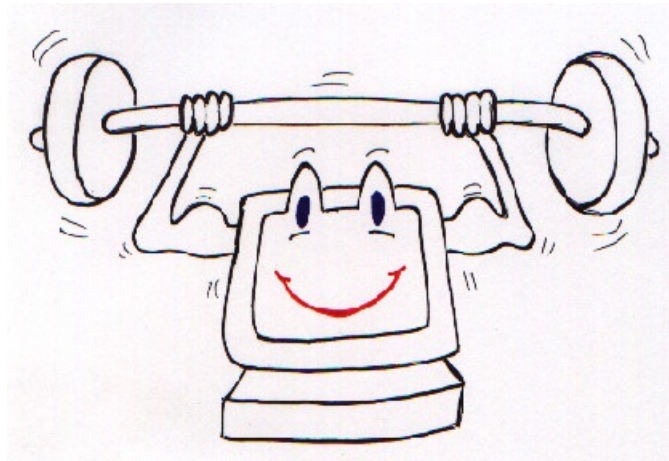


ENTWURF



Projektmitglieder

Erhard Dinhobl

Martin Reiterer

Autor: Martin Reiterer

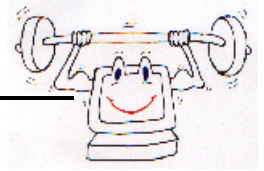
Erhard Dinhobl

Update: 12.05.2004

Version: 1.0

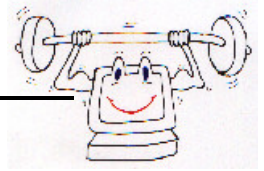
Gedruckt: 12.05.2004

Dateiname: Entwurf.doc



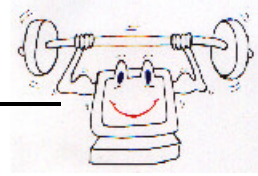
Inhaltsverzeichnis

1	Klassendiagramme	3
1.1	Package: programmiertrainer	4
1.1.1	Klassenbeschreibung	5
1.2	Package: ExampleBuilder	6
1.2.1	Klassenbeschreibung	7
2	Zuständigkeitsübersicht	8
2.1	ProgrammierTrainer	8
2.2	ExampleBuilder	9
3	Pflichtenheft-Funktionalitätsumsetzung	10
3.1	ProgrammierTrainer	10
3.2	ExampleBuilder	11
4	Informationen zu JavaDoc	11
5	Programmiernormen	12
5.1	Programmierspezifische Richtlinien	12
5.2	Interfacespezifische Richtlinien	12

