

Dokumentation Schedulingverfahren

von
Norbert Galuschek
Gordian Maugg
Alexander Hahn
Rebekka Weissinger

June 23, 2011



Contents

1 Aufgabe	3
2 Vorgehensweise	4
2.1 Warum Android	4
3 Vorarbeit und Dokumente	5
3.1 API	5
3.2 GUI	7
4 Programmierung	8
4.1 Programmablauf	8
5 Demonstration	9

1 Aufgabe

Das zu entwickelnde Programm ist eine grafische Anwendung, mit der die Benutzer für eine beliebige Anzahl an Prozessen jeweils die CPU-Laufzeit, Ankunftszeit und Priorität eingeben können. Die Anwendung berechnet für die folgenden Schedulingverfahren die Ausführungsreihenfolgen, mittleren Laufzeiten und mittleren Wartezeiten der Prozesse:

- Round Robin mit einem frei festzulegenden Zeitquantum
- First Come First Served
- Shortest Job First
- Longest Job First
- Longest Remaining Time First
- Shortest Remaining Time First
- Prioritätengesteuertes Scheduling

2 Vorgehensweise

Zuerst wollten wir die Aufgabe auf einer Playstation 2 lösen. Dazu wollten wir Linux über diese Booten und das Programm dort zum Laufen bringen. Dabei hatten wir jedoch das Problem, dass die Boot-CD nicht lief. Nach mehreren Versuchen und verschiedener Boot-CD-Versionen gaben wir diese Idee notgedrungen auf. Nun haben wir uns dafür entschieden eine Anwendung für Android mit Java zu schreiben. Dafür haben wir uns zuerst einmal eine Wikiseite bei Google eingerichtet, da wir mit dieser Form schon letztes Semester gute Erfahrungen gemacht haben. Außerdem haben wir uns dazu entschlossen, dass Versionsverwaltungssystem Mercurial zu benutzen und haben dafür das entsprechende Eclipseplugin heruntergeladen und installiert. Um weiterhin mit Eclipse arbeiten zu können und trotzdem eine Anwendung für Android zu schreiben, haben wir das Eclipse android sdk Plugin eingebunden. Über einen Emulator konnte man genau feststellen, wie sich das Layout entwickelt.

