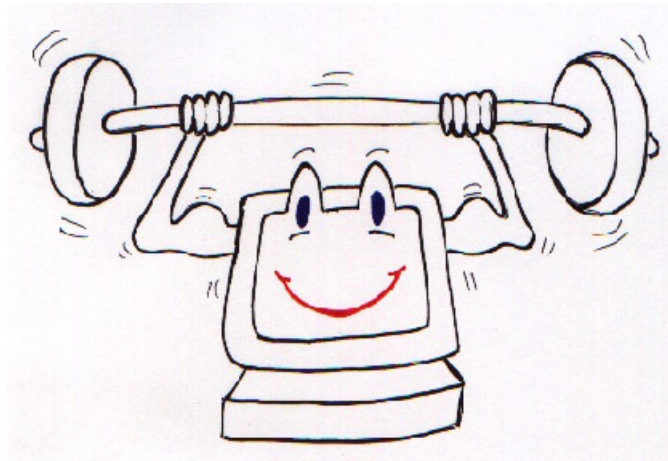


ANFORDERUNGSDOKUMENT



Projektmitglieder

Erhard Dinhobl

Martin Reiterer

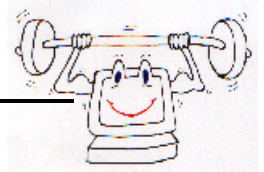
Autor: Martin Reiterer

Update: 28.01.2004

Version: 1.0

Gedruckt: 12.05.2004

Dateiname: Anforderungsdokument.doc



Inhaltsverzeichnis

1	Ausgangssituation	3
2	Zielbestimmung	3
3	Produkteinsatz.....	3
4	Produktfunktionen	4
4.1	Hauptformular	4
4.2	Tool zum Erstellen der Aufgabe	4
4.3	Variablen Input	4
4.4	Optionen von ProgrammierTrainer	4
4.5	Aufgaben öffnen	4
4.6	Interpreter.....	4
4.7	Ergebnisüberprüfung.....	4
4.8	Texteditorfenster	5
4.9	Produktdaten	5
5	Weitere Produktanforderungen	5
5.1	Leistungsanforderungen.....	5
5.2	Qualitätsanforderungen	5



1 Ausgangssituation

Viel wichtiger als die Syntax einer Programmiersprache zu beherrschen ist, dass man sich den Ablauf eines Programms vorstellen kann. Für Programmieranfänger ist es zu Beginn oft schwer sich auf das abstrakte Denken, welches zum Programmieren notwendig ist einzustellen.

Im Wesentlichen besteht der Hauptaufwand beim Schreiben eines Programms im Analysieren der Aufgabenstellung und im Formulieren eines geeigneten Algorithmus.

Derzeit besteht das Problem, dass Programmieranfänger, bevor sie ihr erstes Programm schreiben, die Syntax einer Programmiersprache erlernen müssen. Dies ist ein Grund dafür, dass beim ProgrammierEinstieg die Motivation der Schüler sinkt, da sie sehr lange kein Ergebnis sehen.

2 Zielbestimmung

Mithilfe des ProgrammierTrainers soll programmiersprachenunabhängig das Erstellen von Algorithmen in Form von Anweisungslisten geübt werden können. Gleichzeitig übt der Programmieranfänger logisches Denken und das Analysieren der Aufgabenstellung. Dies alles sind Fertigkeiten, die für das Programmieren notwendig sind.

Mit dem ProgrammierTrainer ist es möglich, sofort mit der Formulierung von Algorithmen zu beginnen. Es muss auch keine Programmiersprache erlernt werden, da die möglichen Anweisungen beliebigen Text enthalten können. Schüler wählen die benötigten Anweisungen, welche zum formulieren eines Algorithmus benötigt werden, aus einer vorgegebenen Liste mit Anweisungen aus und können so durch Pseudo-Code-Anweisungen den Programmablauf bestimmen.

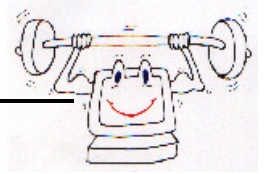
Da beim ProgrammierTrainer die möglichen Anweisungen vorgegeben sind, entfallen auch die Probleme mit Syntaxfehlern (Tippfehler) beim Formulieren der Algorithmen. Trotzdem soll es während dem Erstellen eines Algorithmus möglich sein, den momentan implementierten Programmcode auszuführen.

