

Ausarbeitung für Modul GUI

Inhaltsverzeichnis

Die Vor- und Nachteile von in der Verwendung von C++ (WIN32) und Java (Swing und AWT) Programmen bei der Verwendung von Graphischen Benutzeroberflächen.....	2
Vor und Nachteile in Verwendung von Swing gegenüber AWT bei der Java Entwicklung	3
Begründung für die Entscheidung von Swing.....	3
Funktionsweise des C++ Programms.....	4
Funktionsweise des Java Programms.....	5

Die Vor- und Nachteile von in der Verwendung von C++ (WIN32) und Java (Swing und AWT) Programmen bei der Verwendung von Graphischen Benutzeroberflächen.

Zur Programmierung von grafischen Benutzeroberflächen gibt es verschiedene Programmiersprachen. Diese Programmiersprachen bieten verschiedene APIs an mit denen die Oberflächengestaltung erleichtert werden kann. Zum Programmieren mit C/C++ unter Windows kann man die hauseigene WIN32 API verwenden. Unter Java gibt es verschiedene Bibliotheken, um grafische Oberflächen zu implementieren. Die beiden offiziellen APIs sind Swing und AWT.

Vorteile der WIN32 API unter Windows sind die direkte Integration ins Betriebssystem. So werden die Funktionsaufrufe direkt ans Betriebssystem weitergeleitet und verarbeitet. Dies hat einen geringen Overhead und da mit einem geringeren Ressourcenverbrauch zur Folge. Die Programme werden der visuellen Oberfläche des Betriebssystems angepasst. So kann ein älteres Programm auf einem modernen Betriebssystem ausgeführt immer noch zeitgemäß wirken. Zudem sehen die Dialoge, wie z.B., das Öffnen eines Verzeichnisses, auf mit WIN32 API geschriebenen Programm immer gleich aus. Somit hat der Nutzer eine geringere Einarbeitungsphase in das Programm. Zudem werden Programme, die in der WIN32 API geschrieben sind, nativ vom Betriebssystem unterstützt. Sie werden ausgeführt ohne, dass zusätzliche Software auf dem System installiert werden muss.

Der Nachteil dieser API ist, dass diese auf Windows-Systeme beschränkt ist. Eine Portierung würde einen erheblichen Aufwand bedeuten. Um ein C/C++ Programm mit grafischer Oberfläche für mehrere Betriebssysteme entwickeln zu können muss man deswegen auf andere Grafik APIs, wie z.B. GTK zurückgreifen. Es erfordert zu dem einen höheren Programmieraufwand die Oberfläche an verschiedene Auflösungen und Größen der Anzeigegeräte anzupassen Da es keine Funktionen gibt eine automatische Ausrichtung zu lassen.

Bei Windows-Systemen mit einer festen Auflösung und niedrigen Leistungen haben, ist es daher von Vorteil, wenn diese in C/C++ geschrieben werden.

Ein großer Vorteil grafischer Benutzeroberflächen in Java mit AWT oder Swing zu programmieren ist, dass diese Programme plattformunabhängig funktionieren. Sie werden ohne große Veränderungen auf jedem Betriebssystem laufen können auf dem eine JVM (Java Virtuelle Maschine) installiert ist. Dies ermöglicht das Entwickeln und Kompilieren für mehrere Plattformen gleichzeitig ohne verschiedene Codebasen zu unterhalten. Die API Swing enthält zu dem Layoutmanager, die es ermöglicht Steuerelemente dynamisch anzulegen. So passt sich das Layout der unterschiedlichen Auflösungen und Fenstergröße an.

Der Nachteil dieser Betriebssystemunabhängigkeit ist allerdings, dass auf jedem System eine JVM installiert werden muss. Es wird also nicht auf einem System ohne zusätzliche Installationen laufen. Durch diese virtuelle Maschine entsteht auch ein großer Ressourcen Overhead.

Bei einem Programm, das auf verschiedenen Betriebssystemen laufen soll, ist es von Vorteil dieses auf Java Basis zu programmieren.

