden Sie die Klasse `Kanal` in BlueJ und erzeugen Sie eine Instanz. Mit der Methode `printSettings` können Sie die vorgegebenen Einstellungen ausgeben lassen. Über entsprechende setter-Methoden können einzelne Werte verändert werden. Mit `setParamString` können in einem Schritt mehrere Werte vorgegeben werden (siehe Klassen-Methode `printHelp`).
2. Dokumentieren Sie zunächst an einem Beispiel (gut geeignet sind Bitmap-Bilder) die Auswirkung von Übertragungsfehler bei wachsender Fehlerwahrscheinlichkeit.
3. Fügen Sie in der Methode `paritaet` die Berechnung der Längsparität mittels XOR ein. Als Test können Sie mit der Methode `test` die Parität der Zeichenkette `Friedberg` berechnen lassen und dann das Resultat mit dem Ergebnis aus Übung 22 vergleichen.
4. Untersuchen Sie, inwieweit beim simulierten Übertragen Fehler tatsächlich erkannt werden. Vergleichen Sie das Ergebnis mit dem ohne Fehlerprüfung. Wie groß ist der zusätzliche Aufwand durch die wiederholten Übertragungsversuche (VORSICHT: bei zu großer Fehlerwahrscheinlichkeit werden ständig Fehler erkannt und Rahmen wiederholt, so dass das Program „ewig“ läuft)?