

Inhaltsverzeichnis

Vorwort.....	1
Teil 1 – Grundlagen und Konzepte	3
G1 Motivation für den Einsatz eines modularen Software-Entwurfs	5
1.1 Elemente einer Applikation	9
1.2 Applikationsbeispiel fertigungstechnische Demonstrationsanlage (diskrete Prozesse)	10
1.3 Funktionseinheiten und Aggregate.....	11
1.4 Ablauf und Werkstück.....	12
1.5 Parameter und Rezepte	13
1.6 Betriebsarten.....	13
1.7 Fehlerbehandlung	15
1.8 HMI (Human Machine Interface)	16
1.9 Variantenbildung durch Einflüsse von Hardware (-layout) und Kundenwünschen.....	17
1.10 Datenaufzeichnung zur Qualitätssicherung oder als Nachweis.....	19
1.11 Herausforderungen beim modularen Softwareentwurf für Produktionssysteme durch querliegende Funktionalitäten und verschiedene Betriebsarten	19
G2 Objektorientierte Erweiterung der IEC 61131-3	22
2.1 Motivation für Vorteile durch Objektorientierung – Beispiel Pneumatikzylinder	23
2.2 Theorie: Neue Sprachmittel und Schlüsselwörter	27
2.3 Der erste Schritt zur Objektorientierung: Objekte	29
2.3.1 Aufruf von Operationen eines Objekts.....	30
2.4 Der zweite Schritt zur Objektorientierung: Klassen.....	31
2.4.1 Definition von Klassen.....	32
2.4.2 Erstellung von Varianten durch Vererbung	33
2.5 Standardisierung durch Interfaces	34
G3 Einführung der Demonstrationsanlagen	36
3.1 Applikationsbeispiel einer fertigungstechnischen Anlage	36
3.1.1 Struktur und dynamischer Ablauf	36
3.1.2 Hardwarekomponenten der PPU.....	38
3.1.3 Hardwaremodule (Funktionseinheiten) der PPU.....	40
3.2 Applikationsbeispiel prozesstechnische Demonstrationsanlage.....	45

Teil 2 – Anwendungsbeispiel Fabrikautomatisierung	47
F1 Erstellung und Verwendung einer Methode	49
1.1 Aufgabenstellung	49
1.2 Konventionelle Umsetzung	50
1.2.1 Lösung (a): Direkte Programmierung im FB	53
1.2.2 Lösung (): Verwendung einer Zustandsmaschine (state chart)	55
1.2.3 Lösung (a,): Programm für den Aufruf	59
1.3 Idee.....	60
1.4 Grundlagen: Methoden.....	61
1.5 Umsetzung mit Methoden	63
1.5.1 Lösung (c): Methodenaufruf mit Zustandsmaschine (state chart)	63
1.5.2 Lösung (c): Programm für den Aufruf.....	67
1.5.3 Zusatzinformationen zur Fehlerbehandlung.....	69
1.6 Anhang: Ergänzungen zu Datentypen und speziellen Funktions- bausteinen	70
F2 Erstellung und Verwendung von Basisbaustein und Property	73
2.1 Aufgabenstellung	73
2.2 Umsetzung mit Methoden	74
2.2.1 Lösung (<i>StampExtension</i>): Ansteuerung des Stempels mit Methoden	74
2.2.2 Lösung (<i>StampExtension</i>): Programm für den Aufruf des Stempels mit Methoden	78
2.3 Idee.....	80
2.4 Grundlagen: Basisbausteine und Properties	81
2.5 Umsetzung mit Basisbaustein und Properties	85
2.5.1 Lösung (<i>Base+Prop</i>): Ansteuerung des Stempels mit Methoden, Vererbung mit Basisbaustein und Properties.....	86
2.5.2 Lösung (<i>Base+Prop</i>): Programm für den Aufruf des Stempels mit Methoden, Vererbung mit Basisbaustein und Properties	87
2.6 Gegenüberstellung Lösungsansätze	88
F3 Erstellung und Verwendung von Interfaces –	
Pattern 1: Energieüberwachung.....	90
3.1 Aufgabenstellung	90
3.2 Umsetzung mit Methoden, Basisbaustein und Properties.....	91
3.2.1 Lösung mit bisherigen Mitteln (Projekt <i>EnergyWithoutInterfaces</i>): Energieüberwachung mit Methoden, Vererbung mit Basis- baustein und Properties	92
3.2.2 Lösung mit bisherigen Mitteln (Projekt <i>EnergyWithoutInterfaces</i>): Programm für den Aufruf des Stempels mit Methoden, Vererbung mit Basisbaustein und Properties.....	96

