

SEW- Aufgabe Nebenläufige Roboterfabrik

Willinger Andreas, Brunner Helmuth

4aHit, 10.10.2013 (2013/14)

Inhaltsverzeichnis

Aufgabenstellung:.....	3
Abgabe:.....	3
Arbeitsablauf:.....	4
Implementierung:.....	5
Arbeitsaufteilung:.....	6
Erste Redesignphase:.....	7
Zweite Redesignphase:.....	8
Ausführen des jar:.....	8
Tests:.....	9

Aufgabenstellung:

Es soll eine Spielzeugroboter-Fabrik simuliert werden. Die einzelnen Bestandteile des Spielzeugroboters (kurz Threadee) werden in einem Lager gesammelt. Dieses Lager wird als Verzeichnis und die einzelnen Elementtypen werden als Files im Betriebssystem abgebildet. Der Lagermitarbeiter verwaltet regelmäßig den Ein- und Ausgang des Lagers um Anfragen von Montagemitarbeiter und Kunden zu beantworten. Die Anlieferung der Teile erfolgt durch ändern von Files im Verzeichnis, eine Lagerung fertiger Roboter ebenso.

Ein Spielzeugroboter besteht aus zwei Augen, einem Rumpf, einem Kettenantrieb und zwei Armen. Die Lieferanten schreiben ihre Teile ins Lager-File mit zufällig (PRNG?) erstellten Zahlenfeldern. Die Art der gelieferten Teile soll nach einer bestimmten Zeit gewechselt werden.

Die Montagemitarbeiter müssen nun für einen "Threadee" alle entsprechenden Teile anfordern und diese zusammenbauen. Der Vorgang des Zusammenbauens wird durch das Sortieren der einzelnen Ganzzahlenfelder simuliert. Der fertige "Threadee" wird nun mit der Mitarbeiter-ID des Monteurs versehen.

Es ist zu bedenken, dass ein Roboter immer alle Teile benötigt um hergestellt werden zu können. Sollte ein Monteur nicht alle Teile bekommen, muss er die angeforderten Teile wieder zurückgeben um andere Monteure nicht zu blockieren. Fertige "Threadee"s werden zur Auslieferung in das Lager zurück gestellt.

Alle Aktivitäten der Mitarbeiter muss in einem Logfile protokolliert werden. Verwenden Sie dazu Log4J.

Die IDs der Mitarbeiter werden in der Fabrik durch das Sekretariat verwaltet. Es dürfen nur eindeutige IDs vergeben werden. Das Sekretariat vergibt auch die eindeutigen Kennungen für die erstellten "Threadee"s.

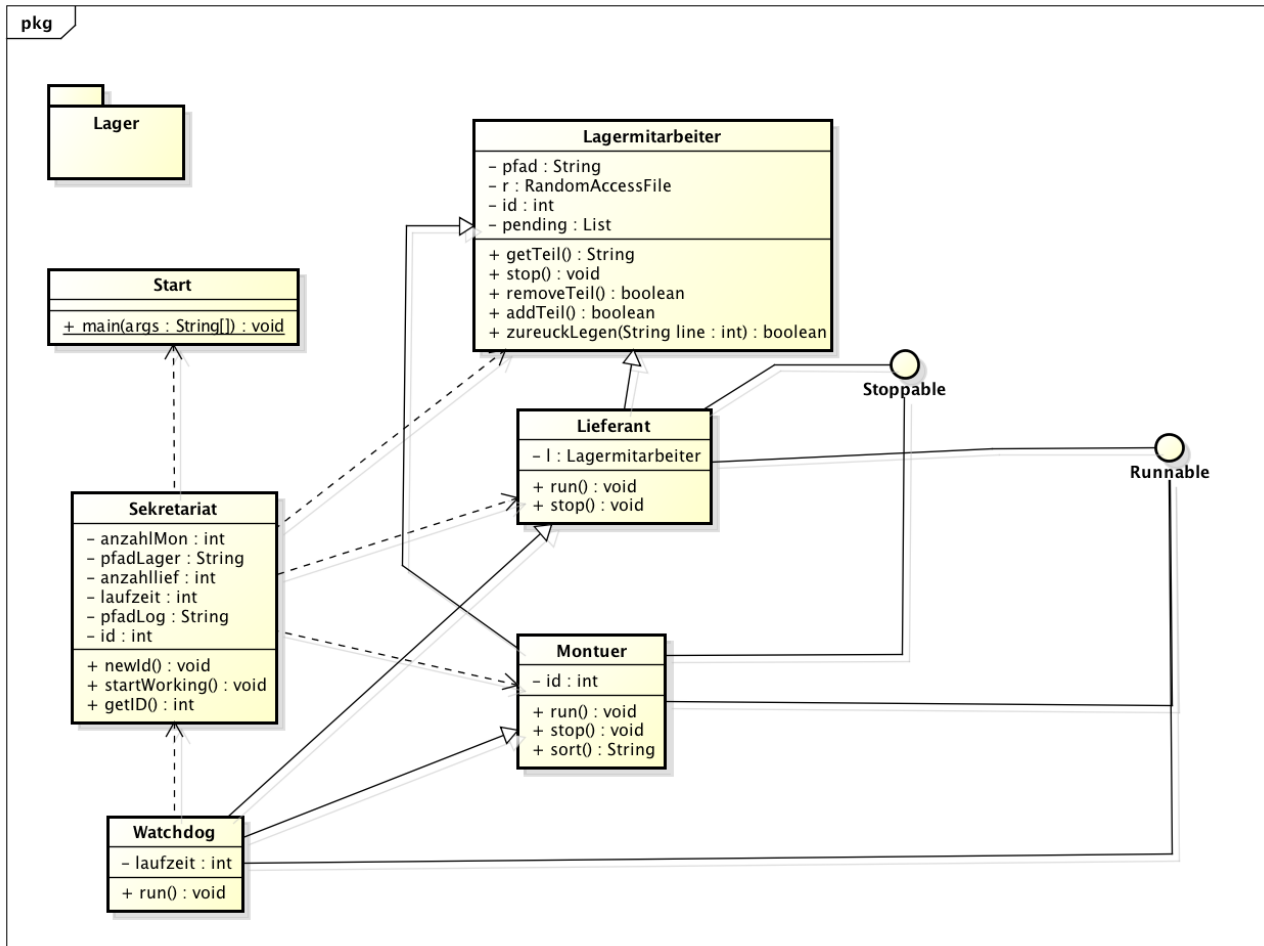
Beachten Sie beim Einlesen die Möglichkeit der Fehler von Files. Diese Fehler müssen im Log protokolliert werden und entsprechend mit Exceptions abgefangen werden.

