Vorkurs - Tag 3

FSI Informatik

Uni Erlangen-Nürnberg

13. Oktober 2006

Editoren

- Mit einem normalen Editor bearbeitet man Dateien, die **reinen** Text enthalten
- Für formatierten Text o.ä. nimmt man Textverarbeitungsprogramme
- Ein guter Editor (bzw. eine IDE) ist oft ebenso funktionsreich, wie eine Textverarbeitung - nur für Programmierer

- Es lohnt sich, viel Zeit zur Auswahl und zum Einarbeiten in deinen zukünftigen besten Freund zu stecken - er wird sich dafür revanchieren

Graphische Editoren

- Kleine Auswahl (im CIP):
  - Scite
  - Kate
  - Gedit
- Vorteile:
  - Meist intuitiv bedienbar
  - Gute Übersichtlichkeit

Nicht-graphische Editoren

- Kleine Auswahl (im CIP):
  - Vim
  - Emacs
  - Nano
  - Joe
- Vorteile:
  - Starke Affinität zur Tastatur
  - Können problemlos per SSH verwendet werden

Zusätzliche Features

Was man als Programmierer vom Editor erwartet:

- Syntax-Highlighting
  - hebt Schlüsselwörter der Sprache (meist farblich) hervor
  - verbessert die Lesbarkeit und Übersicht
- Automatische Formatierung
  - Einrücken abhängig vom Kontext
  - Geschachtelte Schleifen etc. bleiben durchschaubar
- Code-Faltung
  - Teile des Codes können ausgeblendet werden
  - erhöht die Übersichtlichkeit

Wenn euer Editor das nicht kann, sucht euch einen anderen!

Vor- und Nachteile von Editoren

Vorteile

- gut geeignet für kleinere Programme oder Scripte
- (meist) einfach zu bedienen
- Ressourcenschonend, stehen nicht im Weg
- können auch über ssh genutzt werden (Kommandozeile)

Nachteile

- nur rudimentäre Buildumgebung
- unübersichtlich bei vielen Dateien
- keine Verknüpfung unterhalb der Dateien
- keine oder nur kümmerliche Eingabehilfen

IDE

Integrierte Entwicklungsumgebung

Das Schweizer Taschenmesser des Softwareentwicklers! Sollte alles enthalten, was man als Entwickler brauchen kann.

Bestandteile

- Texteditor (mit erweiterter Funktionalität)
- Compiler bzw. Interpreter eingebunden
- Debuggen direkt im Editor
- Versionsverwaltung und Teamfunktionen
- Spezielle Unterstützung z.B. für GUI
- Referenzdokumentation eingebunden

Vor- und Nachteile von IDEs

Vorteile

- größere Projekte sinnvoll verwalten
- volle Unterstützen bei Eingabe und Fehlersuche
- von der ersten Codezeile bis zur Auslieferung

Nachteile

- langsam und aufgebläht (laut Die-Hard-Fraktion)
- für kleine Projekte "Overkill"
- stößt oftmals doch an seine Grenzen

Erst wenn man sich ausführlich mit einer IDE auseinander gesetzt hat, erkennt man all ihre Stärken!

Eclipse

Eine populäre IDE, die wir auch benutzen!

Fakten

- Freie software
- `http://www.eclipse.org`
- viele Plugins verfügbar
- aus dem Java-Umfeld

Eclipse im CIP-Pool

- unter /local/eclipse zu finden
- subclipse und CDT zusätzlich installiert

Java - Eine Insel oder was?

- Objektorientierte Programmiersprache
- State of the art!

- Quellcode wird in Bytecode übersetzt und dann interpretiert
- Java-Code ist plattformunabhängig!

- umfangreiche Klassenbibliothek vorhanden
- zu finden unter: `http://java.sun.com/j2se/1.5.0/docs/api/`
- wer sich in der Java-API (Application Programming Interface) auskennt, spart sich viel Arbeit!

Die ersten Schritte

In Java entwickelt man ähnlich wie in C, C++, Pascal, ...

- Menschenlesbarer (und hoffentlich schreibbarer) Code in `.java`-Dateien
- Programm, das diesen Code in Bytecode übersetzt (Compiler) - der Java-Übersetzer heißt `javac`

- Aus jeder `.java`-Datei wird eine `.class`-Datei
- Die sog. Java Virtual Machine interpretiert diese Dateien und führt sie aus - Aufruf mit `java`

- Eine große Java-Anwendung wird in ein Archiv gepackt (`.jar` oder `.war`), das viele `.class`-Dateien enthält

Ein kleines Beispiel

- Wir erstellen ein Programm namens **HalloWelt**:

HalloWelt.java

```
import java.io.*;

public class HalloWelt {
    public static void main(String args[]) {
        System.out.println("Hallo Welt!");
    }
}
```

- Wir lassen es übersetzen:
  `javac HalloWelt.java`
- Nun können wir es ausprobieren:
  `java HalloWelt`