

Inhaltsverzeichnis

1	Einführung	1
2	Webanwendung für Energiemanagement	3
2.1	Schnittstelle von IT und Energietechnik	3
2.2	Visuelle Java Entwicklungsumgebung XDEV 4 für die Schnittstelle von IT und Energietechnik	4
2.2.1	Graphical User Interface-Builder (GUI-Builder)	5
2.2.2	Generieren von Formularen	6
2.2.3	Verteilen von Formularen	8
2.2.4	Gestaltung der Datenbank <i>energiemanagement</i> mit Hilfe von MySQL und Xampp	8
2.2.5	Drag-and-drop zum Data Binding	9
2.2.6	ER-Diagramm	11
2.2.7	Erstellen von Frontend-Anwendungen	13
2.3	Zusammenfassung	17
	Literatur	17
3	Anwendung der Architektur von Swing in der Entwicklung der grafischen Oberfläche	19
3.1	Benutzeroberfläche der Energiemanagement-Anwendungen des Asynchronmotors	19
3.1.1	Model-View-Controller (MVC) Konzept	19
3.1.2	Implementierung der Klasse <i>XdevWindow</i>	20
3.1.3	Datenquelle einer Virtuelle Tabelle	30
3.1.4	Entwicklung von Administrationsfenster	33

3.2	Event-Handlings	35
3.2.1	Überblick über <i>EventHandlerDelegate</i>	35
3.2.2	Delegation <i>Event Model</i> zum Kommunikationsmodell zwischen <i>Event Sources</i> und <i>Event Listeners</i>	36
3.3	Überblick über Adapterklassen in dem Bezug auf die Implementierung des Interface <i>Event-Listeners</i>	36
3.3.1	Implementieren von Schnittstellen wie <i>FormularAdapter</i> und <i>WindowAdapter</i>	36
3.3.2	Definition der Methode ohne funktionellen Code	37
3.4	Anwendung anonymer Klassen für die Ereignisbehandlung	37
3.4.1	Implementierung von <i>EventListener</i>	37
3.5	Überblick über elementare Ereignisarten: Window-Event oder Low-Level-Event	38
3.5.1	Anwendung eines Window-Events	38
3.5.2	Implementierung des Interfaces <i>WindowListener</i>	39
3.5.3	XDEV 4 GUI-Komponente zum Implementieren der Schnittstellen durch lokale Klassen	39
3.6	Überblick über nichtstatischen Initialisierer der inneren Klasse	40
3.6.1	Erzeugung von Objektinstanzen innerhalb der Klasse <i>XdevWindow</i>	40
3.6.2	Prinzip der Kapselung in der objektorientierten Programmierung	41
3.6.3	Überlagerung bei der objektorientierten Programmierung ..	42
3.6.4	Einsatz von innerer anonymer Klasse für GUI-Anwendung ..	43
3.7	Überblick über Anordnen von Komponenten in Dialogelementen mit Hilfe von <i>Layout-Manager</i>	43
3.7.1	Das Erzeugen des Fensters mit Hilfe von GUI- Dialogelementen	43
3.7.2	Realisierung eines <i>GridBagLayouts</i>	44
3.8	Implementierung und Instanziierung der anonymen Klasse <i>Beginn</i> mit Hilfe der Lamda-Ausdrücke	45
3.8.1	Übergabe von Parametern eines <i>funktionalen Interface</i> ...	45
3.8.2	Implementieren funktionaler Schnittstellen mit Hilfe von Lamda-Ausdrücken	45
3.9	Zusammenfassung	46
	Literatur	47

4	Entwicklung von Asynchronmotoren-Anwendungen mit Hilfe von MySQL-Datenbank	49
4.1	Entwicklung von grafischen Oberflächen in Java	50
4.1.1	Tools für die Entwicklung	50
4.1.2	Toolset von Java XDEV 4 Framework	51
4.2	Daten vom Energiemanagement des Asynchronmotors komfortabel mit MySQL verwalten	52
4.2.1	Anwendung von den Datenbanktabellen und Java-Objekten in den elektrischen Antrieben	52
4.2.2	Entwicklung von Asynchronmotoren-Anwendungen mit MySQL	53
4.2.3	Programmierung von Datenbank „energiemanagement“ mit Java XDEV Framework	56
4.3	ER-Diagramm für die Energiemanagement-Anwendungen des Asynchronmotors	57
4.3.1	Visualisierung des Datenmodells mit Hilfe des ER-Diagramms	57
4.3.2	Anwendung von statischer Methode in dem ER-Diagramm	58
4.4	Datenmodell der Datenbank „energiemanagement“	59
4.5	IT-Lösungen für energieeffiziente Optimierung des Asynchronmotors	61
4.5.1	Energieeffizienter Asynchronmotor	61
4.5.2	Virtuelle Tabelle des Frameworks Java XDEV 4 für den energieeffizienten Asynchronmotor	62
4.6	Zugriff auf Datenbanktabelle mit Hilfe objektorientierter Programmierung:	63
4.6.1	Schnittstelle von IT und Energieeffizienz mit Hilfe der objektorientierten Programmierung	63
4.6.2	Objektorientierte Programmierung der energieeffizienten Optimierung des Asynchronmotors	71
4.6.3	Überblick über Membervariablen in Bezug auf energieeffiziente Optimierung des Asynchronmotors	72
4.7	Zusammenfassung	73
	Literatur	73

5 IT-Lösungen für den Drehstromantrieb: Anwendung vom Java XDEV 4 Framework in den Drehzahlstellungen	75
5.1 Anwendung der IT in Gleichstromfrequenzumrichtern	75
5.2 Implementierung des Interface <i>Leistungsbetrachtung</i>	76
5.2.1 Überblick über Anwendung des Interface <i>Leistungsbetrachtung</i> im Asynchronmotor	76
5.2.2 Implementierung des Interface <i>Leistungsbetrachtung</i> zur Motordrehzahlverstellung	77
5.2.3 Motorspannungseinstellung mit Hilfe des Motorschlupfes ..	81
5.2.4 Kompilieren des Programms	83
5.3 Zusammenfassung	85
Literatur	86
6 Charakterisierung der Asynchronmotoren mit Hilfe von Java	
Design-Pattern	87
6.1 Java Design Pattern für die Anwendungen der Stromrichter in der elektrischen Antriebstechnik.	88
6.1.1 Einsatz von Asynchronmaschinen	88
6.1.2 Design Pattern für die Energieersparnis	88
6.1.3 Design Pattern „Visitor“ für die Stromrichteranwendungen ..	93
6.2 Anwendungen der abstrakten Klasse und Schnittstellen zur Ermittlung der Energieverluste des Asynchronmotors	111
6.2.1 Nutzung der abstrakten Klasse zur Analyse des Betriebsverhaltens der Asynchronmaschine	113
6.2.2 Schnittstellen zur Analyse der Verluste	117
6.3 Zusammenfassung	140
Literatur	140
Stichwortverzeichnis	143