Abgabe:

Ein jar-File mit einem Source Folder, javaDoc, das UML als Bild und die Dokumentation

Arbeitsablauf:

Erster Entwurf für unser UML.



Implementierung:

Jedoch im Laufe der ersten Implementierungen haben wir feststellen müssen das sich das UML sowie wir es in der ersten Ausführung hatten nicht realisierbar sei. Daher haben wir uns wieder mit weiteren Überlegen an das UML gemacht um die kritischen Teile zu überwinden und das fehlen von machen Funktionen eingebundenen.

Um bestmöglich und immer die aktuellste Version des Source-Codes zur Verfügung zu haben, verwendeten wir zur Versionierung den Dienst von Git.

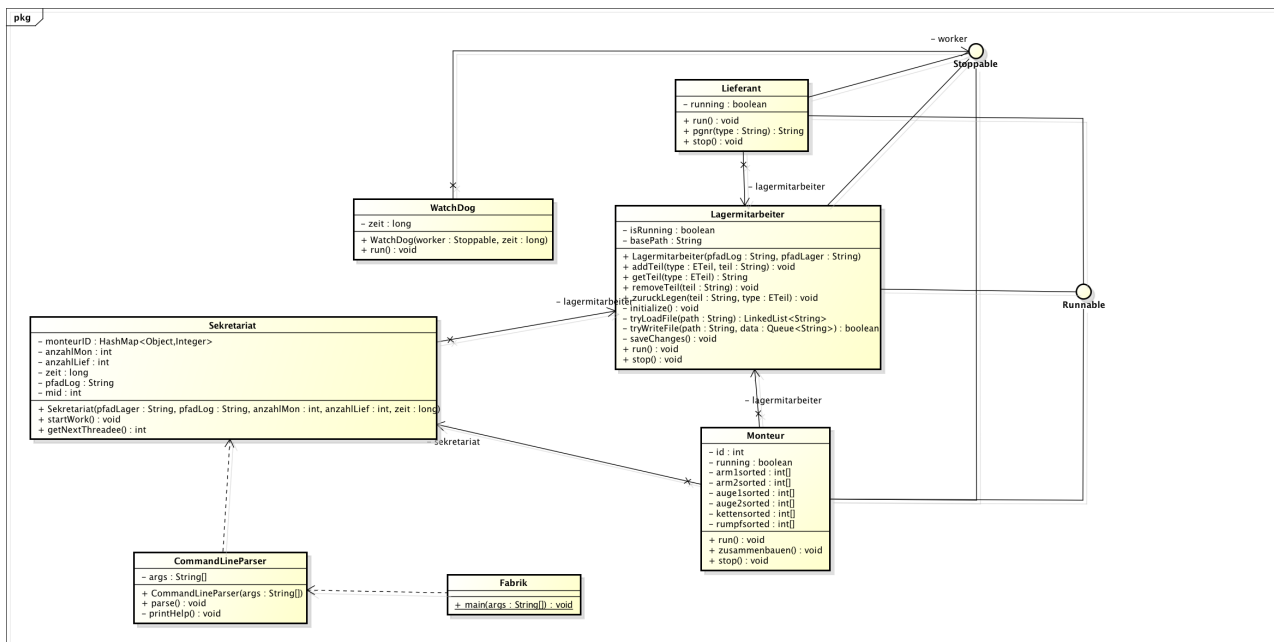
Link zum Repository: <https://github.com/hbrunner-tgm/Threadee>

