



Universität Karlsruhe (TH)

Lehrstuhl für Programmierparadigmen

Compilerpraktikum SS 2010

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

Betreuer: Matthias Braun

Betreuer: Sebastian Buchwald

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

snelting@ipd.info.uni-karlsruhe.de

matthias.braun@kit.edu

sebastian.buchwald@kit.edu

Übungsblatt 1

Ausgabe: 14.4.2010

Besprechung: 21.04.2010

Aufgabe 1: Infrastruktur, Planung

- Tauschen Sie innerhalb ihrer Gruppe Kontaktdaten aus.
- Organisieren Sie Infrastruktur (Versionsverwaltung, Mailingliste, ...).¹
- Machen Sie sich mit dem Compilerbau-Wiki vertraut: http://pp.info.uni-karlsruhe.de/lehre/SS2010/compprakt/wiki/index.php/Main_Page.
- Entscheiden Sie sich für eine Programmiersprache. Je nachdem welche Programmiersprache Ihnen am besten liegt können Sie Ihren Compiler in C/C++ oder Java entwickeln. Die später für Optimierungen benutzte Bibliothek libFirm (<http://www.libfirm.org>) ist in C geschrieben; Eine Java-Anbindung für libFirm ist allerdings vorhanden. Machen Sie die Entscheidung für C/C++ oder Java abhängig von den Kenntnissen in der Gruppe. Compiler in C/C++ sind typischerweise schneller und effizienter, Java ist einsteigerfreundlicher.
- Wir geben nur Support für Linux und Mac.

Aufgabe 2: Vorbereitung

Als nächstes sollten Sie sich mit den Details der Programmiersprache MiniJava vertraut machen. MiniJava ist eine vereinfachte Variante der Programmiersprache Java von Sun Microsystems. Einen Sprachbericht finden Sie auf der Praktikumswebseite. Achtung: Im Internet finden sich zahllose Sprachen mit dem Namen „MiniJava“ die nichts mit der Sprache in diesem Praktikum zu tun haben.

- 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?

¹Wir erzeugen Ihnen gerne Accounts auf unseren Lehrstuhlrechnern — 1GB Platz, ssh logins, apache+php webserver

```
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