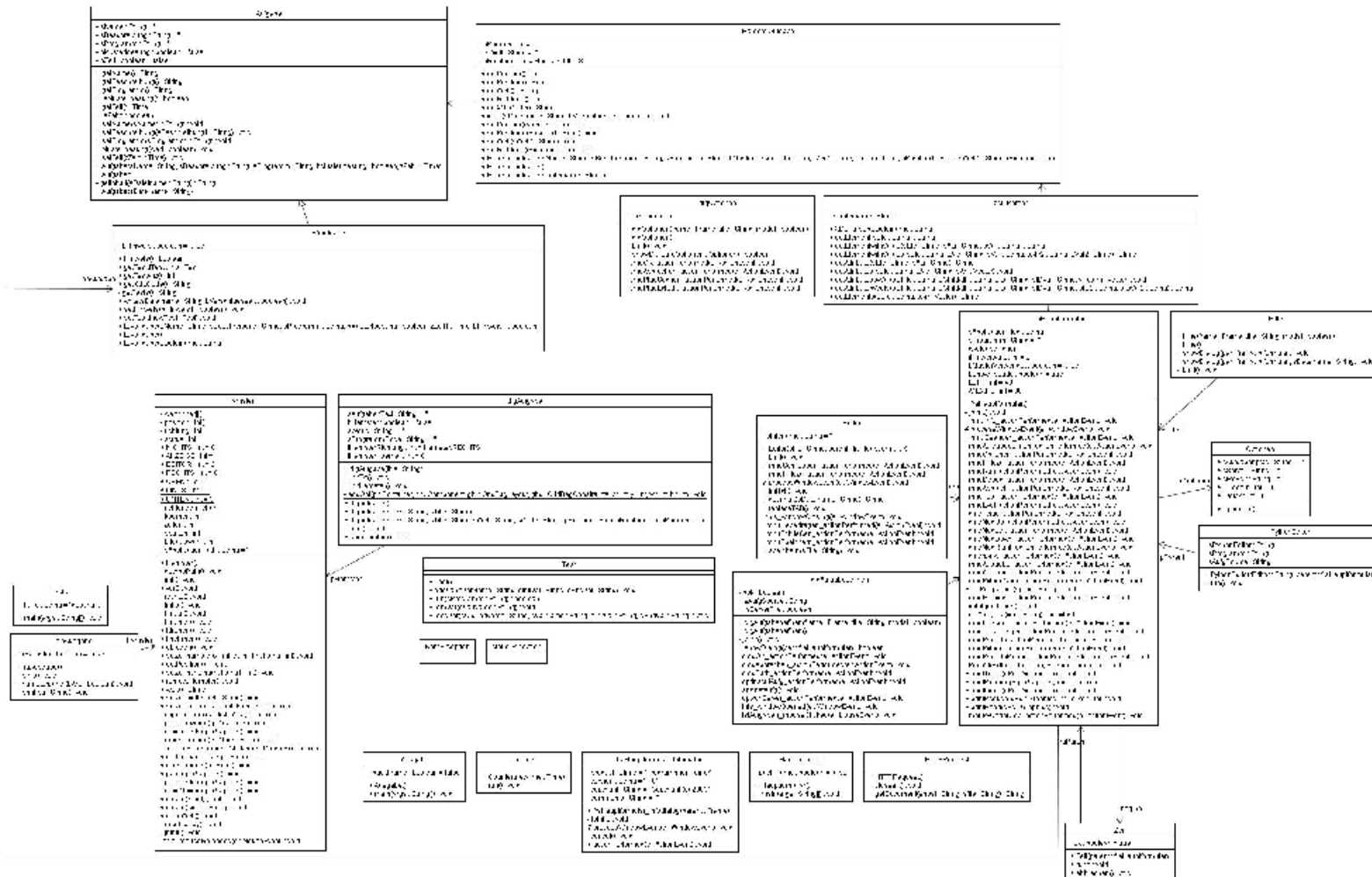


1 Klassendiagramme

In den folgenden Kapiteln werden zu jedem Package aus denen das Programm besteht ein Klassendiagramm dargestellt und zu jeder Klasse die Hauptfunktionen erläutert. Im Diagramm werden kleiner Klassen und Hilfsklassen nicht angezeigt.