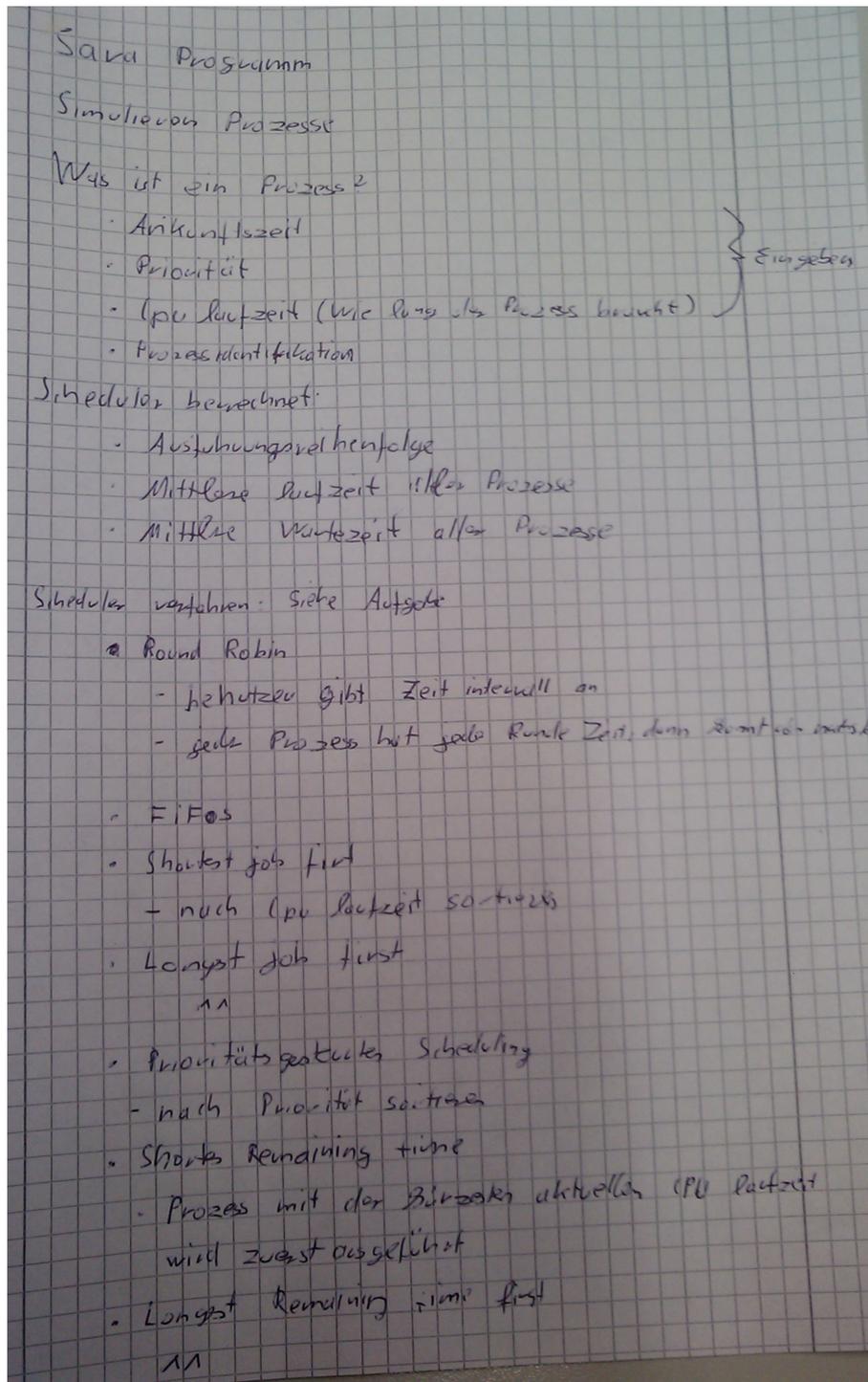
2.1 Warum Android

Da seit Längerem die Smartphones auf dem Markt sind und vor allem das Interesse an Apps besonders groß ist, haben wir uns dieser Herausforderung gestellt. Eine Herausforderung war es, da keiner von uns bis jetzt eine Anwendung für ein Smartphone geschrieben hat. Glücklicherweise hat Norbert ein Android-Phone, damit wir die Software nicht nur über einen Emulator testen müssen. Desweiteren ist das zur Zeit das gängigste Betriebssystem auf Smartphones.