Die Programmiersprachen haben beide Vor und Nachteile und werden deswegen anwendungsspezifisch benutzt. So können Programme die unter C/C++ mit der WIN32 API geschrieben werden, auf sehr ressourcenarm Computer laufen wie z.B. Kassensystemen oder Industrieanlagen. Desktop-Apps auf Java Basis geschrieben unterstützen eine größere Anzahl an Betriebssystemen. Dabei kann gleichzeitig der Ressourcen Hunger durch die stärkeren Geräte ausgeglichen werden.

Vor und Nachteile in Verwendung von Swing gegenüber AWT bei der Java Entwicklung

Wenn man grafische Oberflächen in einem Java Programm entwickeln möchte, gibt es anderen Grafik APIs die beiden in der Standardbibliothek enthaltenden Lösungen: AWT und Swing.

AWT ist die ältere dieser beiden APIs und wurde im Jahre 1996 mit Java 1.0 eingeführt. AWT steht für Abstract Window Toolkit. Es benutzt die vom Betriebssystem vorgegebenen Komponenten. Dadurch passt es sich auf den unterschiedlichen Betriebssystemen dem Look des entsprechenden Betriebssystems an. Da die Betriebssysteme das Zeichnen der Komponenten übernehmen, gibt es dadurch einen geringeren Ressourcenverbrauch auf Seiten der Java Laufzeitumgebung. Ein großer Vorteil ist die Flexibilität auf verschiedenen Displaygrößen. So hat AWT einen Fenstermanager der dynamisch befüllt werden kann. AWT ist zudem threadsicher der Programmierer muss sich bei einem Programm mit mehreren Threads keine Sorgen um die Zugriffe auf das GUI API machen.

Da AWT die vom Betriebssystem bereitgestellten Klassen nutzt gibt es hier allerdings ein Kompatibilitätsproblem. Denn nicht alle Betriebssysteme stellen die gleichen Komponenten an Oberflächen zu Verfügung. Es wird also ein gewisser Zeitaufwand benötigt, um das Programm für alle Systeme benutzbar zu machen. Das AWT die Betriebssystemkomponenten nutzt kann sich für einen Visuellen Diesiger der Oberfläche auch zu Nachteil auswirken. Zudem gibt es in AWT spezielle Komponenten nicht die nur mit erheblichen Programmieraufwand zu bewältigen sind, wie z.B. Tooltips und Drag & Drop Funktion.

Swing wurde im Jahr 1998 mit Java 1.2 eingeführt und nutzt keine vom Betriebssystem bereitgestellten Komponenten. Es benutzt sein eigenes Set aus Komponenten, die damit auf allen Betriebssystemen verfügbar sind. Es erfordert also weniger Anpassungen zwischen Betriebssystemen. Swing hat zu den kompletten AWT Funktionalitäten auch noch weitere nützliche Features, wie z.B. Tooltips und Drag & Drop Funktion. Man kann diese auch mit ein wenig Aufwand in den jeweiligen Look des Betriebssystems bringen. Da Swing nicht die Systemkomponenten nutzt ist es darauf angewiesen, dass die Implementierung für Swing Komponenten eine Hardware Beschleunigung bietet. In aktuellen Betriebssystemen ist es überall vorhanden der Fall. Lediglich bei älteren Implementierungen und vereinzelt Ausnahmen kann die hardwareseitige Grafikbeschleunigung fehlen. Da Swing im Vergleich zu AWT deutlich weiterentwickelt ist und aufgrund der Design Entscheidungen, die getroffen wurden, ist Swing in aktuellen Implementationen meistens schneller.

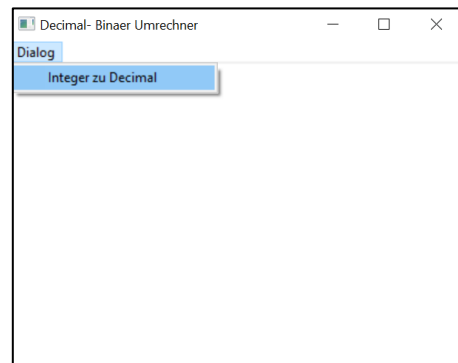
Einer der Entscheidungen ist es, Teile des Programms nicht threadsicher zu gestalten, um eine größere Performance zu erreichen. Diese Eigenschaft muss bei Programmen mit mehreren Threads vom Entwickler berücksichtigt werden.