Weiters soll im ProgrammierTrainer eine Funktionalität zur Erstellung von Aufgabestellungen implementiert werden. Die Lösung bzw. das Ergebnis der Aufgabenstellung darf von den Schülern nicht eingesehen werden können, deshalb soll die Datei mit der erstellten Aufgabestellung verschlüsselt werden.

Ein weiteres Ziel ist es, den Programmiertrainer Systemunabhängig zu machen. D.h. er soll sowohl unter Windows wie auch unter anderen Betriebssystemen laufen.

3 Produkteinsatz

Der ProgrammierTrainer wird für die Programmieranfänger in den ersten Klassen der HTL, Abteilung EDV und Organisation verwendet, kann aber auch für alle anderen Programmieranfänger eingesetzt werden.



4 Produktfunktionen

4.1 Hauptformular

Der Benutzer hat 2 Listenfelder vor sich.

Das eine Listenfeld beinhaltet die vorgegebenen Anweisungen, welche zur Erstellung eines Algorithmus benötigt werden. Außerdem sind zur Verwirrung noch andere Anweisungen, die zur Lösung der Aufgabenstellung nicht benötigt werden, in die Anweisungsliste hineingemischt.

Die zweite Liste ist die Ergebnisliste. Sie dient zur Erstellung des Algorithmus. In diese Liste muss der Benutzer die benötigten Anweisungen von dem ersten Listenfeld Hineinverschieben und in die richtige Reihenfolge bringen. Außerdem kann der Benutzer durch einrücken Anweisungsblöcke deklarieren.

Während der Benutzer sein Programm erstellt, kann er auch jederzeit das Interpreterfenster aufrufen um das Ergebnis seiner Arbeit zu überprüfen.

4.2 Tool zum Erstellen der Aufgabe

Mithilfe des Aufgabeneditors soll es möglich sein eigene Aufgabenstellungen zu formulieren. Die erstellten Aufgabestellungen werden anschließend verschlüsselt und in einer Datei abgespeichert.

Außerdem soll es möglich sein, die erstellte Musterlösung auszuführen und auf etwaige Syntaxfehler zu überprüfen. Dazu wird wiederum das Ausführfenster verwendet.

4.3 Variablen Input

Interface zum Einlesen der Ausgangswerte für die Startvariablen.

4.4 Optionen von ProgrammierTrainer

Allgemeine Einstellungen zum Programmiertrainer

4.5 Aufgaben öffnen

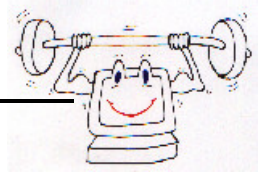
Interface zum Öffnen der Aufgabestellungen.

4.6 Interpreter

Im Interpreterfenster läuft der vom Benutzer erstellte Algorithmus ab und gibt die Ausgaben in einer Liste aus.

4.7 Ergebnisüberprüfung

In der Ergebnisüberprüfung kann die vom Benutzer erstellte Lösung auf ihre Richtigkeit überprüft werden.



4.8 Texteditorfenster

Im Texteditorfenster kann man direkt Python-Code programmieren.

4.9 Produktdaten

Die für das Programm benötigten Aufgabestellungen werden in Dateien zentral auf einem Server gespeichert. Die Speicherung der Programmeinstellungen findet auf Dateien auf dem lokalen Datenträger statt.

5 Weitere Produktanforderungen

5.1 Leistungsanforderungen

In 90% aller Fälle sollte die Antwortzeit beim Laden einer Aufgabenstellung nicht länger als zwei Sekunde dauern.

5.2 Qualitätsanforderungen

Produktqualität	sehr gut	gut	normal	nicht relevant
Zuverlässigkeit	x			
Erlernbarkeit	x			
Laufzeiteffizienz		x		
Portabilität	x			
Benutzerfreundlichkeit	x			
Datensicherheit	x			
Wartbarkeit		x		

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.