

Protokoll vom 18. - 19.02.13

In dieser Woche ging es darum das in den vorherigen Stunden über „Schleifen“ oder auch „loops“ gesammelte Wissen zu vertiefen und durch weitere Anwendung zu üben.

Deshalb sollte ein sogenanntes „Summenprogramm“ geschrieben werden. Das bedeutet, dass alle Zahlen von einem Startwert bis zu einem Endwert addiert werden. Dies sollte in drei Versionen geschehen:

1. „Repeat-Schleife“: Der Schleifenrumpf wird solange durchlaufen , bis die *Abbruchbedingung* erfüllt ist. Diese Schleife sollte möglichst mit einer if-Abfrage verbunden werden, da diese Schleifenart immer mindestens einmal durchlaufen wird. Richtige Vorteile gegenüber den anderen Schleifen hat diese nicht, weshalb sie auch dementsprechend wenig benutzt wird.

```
repeat  
  <Anweisung>  
until <Abbruchbedingung>
```

2. „While-Schleife“: hier wird solange der Vorgang in der Schleife solange wiederholt, wie die *Eingangsbedingung* erfüllt ist. Diese Schleife eignet sich gut für Anwendungen, bei denen man den Endwert nicht kennt.

```
while <Eingangsbedingung> do  
  <Anweisung>
```

3. „For-Schleife“: die For-Schleife ist praktisch, wenn man die Anzahl der Durchläufe kennt und kann dann am einfachsten angewendet werden.

```
for <Zaehlvariable> := <Anfangswert> to <Endwert> do  
  <Anweisung>
```

Die nächste Aufgabe war es eine Fakultät zu berechnen. Das bedeutet, dass alle Zahlen von einem Start bis zu einen Endwert multipliziert werden. Also z.B $1*2*3...$ usw. Auch dies soll in drei Versionen mit den oben genannten Schleifen geschehen.

→ Bei dieser Aufgabe war das Ergebnis, dass bei großen Zahlen falsche Ergebnisse ausgegeben werden. Dies liegt an dem *integer-Typ*. Um das zu verhindern muss hier daher der *int64-Typ* bei der Ein- sowie Ausgabe verwendet werden.