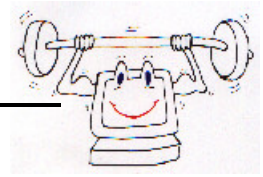


1.1 Package: programmiertrainer

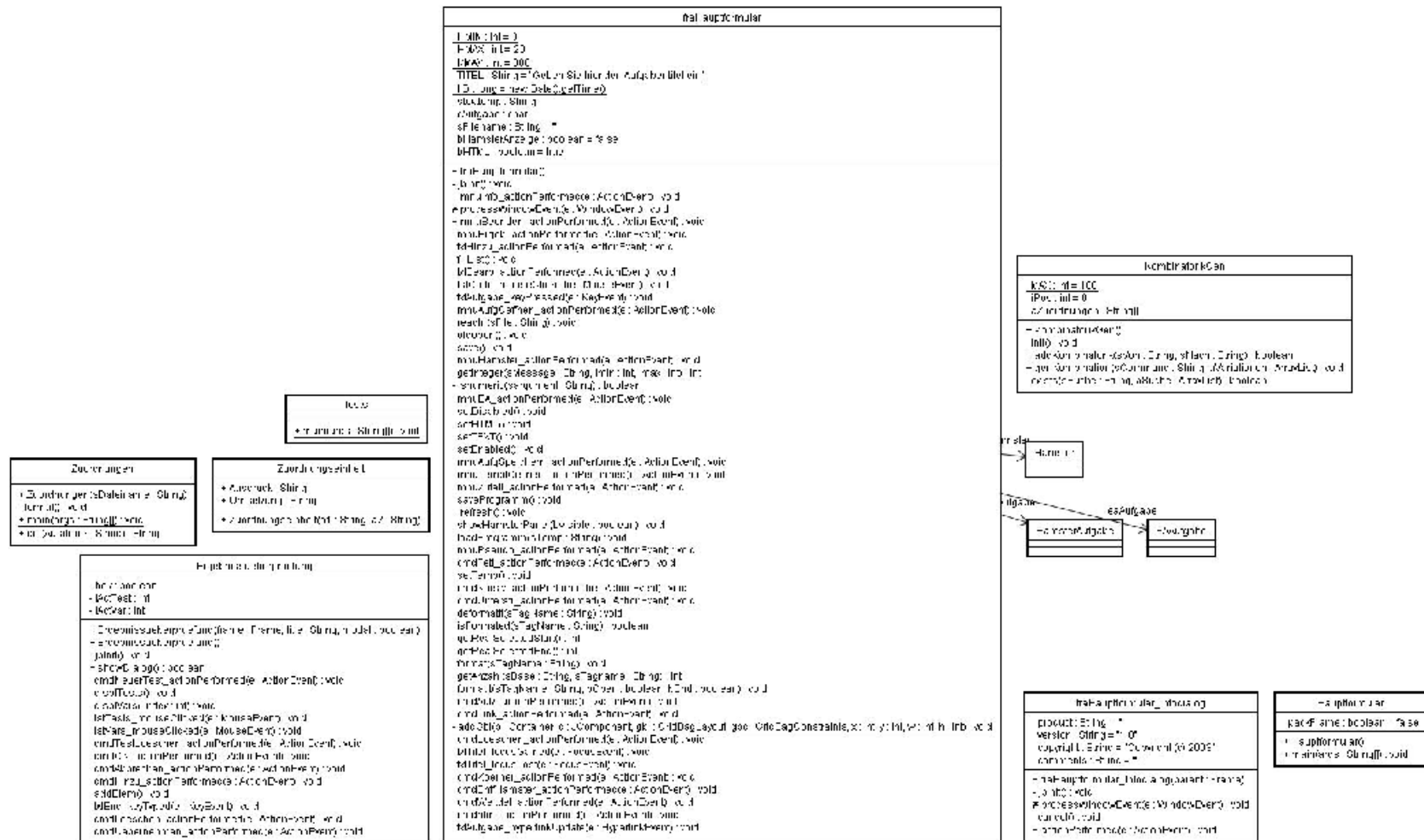


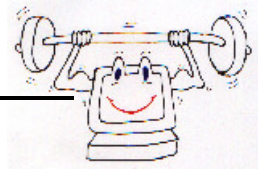
1.1.1 Klassenbeschreibung

Die Klassenbeschreibungen zum ProgrammierTrainer wurden mittels JavaDoc erstellt und sind im Projektordner verfügbar.



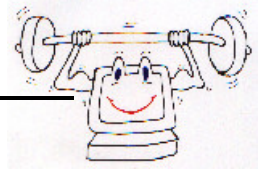
1.2 Package: ExampleBuilder





1.2.1 Klassenbeschreibung

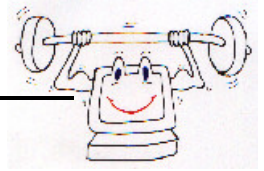
Die Klassenbeschreibungen zum ExampleBuilder wurden mittels JavaDoc erstellt und sind im Projektordner verfügbar. Im Diagramm werden kleiner Klassen und Hilfsklassen nicht angezeigt.