Begründung für die Entscheidung von Swing

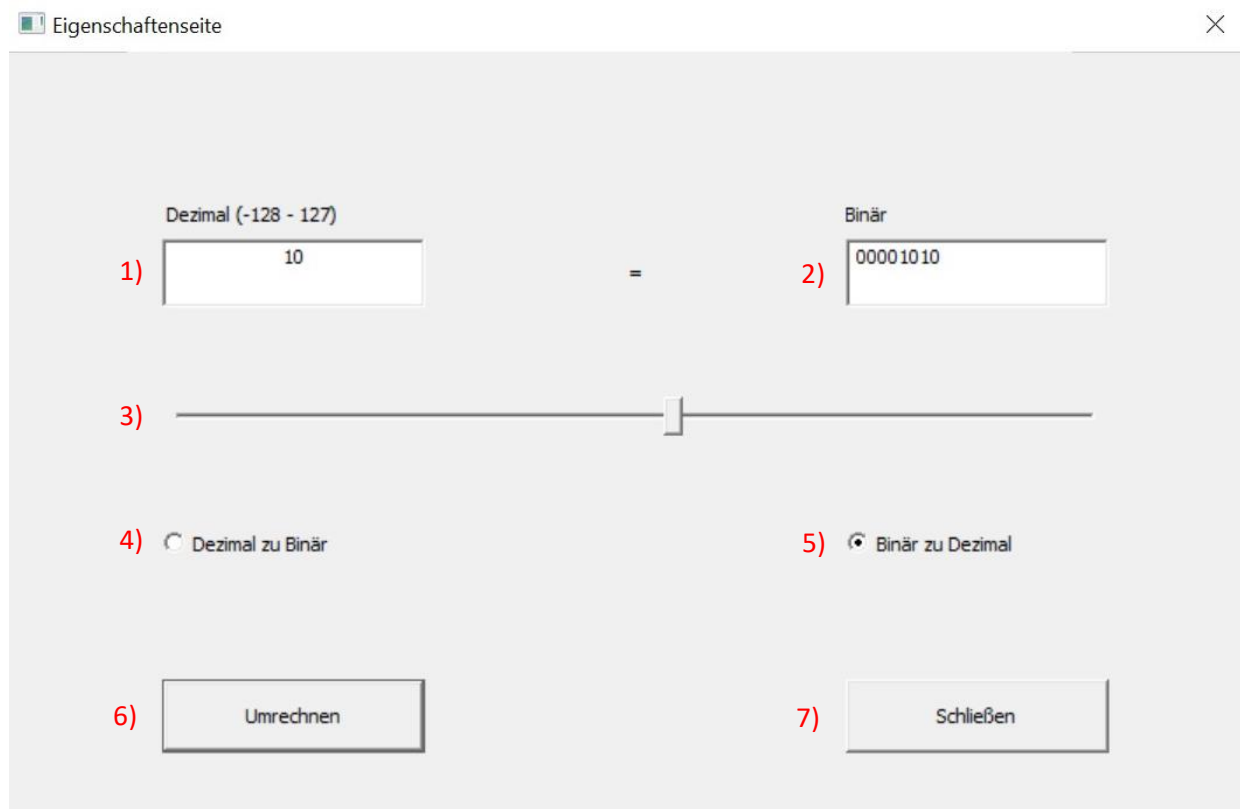
Meine Entscheidung das Programm mit dem Swing API zu programmieren fußt darauf, dass ich das Zielsystem nicht kenne. Mit Swing kann ich ein Programm kreieren was auf dem Ziel Betriebssystem genauso aussieht wie ich es auf meinem Betriebssystem designe. Da mein Programm nicht auf mehreren Threads läuft muss ich mir um das Synchronisieren keine Gedanken machen. Die Drag Funktion des Rechtecks die ich in meinem Programm erstelle, ist unter AWT nur mit erheblichem Mehraufwand zu erzeugen. Es ist daher einfacher das Swing API für diese Art Aufgabenstellung zu nutzen.

