Die Berechnung einer Wagennummer (Prüfziffer)

Heute melde ich mich einmal mit einem reinen Codeschnipsel. Dabei geht es um die Umsetzung der Berechnung der Prüfziffer einer Wagennummer nach dem deutschen Baureihenschema.
Wir erstellen in diesem Teil kein neues Script. Es geht lediglich um die Rechnung.

Zu Beginn sollte man zuerst wissen, wie die Wagennummer aufgebaut ist.
Beispiele für Wagennummern anhand eines Triebwagens der Baureihe 425/435:

425 155-0
435 155-8
435 655-7
425 655-9

Die ersten drei Ziffern bilden die Baureihennummer (425, 435). Danach kommt die Ordnungsnummer (155, 655) und zum Schluss die Prüfziffer.

Wenn man aber jetzt benutzerdefinierte Wagennummern realisieren möchte, hat man das Problem mit der Prüfziffer. Diese ist Entscheidend für eine gültige Wagennummer.
Die Baureihe lässt sich nicht editieren, die ist gegeben. Die Ordnungsnummer (155, 655) sollte veränderbar sein, damit der Benutzer am Ende die Möglichkeit hat diese neuzudefinieren.
So fahren am Ende nicht immer mehrere Triebwagen mit der gleichen Wagennummer in unserer Trainz-Welt herum.

Jetzt muss man wissen, wie diese Wagennummer, oder vielmehr die Prüfziffer am Ende berechnet wird.
Es ist gar nicht so schwierig:

Alle Ziffern der Wagennummer werden nacheinander mit 1 und 2 multipliziert. Das sähe so aus:

4 * 1 = 4
2 * 2 = 4
5 * 1 = 5

1 * 2 = 2
5 * 1 = 5
5 * 2 = 10

Damit ergibt sich eine neue Zahlenreihe:
4 4 5 2 5 10

Nun errechnet man daraus die Quersumme:
4 + 4 + 5 + 2 + 5 + 10 = 30

Anschließend ermittelt man das nächste Vielfache von 10, in diesem Fall wäre das die 30 selbst.

Nun errechnet man aus der eben errechneten Quersumme und dem Vielfachen von 10 die Differenz:
30 ? 30 = 0, also ist die Prüfziffer von 425 155 die 0 und somit ergibt sich die Wagennummer:

425 155-0

Ein anderes Beispiel:
4 * 1 = 4
3 * 2 = 6
5 * 1 = 5

1 * 2 = 2
5 * 1 = 5
5 * 2 = 10

=> 4 6 5 2 5 10

Quersumme: 4 + 6 + 5 + 2 + 5 + 10 = 32

Das nächste Vielfache von 10: 40
Die Prüfziffer ist also: 40 ? 32 = 8

Nun geht es darum, diese Rechnung in einem Script umzusetzen.
Ich beschränke mich dabei nur auf die Rechnung, ohne dabei auf das restliche Script drumherum einzugehen.

Nach meinem Vorschlag nutzt man für die einzelnen Ziffern der Wagennummer einen Integer-Array.
Man geht jeden Eintrag dieses Arrays durch und multipliziert ihn mit seinem jeweiligen Wert.
Ein Array hat Indizes. Sie fangen bei 0 an und hören bei array.size()-1; auf.
Das bedeutet, jeder Eintrag eines Arrays lässt sich durch seinen Index ansprechen und verarbeiten.

Wir bauen uns also einen Array zusammen, der so aussieht:

Code

```
int[]m_aiWagenNummerZiffern=newint[7];//Diesenarraylegenwir"global"fürdieKlassean

...

Irgendwo im Script, da wo es Sinn macht, können wir den Array den mit den Inhalten füllen.
Das                  funktioniert                  nicht                  "global"!

...

m_aiWagenNummerZiffern[0]   =   4;         //   erste   Ziffer   der   Baureihennummer
m_aiWagenNummerZiffern[1]   =   2;        //   zweite   Ziffer   der   Baureihennummer
m_aiWagenNummerZiffern[2]   =   5;        //   dritte   Ziffer   der   Baureihennummer
m_aiWagenNummerZiffern[3]   =   1;         //   erste   Ziffer   der   Ordnungsnummer
m_aiWagenNummerZiffern[4]   =   5;         //   zweite   Ziffer   der   Ordnungsnummer
m_aiWagenNummerZiffern[5]   =   5;         //   dritte   Ziffer   der   Ordnungsnummer
m_aiWagenNummerZiffern[6]   =  -1;     //   Das   wird   am   Ende   die   Prüfziffer,
                                                                                  //   -1
                                                                                     //   s
                                    // berechnet worden ist.
```

Alles anzeigen

Jetzt können wir uns doch eine Methode schreiben, die anhand dieses Arrays für uns die Berechnung der Wagennummer durchführt.
Wir nennen diese Methode mal ?BerechnePruefziffer?. Dieser übergeben wir unseren Array und lassen uns von ihr einen Array zurückgeben, dessen Wert mit dem Index ?6? unsere Prüfziffer enthält.