2 Zuständigkeitsübersicht

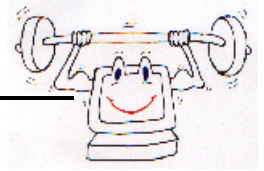
2.1 ProgrammierTrainer

Klasse	Implementiert von	Start - Ende
dlgOptionen	ED	100%
dlgAufgabe	MR	100%
fraHauptformular_Infodialog	ED	100%
XMLParser	ED	100%
Editor	MR	100%
Hauptformular	ED	100%
Ausgabe	ED	100%
HTTPRequest	ED	100%
dlgAufgabeoeffnen	ED	100%
fraHauptformular	ED; MR	/
→ Interface	ED	100%
→ Menü: Datei	ED; MR	/
→ Aufgabe öffnen	ED	100%
→ Python-Code	MR	100%
→ Drucken	MR	100%
→ Beenden	ED	100%
→ Menü: Fenster	ED; MR	/
→ Ausgabefenster	ED	100%
→ Aufgabe	MR	100%
→ Optionen	ED	100%
→ Python-Code anzeigen	MR	100%
→ Zeit anzeigen	MR	100%
→ Menü: Interpreter	ED	100%
→ Menü: Hilfe	MR	100%
fraAusgabe	ED	100%
Hilfe	ED	100%
Aufgabe	MR	100%
Zeit	MR	100%
EAAufgabe	MR	100%
HamsterAufgabe	MR	100%
Test	ED	100%
Hamster	MR	100%
Counter	MR	100%



2.2 ExampleBuilder

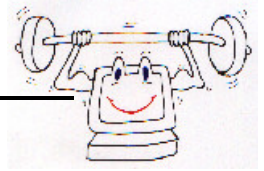
Klasse	Implementiert von	Start - Ende
fraHauptformular	ED; MR	/
→ Interface	MR	100%
→ Menü: Datei	ED; MR	/
→ Neu	MR	100%
→ Aufgabe öffnen	MR	100%
→ Aufgabe speichern	MR	100%
→ Template öffnen	MR	100%
→ Testen	ED	100%
→ Beenden	MR	100%
→ Menü: Aufgabe	ED; MR	/
→ Ergebnisüberprüfung	ED	100%
→ Pseudo-Code erzeugen	MR	100%
→ Zufalls-Anweisungen erzeugen	MR	100%
→ Hinweise verfügbar	ED	100%
→ Musterlösung verfügbar	ED	100%
→ Menü: Dienstprogramme	ED	100%
→ Menü: Hilfe	ED	100%
fraHauptformular_Infodialog	ED	100%
Hauptformular	ED	100%
Zuordnungen	MR	100%
Ergebnisueberpruefung	ED	100%
KombinatorikGen	MR	100%



3 Pflichtenheft-Funktionalitätsumsetzung

3.1 ProgrammierTrainer

Funktionen	Erfüllt durch folgende Klasse(n)
4.1.1 Hauptformular	fraHauptformular Hauptformular XMLParser
4.1.1.1 Eingabe/Ausgabe-Aufgabe	Aufgabe EAAufgabe
4.1.1.2 Hamster-Aufgabe	Aufgabe HamsterAufgabe
4.1.1.3.1.1 Aufgabe öffnen	dlgAufgabeoeffnen
4.1.1.3.1.3 Drucken	fraHauptformular
4.1.1.3.2.3 Optionen	dlgOptionen
4.1.1.3.2.4 Python-Code anzeigen	fraHauptformular
4.1.1.3.2.5 Zeit anzeigen	fraHauptformular →Zeit
4.1.1.3.3.1 Run	fraHauptformular
4.1.1.3.3.2 Debug	fraHauptformular
4.1.1.3.3.4 Test	fraHauptformular →Test
4.1.1.3.4.3 Hinweis	fraHauptformular
4.1.1.3.4.4 Lösung	fraHauptformular
4.1.2 Aufgabedialog	dlgAufgabe Hamster
4.1.3 Ausgabefenster	Ausgabe fraAusgabe Hamster
4.1.4 Aufgabe-Öffnendialog	dlgAufgabeoeffnen HTTPRequest XMLParser
4.1.5 Optionendialog	dlgOptionen
4.1.6 Editor	fraEditor Editor
4.1.7 Hilfedialog	Hilfe
4.1.8 Infodialog	fraHauptformular_Infodialog



