# Aufgabe 4 - Sägerei in SLX

Modellieren Sie das gleiche System wie aus Aufgabe 3, aber verwenden Sie als Modellierungssprache diesmal SLX.

Zusätzlich gelten die folgenden weiteren Anforderungen an das Modell in SLX:

- Die Modellierung soll aus Sicht der Stationen als aktive Elemente erfolgen. Aufträge, Blöcke und Blöckchen sind dabei die passiven Elemente.
- Die Modellierung soll im Hinblick auf eine möglichst einfache Austauschbarkeit sowie eine mögliche Erhöhung der Anzahl der einzelnen Stationen (Brückenkräne, Sägen und ATK) erfolgen. Dazu sind die einzelnen Stationen über geeignete Datenstrukturen, die als Zwischenspeicher fungieren, zu entkoppeln.
- Die Modellierung soll nur unter Verwendung von SLX-Kernkonzepten und ohne GPSS-Konzepte erfolgen (u.a. wait until, wait und reactivate).

Zusätzlich gelten die folgenden weiteren Präzisierungen zum Sägerei-System:

- Der Transport eines Blockes vom Blocklager mit einem der Brückenkräne über einen Rollengang zu einer Säge erfolgt erst, wenn die entsprechende Säge frei ist. Das bedeutet, dass sich während des Sägens eines Blockes kein weiterer Block auf dem Rollengang oder davor befindet. Erst mit Beendigung eines Sägevorgangs wird ein Brückenkran wieder aktiv und beschafft den nächsten Block.
- Die Anfangs- und Endstücke eines Blockes sind unbrauchbar und müssen entfernt werden.
- Das Sägen eines weiteren Blöckchens darf erst beginnen nachdem das zuletzt abgesägte Blöckchen vom ATK abtransportiert wurde.

Die Gesamtabarbeitungszeit sollte für den Walzplan „w48-short“ 1506,85 min betragen.

*SLX-Fragment für das Einlesen des Walzplans*

```
#define SLX2 ON

module saegerei {

passive class Walzplan(string(*) datei) {
      set(Auftrag) auftraege;

      initial {
            filedef walzplan input name=datei;
            open walzplan input;
            read newline file=walzplan; // Titelzeile einlesen

            pointer(Auftrag) auftrag;
            forever {
                  auftrag = new Auftrag();
                  string(10) tmp = "";
                  read newline file=walzplan end=lblWalzplanEof (
                        auftrag->nr, auftrag->ringe, tmp,
                        auftrag->bHoehe, tmp, tmp, auftrag->bFormat);
```

```
                  // in dieser Aufgabe werden nur die HMS umgesetzt, deshalb
                  // werden alle nicht verarbeitbaren Aufträge nicht betrachtet
                  if (auftrag->bFormat <= 510) {
                        place auftrag into auftraege;
                  }
            }
            lblWalzplanEof:
                  close walzplan;
      }

}

passive class Auftrag {
      string(8) nr;
      int ringe;
      int bHoehe;
      int bFormat;
}

passive class Ring (pointer(Auftrag) inAuftrag) {
      pointer(Auftrag) auftrag;
      static int idCounter = 0;
      int id;
      initial {
            auftrag = inAuftrag;
            id = idCounter;
            idCounter++;
      }
}

procedure main() {
      sp = new Walzplan("w48-short.txt");

      ...

      int gesamtRinge = 0;
      pointer(Auftrag) a;
      for (a = each Auftrag in sp->auftraege) {
            gesamtRinge += a->ringe;
      }

      print (gesamtRinge) "Gesamtringe: _\n";
      wait until (abl.size == gesamtRinge);
      print (abl.size) "Produzierte Ringe: _\n";
      report(system);
      exit(0);
}

}
```