Funktionsweise des C++ Programms

Nach dem Öffnen des Programms ist im Menü unter dem Punkt „Dialog“, der Unterpunkt „Integer zu Dezimal“ zu finden.



Durch Klicken dieses Punktes öffnet sich dieses Dialogfenster.



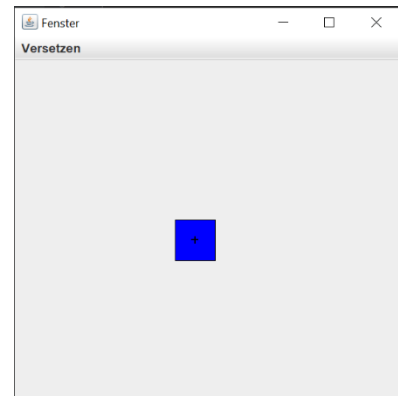
In dem Dialogfenster befinden sich zwei Textfelder. Im ersten werden Dezimalzahlen eingetragen die im Wertebereich -128 zu 127 liegen dürfen. Im zweiten können Binärzahlen eingetragen werden, die höchstens eine Länge von 8 Ziffern haben. Dabei wird hier die Repräsentation der Bits in signed angegeben.

Der Wert im „Binär“ Feld kann zudem über den Slider 3) unter den Text Feldern manipuliert werden. Die Richtung des Umrechnens kann durch die beiden Radiobuttons „Dezimal zu Binär“ 4) oder „Binär zu Dezimal“ 5) beeinflusst werden.

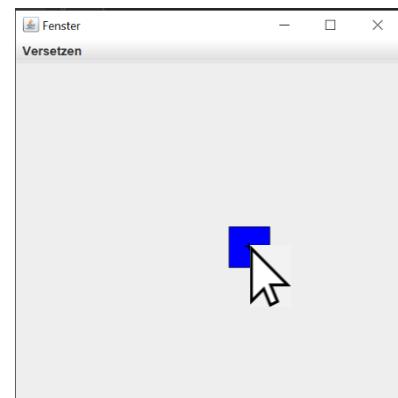
Dies geschieht aber erst wenn der Button „Umrechnen“ 6) dazu gedrückt wird. Der Slider passt sich dem Wert an. Bei falschen Angaben werden entsprechende Fehlermeldungen ausgegeben. Durch den „Schließen“ Button 7) kann dieses Dialogfenster geschlossen werden ohne dass sich das Programm beendet.

Funktionsweise des Java Programms

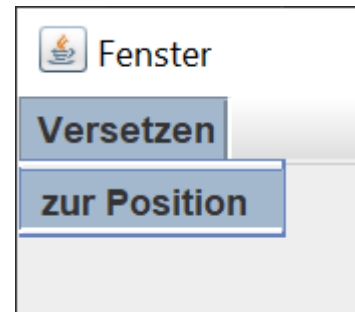
Nach dem Ausführen des Programms erscheint in dem Hauptfenster ein blaues Rechteck. Durch das Anklicken des Rechtecks mit der rechten Maustaste verschiebt sich das Rechteck so dass der Mittelpunkt auf der Spitze des Cursors positioniert wird. Bei gedrückter rechter Maustaste kann das Rechteck über das Fenster gezogen werden und beim Loslassen an einer beliebigen Stelle positioniert werden.



Durch Doppelklicken der rechten Maustaste wird das Rechteck an die Ursprungsposition zurückgesetzt. Durch Doppelklicken der linken Maustaste wird die Drag and Drop Funktion deaktiviert.



Wenn man den Menü Punkt „Versetzen“ in der Menü Bar aufruft ist in diesem Fall ein Eintrag „Zur Position“ vorhanden.



Dieser öffnet einen Dialog, in den man die Position des Rechtecks festlegen kann nach dem Bestätigen wird das Rechteck an dieser Position zu finden sein. Die Textfelder sind so formatiert, dass die eingegebenen Werte in gültige Zahlen umgeändert werden. Wenn keine Werte eingegeben werden wird das Fenster nicht auf die Eingabe reagieren.

