



Universität Karlsruhe (TH)

Lehrstuhl für Programmierparadigmen

Compilerpraktikum SS 2009

Dozent: Prof. Dr.-Ing. G. Snelting

Betreuer: Matthias Braun

Betreuer: Jürgen Graf

<http://pp.info.uni-karlsruhe.de/>

snelting@ipd.info.uni-karlsruhe.de

braun@ipd.info.uni-karlsruhe.de

graf@ipd.info.uni-karlsruhe.de

Übungsblatt 1

Ausgabe: 22.4.2009

Besprechung: 29.04.2009

Aufgabe 1: Infrastruktur

Machen Sie sich mit der Rechnerinfrastruktur am IPD vertraut. Unter <http://pp.info.uni-karlsruhe.de/wiki> finden Sie weitere Informationen.

- Machen Sie sich mit dem Compilerbau-Wiki vertraut: http://pp.info.uni-karlsruhe.de/lehre/SS2009/compprakt/wiki/index.php/Main_Page. Achtung: Es gibt 2 Wikis - Das Lehrstuhl-Wiki und das Compilerbau-Wiki.
- Loggen Sie sich auf einem der Linuxrechner in den Studentenräumen oder von zu Hause auf dem Rechner ssh.info.uni-karlsruhe.de ein. Ändern Sie das Paßwort ihres Benutzerkontos mit `passwd`.
- Überprüfen Sie, ob ihre bevorzugte Entwicklungsumgebung und Werkzeuge funktionieren.
- Legen Sie ein Repository für ein Versionsverwaltungsprogramm Ihrer Wahl an. Eine Anleitung für Subversion befindet sich hier: http://pp.info.uni-karlsruhe.de/wiki/index.php/SVN_Aufsetzen. Testen Sie das Repository.

Aufgabe 2: Vorbereitung

Als nächstes sollten Sie sich mit den Details von MiniJava vertraut machen. MiniJava ist eine vereinfachte Variante der Programmiersprache Java von Sun Microsystems. Einen Sprachbericht finden Sie auf der Praktikumswebseite.

- Schreiben Sie ein MiniJava Programm!
- Ist MiniJava Turingmächtig?
- Handelt es sich um eine strikte Untermenge von Java?
- Welche der Programme 1, 2, 3 sind legale MiniJava Programme?

Aufgabe 3: Planung

Je nachdem welche Programmiersprache Ihnen am besten liegt können Sie sich für C/C++ oder Java entscheiden. Die später für Optimierungen benutzte Bibliothek libFirm (<http://www.libfirm.org>) ist in C geschrieben; Ein Java Wrapper für libFirm ist vorhanden.

```
public class Prog1 {  
    public static void main(String [] args) {  
    }  
}
```

Abbildung 1: Programm 1

```
public class Prog2 {  
    public static void main(String[] args) {  
        System.out.println(args[0]);  
    }  
}
```

Abbildung 2: Programm 2

```
public class Factorial {  
    public int fac(int n) {  
        if (n < 2)  
            return 1;  
        return n * fac(n-1);  
    }  
}  
  
public class Prog3 {  
    public static void main() {  
        Factorial f = new Factorial();  
        int n = fac(42);  
        System.out.println(n);  
    }  
}
```

Abbildung 3: Programm 3