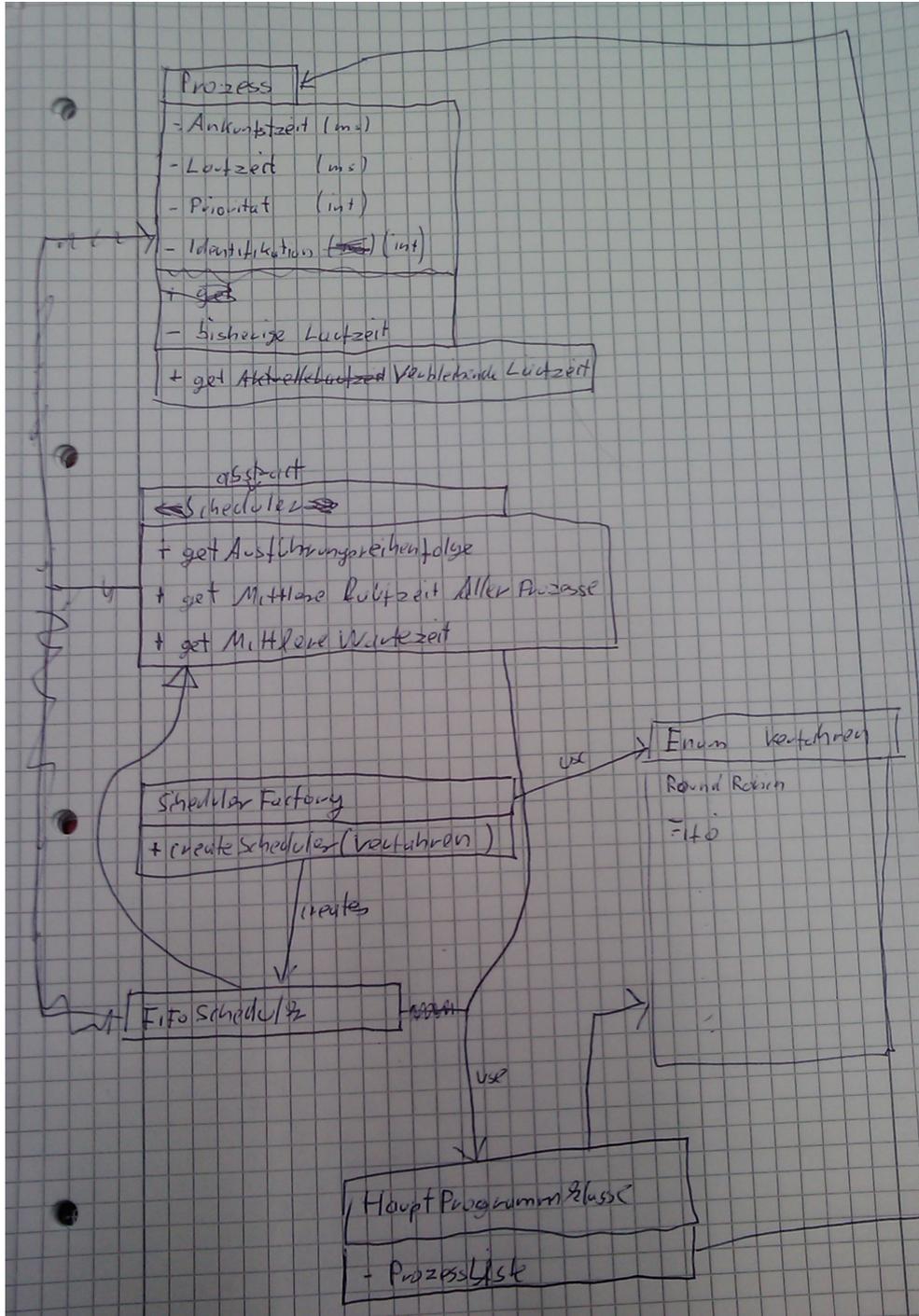
3 Vorarbeit und Dokumente

3.1 API

An erster Stelle machten wir uns Gedanken über die einzelnen Begriffe die in der Aufgabenstellung auftraten. Was ist ein Prozess, bzw was zeichnet ihn aus. Wie sind die einzelnen Scheduling-Verfahren zu verstehen, bzw. was unterscheidet sie von einander. Das Ergebnis ist hier zu sehen.