Arbeitsaufteilung:

Nur eine Grobeübersicht:

Arbeitsschritte	Willinger Andreas	Brunner Helmuth
Logik	x	
Code	x	x
Dokumentation	x	x
Protokoll		x
UML		x

Erste Redesignphase:



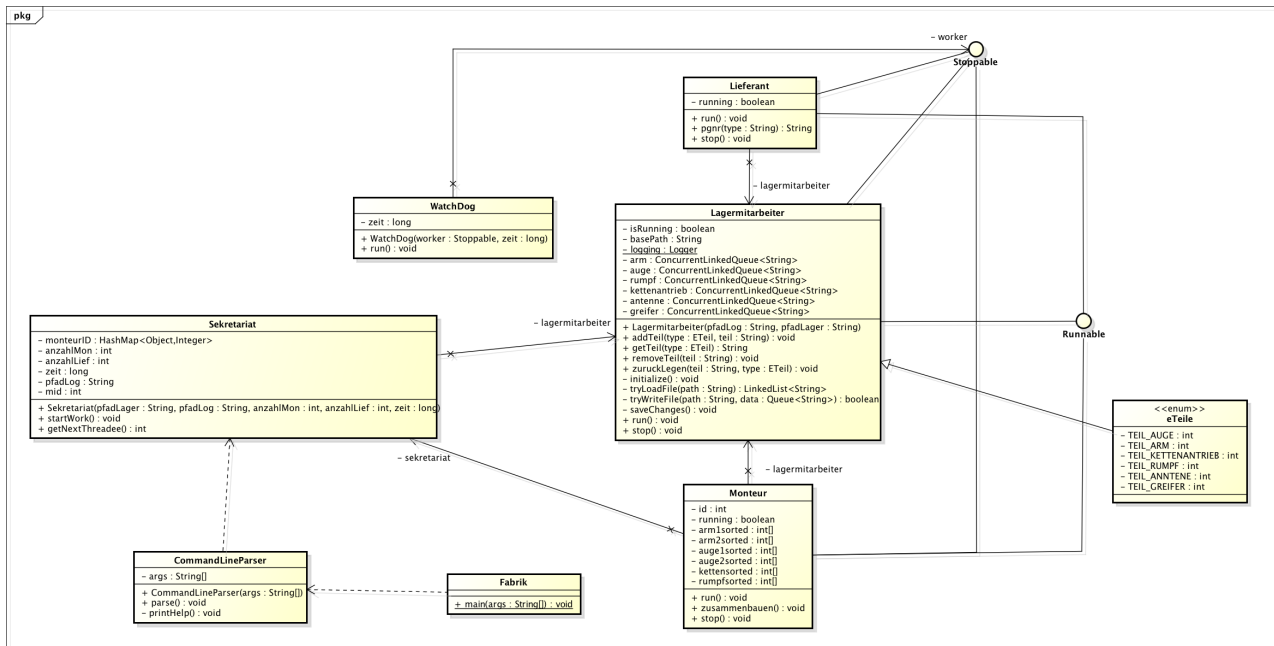
Implementierung weiterer Methoden die für die Umsetzung von Nöten waren.
Einbauen eines Watchdogs sowie den CommandLineParser.

CLI haben wir als externe Library verwendet.

Weiters haben wir den Logging-Services verwendet.

Zweite Redesignphase:

Durch die Erweiterung der Aufgabe haben wir noch die weiteren Teile, die Antenne und die Greifer für die Arme implementiert. In der Folge habe wir auch das Uml-Diagramm geändert.



Ausführen des jar:

Genaus so wie das Beispiel in der Angabe auf elearning.

```
java -jar Threadee --lager /verzeichnis/zum/lager --logs /verzeichnis/zum/loggen
--lieferanten 12 --monteure 25 --laufzeit 10000
```


Tests:

Tests die wir durch geführt haben:

12 Lieferanten, 25 Monteure, 10s Laufzeit.

Ergebniss:

1 Lieferant, 25 Monteure, 10s Laufzeit

Ergebniss:

0 Lieferanten, 25 Monteuer, 10s Laufzeit

Ergebniss: Ausgabe einer Hilfe auf der Commandline weil mehr als 0 Lieferanten angegeben sein müssen.

Quellen angeben

<http://commons.apache.org/sandbox/commons-cli2/manual/index.html>

<http://logging.apache.org/log4j/2.0/manual/configuration.html>