3.2 ExampleBuilder

Funktionen	Erfüllt durch folgende Klasse(n)
4.2.1 Hauptformular	Hauptformular fraHauptformular Aufgabe EAAufgabe HamsterAufgabe Hamster XMLParser
4.2.1.1.1.1 Neu	fraHauptformular
4.2.1.1.1.5 Testen	fraHauptformular Ausgabe fraAusgabe
4.2.1.1.2.2 Pseudo-Code erzeugen	fraHauptformular Zuordnungen
4.2.1.1.2.3 Zufalls-Anweisungen erzeugen	fraHauptformular KombinatorikGen
4.2.1.1.2.6.1 Verzeichnis Indizieren	fraHauptformular
4.2.1.1.3.1 Hilfe	Hilfe
4.2.2 Ergebnisüberprüfungsdialog	Ergebnisueberpruefung Test
4.2.3 Infodialog	fraHauptformular_Infodialog

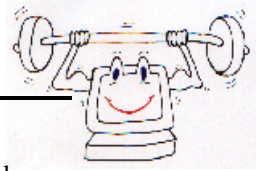
4 Informationen zu JavaDoc

Vor einer Methode können zu Dokumentationszwecken Informationen bezüglich ihrer Beschreibung angegeben werden. Diese Informationen werden dann mit Hilfe von JavaDoc (Software) in eine Dokumentation die aus HTML-Dateien besteht überführt.

Um eine Beschreibung für eine Methode anzugeben:

```
/**
 * Hier steht eine Beschreibung zur Methode methodel.
 * Sie kann auch mehrere Zeilen beinhalten.
 * Nach der Beschreibung für die Methode können auch
 * Informationen die die Software JavaDoc „versteht“
 * angegeben werden. Sie werden jeweils mit einem @
 * eingeleitet
 *
 * @author Name Des Authors
 * @version 1.4
 * @param x Kommentar zum Parameter x (1. Parameter)
 * @param y Kommentar zu y
 * @return Beschreibung zum Returnwert
 *
 */

public int methodel(int x, int y) {
    ...
}
```



Um eine Beschreibung für eine Klasse anzugeben (muss nach den import-Anweisungen und vor der class-Anweisung stehen):

```
import java.io.*;

/**
 * Beschreibung für die Klasse X.
 * Und hier können ebenfalls Informationen angegeben werden.
 *
 * @author Name Des Authors
 * ...
 */

class X {
    ...
}
```

5 Programmiernormen

Im Allgemeinen gelten die Java-Programmiernormen.

5.1 Programmierspezifische Richtlinien

- Jegliche Pfade, d.s. URLs und lokale Pfade, die im gesamten Projekt verwendet werden, verwenden als File-Separator nur einen „/“.
- Jegliche Pfade, d.s. URLs und lokale Pfade, die im Hauptformular in Variablen gespeichert werden, sollen am Schluss keinen „/“ enthalten.
- Jede „größere“ Methode soll mit JavaDoc Anweisungen versehen werden.
- Jede Klasse soll mit JavaDoc Anweisungen versehen werden.

5.2 Interfacespezifische Richtlinien

- Tooltip-Text: Tooltip-Texte sollen Informationen zum Steuerelement enthalten, aber nicht mit einem Punkt abgeschlossen werden.
- Alle Steuerelemente eines Formulars sollen vom Formularrand 5 Pixel entfernt liegen.

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.