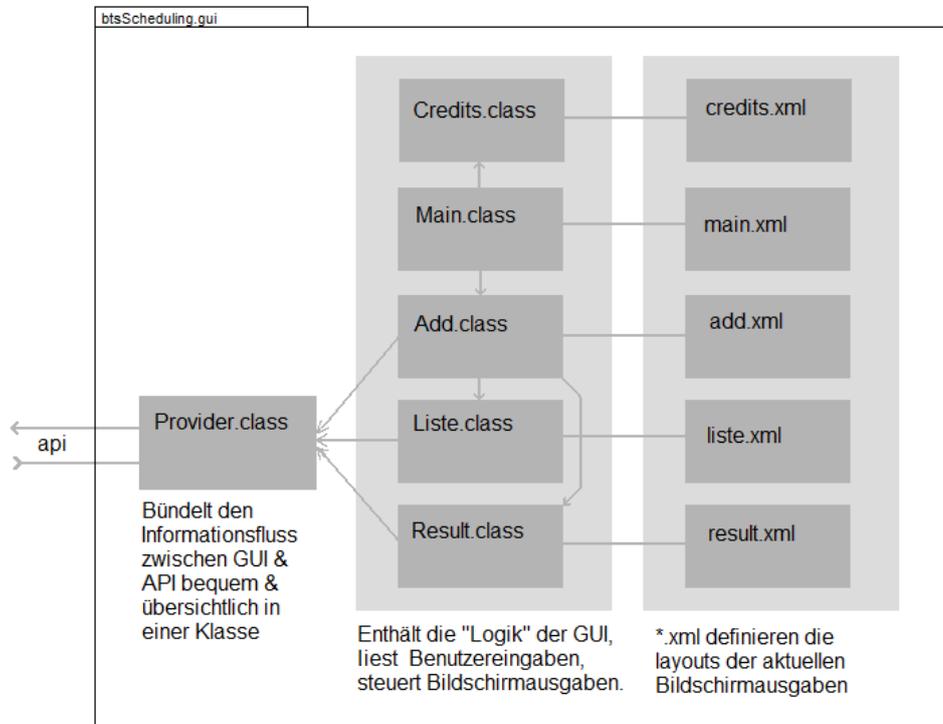


Anschließend machten wir uns Gedanken über das Design der Software. Welche Klassen würden wir benötigen, welche Schnittstellen zwischen gui und api sind sinnvoll, bzw. nötig. Ein erstes Klassendiagramm sah wie folgt aus:



3.2 GUI

Auch für die GUI haben wir einen Strukturplan erstellt. Dieser zeigt die einzelnen java-Klassen, die xml-Layout-Klassen und die Schnittstelle (die Provider-Klasse).



4 Programmierung

Wir haben eine abstrakte Klasse Scheduler erstellt, in der sich die abstrakte Methode getComperator befindet. Von dieser erben die abstrakten Klassen PraeemptivScheduler und NichtPraeemptivScheduler. So musste in den einzelnen Verfahren nur noch der jeweilige Comperator implementiert werden, was die Sache sehr vereinfachte. Außerdem haben wir eine Klasse SchedulerFactory, in der wir die Auswahl des jeweiligen Schedulerverfahrens über eine Enumeration behandeln.

4.1 Programmablauf

Wenn der Benutzer das Programm startet, kommt er in unser Startmenü. Dort hat er die Möglichkeit zwischen den Buttons 'Start' oder 'Credits' zu wählen. Wenn er auf den Creditsbutton drückt, kommt er zu einer Seite, auf der die Mitglieder unseres Teams angezeigt werden. Wenn sich der Benutzer dafür entscheidet auf den Startknopf zu drücken, gelangt er zu dem Bildschirm auf dem man beliebig viele Prozesse erstellen kann. Um einen Prozess anzulegen, gibt er in die Felder 'Ankunftszeit', 'Laufzeit' und 'Priorität' entsprechende Werte ein. Falls der Benutzer das nicht tut, haben wir alle Attributfelder mit Randomwerten vorbelegt. Es gibt den Knopf 'hinzufügen', der den Prozess anlegt. Der Benutzer kann sich alle Prozesse über den Button 'anzeigen' anzeigen lassen, die über eine Liste realisiert wurde, oder er wechselt zur Scheduleransicht ('Schendule!'). Wenn man zur Schudleransicht wechselt, kann der Benutzer dort ein Schedulerverfahren auswählen und ihm wir dann angezeigt wie geschedult wird. Außerdem wird ihm die mittlere Wartezeit und die mittlere Laufzeit der Prozesse angezeigt.

5 Demonstration