Zuerst kümmern wir uns aber um die Multiplikation mit 1 und 2 und die anschließende Berechnung der Quersumme, da dies einfach ist:

Code

```
int[]                   BerechnePruefziffer(int[]                   aiWagenNummer);
int[]                    BerechnePruefziffer(int[]                    aiWagenNummer)
{
                                //        Wir        gehen        den        Array        kompl
//Wirüberspringendenletzten Index,weildieserja-1ist(aiWagenNummer.size()-1[!])

                                                                                  int
                            for(i       =       0;       i       <       aiWagenNummer.size()-1
    {
                                 if(i    %    2    ==    0)                //    i    ist
        {
                                                    iQuersumme    =    iQuersumme
        }
                                     else
        {
                                                        iQuerSumme    =    iQuerSumme    +
        }
    }

 }
```

Alles anzeigen

Damit haben wir schonmal unsere Quersumme. Sieht doch recht ansehnlich aus.
Das ?i % 2? ist einfach nichts anderes als eine Division. ?i? wird durch 2 geteilt. Als Ergebnis erhält man aber nicht den Quotienten, sondern den Rest der Division. Wie damals in den Anfängen der Division 😉

Beispiel: 5 % 2 = 1 [5 / 2 = 2 Rest **1**]

Diesen Operator (?%?) nennt man ?Modulo?. Gesprochen wird die Rechnung: i Modulo 2.

Ich habe das erwähnt, da es im nächsten Schritt eine große Rolle einnehmen wird.
Jetzt geht es darum, die Prüfziffer aus der Quersumme zu berechnen.

Zur Erinnerung: Wir brauchen die Differenz zum nächsten Vielfachen von 10.
Wenn wir mittels Modulo den Rest aus einer Division errechnen können, so können wir als ersten Schritt die Quersumme Modulo 10 rechnen.
Damit bekommen wir den Rest, der übrig bleibt, wenn man die Quersumme durch 10 teilt.
Ein Beispiel: 24 % 10 = 2 Rest 4. Da in der Mathematik bei der Punktrechnung (Multiplikation und Division) die ?Regel? ?positive Zahl? Mal/Durch ?positive Zahl? gleich ?positive Zahl?,
bekommen wir immer ein positives Ergebnis aus der Modulo-Rechnung. Nun können wir einfach das Ergebnis der Modulo-Rechnung von 10 abziehen.
Wir erhalten die Prüfziffer!

Kurz:
10 ? (24 % 10) = 6

Passt!

Nun kann es aber sein, daß die Prüfziffer auch wieder ein Vielfaches von 10 ist. Das kommt auf die Quersumme an.
Darum müssen wir danach noch überprüfen, ob die Prüfziffer ein Vielfaches von 10 ist. Wenn ja, ist die Prüfziffer automatisch gleich 0.
Im Script können wir nun eine Variable für die Prüfziffer deklarieren. Anschließend berechnen wir diese und fügen sie in unseren Array ein und geben ihn zurück.

Code

```
int[]                   BerechnePruefziffer(int[]                   aiWagenNummer);

int[]                    BerechnePruefziffer(int[]                    aiWagenNummer)
{
                                //        Wir        gehen        den        Array        kompl
//Wirüberspringendenletzten Index,weildieserja-1ist(aiWagenNummer.size()-1[!])

                                                                                    int
                            for(i       =       0;       i       <       aiWagenNummer.size()-1
    {
                                 if(i    %    2    ==    0)                //    i    ist
        {
                                                        iQuersumme     =     iQuersumme     +

        }
                                    else
        {
                                                        iQuerSumme     =     iQuerSumme     +

        }
    }

                          int      iPruefZiffer      =      10      -      (iQuersumme      %
   // Prüfen ob die Prüfziffer ein Vielfaches von 10 ist und sie ggf. auf 0 setzen
                         if(iPruefZiffer      %      10      ==      0)      iPruefZiffer      =
           //   Die   Prüfziffer   in   den   Array   eintragen   (Index   =   6,   s.o.)

                                  //         Nun         den         veränderten         Array

}
```

Alles anzeigen

Jetzt können wir diese Methode im Script nutzen. Da muss vom Scripter selbst entschieden werden, wann es sinnvoll ist diese zu verwenden.

Machen kann man das so:
Wir haben zu Anfang unseren Array mit den Werten gefüllt. An der Stelle an der wir die komplette Wagennummer benötigen können wir also eine kleine Abfrage einbauen, die uns verrät, ob unsere Prüfziffer schon berechnet worden ist. Falls nicht, wird sie berechnet und wir können die Wagennummer dann nutzen.

Hier ein möglicher Vorschlag:

Code

```
if(m_aiWagenNummer[6] == -1)    // Die Prüfziffer ist noch -1 => nicht berechnet
{
     // Wagennummer berechnen. Den Alten Array mit dem neuen Array überschreiben
                                                m_aiWagenNummer            =            Berec
}
```

Was macht dieser Codefragment genau?

Er überprüft, ob die Prüfziffer unseren Standardwert hat (-1). Ist dem so, überschreiben wir den aktuellen Array mit dem durch die Methode ?BerechnePruefziffer? zurückgegebenen Array. Mit dem Parameter übergeben wir ja den alten Array und die Methode gibt uns den gleichen Array zurück, nur daß sie die Prüfziffer, die ja noch -1 ist, durch die berechnete Prüfziffer ersetzt. Wenn wir also den Rückgabewert der Methode unserem alten Array zuweisen ist es der gleiche Array, nur mit der Prüfziffer und nicht mit -1. Ich denke, das ist klar.
Die weitere Verwendung steht dem Scripter am Ende frei. Dies hier war wie gesagt nur eine Demonstration der Rechnung.

Danke für deinen Besuch!
Pascal