3.3	Idee	98
3.4	Grundlagen Interfaces	100
3.5	Umsetzung mit Interfaces – Pattern 1	101
3.5.1	Lösung mit Interface Pattern 1 (Projekt <i>Interface</i>): Energieüberwachung mit Methoden, Vererbung mit Basisbaustein und Properties, Interfaces	103
3.5.2	Lösung mit Interface Pattern 1 (Interface): Programm für den Aufruf des Stempels mit Methoden, Vererbung mit Basisbaustein und Properties, Interfaces	104
3.6	Gegenüberstellung der Lösungsansätze	106
F4	Erstellung und Verwendung von Interfaces – Pattern 2: Meldefunktion .	108
4.1	Aufgabenstellung und theoretische Grundlagen	108
4.2	Umsetzung mit Methoden, Basisbaustein und Properties, Interfaces (gemäß Pattern 1)	110
4.2.1	Einführung einer Meldefunktion mit Visualisierung (Projekt <i>ErrorClassic</i>)	112
4.2.2	Anschluss der Module an die Meldefunktion mit Methoden, Vererbung mit Basisbaustein und Properties, Interfaces (gemäß Pattern 1, Projekt <i>ErrorClassic</i>).	115
4.3	Idee abstraktes Interface (Pattern 2)	117
4.4	Umsetzung mit Interface (Pattern 2)	119
4.4.1	Lösung (ErrorOOP): Erweiterte Meldefunktion mit Methoden, Vererbung mit Basisbaustein und Properties, Interfaces (Pattern 1 & 2)	122
4.4.2	Anschluss der Module an die erweiterte Meldefunktion mit Methoden, Vererbung mit Basisbaustein und Properties, Interfaces (Pattern 1 & 2) (Projekt <i>ErrorOOP</i>)	130
4.5	Gegenüberstellung der Lösungsansätze	135
F5	Beschreibung des HMIs der Anlage	136
5.1	Aufbau und Funktionsweise des HMIs.	136
5.1.1	Komponente 1: Bedienfeld	138
5.1.2	Komponente 2: Meldeprotokoll	139
5.1.3	Komponente 3: Energieüberwachung	140
5.2	Zugrundeliegende Gestaltungsrichtlinien.	141
5.3	Anpassungen der Steuerungssoftware.	145
F6	Anlagenerweiterung.	154
6.1	Aufgabenstellung.	154
6.2	Erweiterung der PPU.	155
6.2.1	Beschreibung Stempel.	155
6.2.2	Notwendige Änderungen der Steuerungssoftware ohne objektorientierte Erweiterung.	156

6.2.3	Notwendige Änderungen der Steuerungssoftware unter Verwendung von OOP.....	157
6.3	Umsetzung – Erweiterung des Stempels um die Module „Greifer“ und „Kran“	157
6.3.1	Schritt 1: Analyse der hinzukommenden Hardware-Module.	157
6.3.2	Schritt 2: Wiederverwendbare Implementierung der neuen Objekttypen „Greifer“ und „Kranbasis“.....	160
6.3.3	Schritt 3: Anpassung der applikationsspezifischen Software-Teile zur Integration der neuen Module.....	167
F7	Handbetrieb und Verriegelung	174
7.1	Aufgabenstellung	174
7.2	Umsetzung Handbetrieb und Verriegelung	177
7.2.1	Schritt 1: Einführung eines Handbetriebs für die erweiterte Anlage.....	177
7.2.2	Schritt 2: Einführung einer Verriegelung für die erweiterte Anlage.....	186
F8	Gesamtanlage mit erweiterter Fehlerüberwachung und Abläufen als Objekt: Methodenpropagation.....	192
8.1	Aufgabenstellung	192
8.2	Umsetzung der Aufgabenstellung	196
8.2.1	Erweiterung der Anlage um das Modul „Sortierstrecke“ als komplexes Objekt	196
8.2.2	Lichtschraken mit Fehlerüberwachung als Submodule der Sortierstrecke	209
8.2.3	Implementierung unterschiedlicher Abläufe als Objekt	216
F9	Zusammenfassung des funktionstechnischen Beispiels.....	223
Teil 3	– Anwendungsbeispiel Prozessautomatisierung.....	227
P1	Einführung und Umsetzung des Prozessbeispiels	229
1.1	Einführung Prozessbeispiel.....	229
1.2	Applikationsbeispiel Prozessanlage	231
1.3	Konventionelle Programmierung der Prozessanlage	232
1.3.1	Schritt 1: Projektaufbau	233
1.3.2	Schritt 2: Applikationsspezifische Teile.....	235
1.3.3	Schritt 3: Wiederverwendbare Teile	240
P2	Erstellung und Entwicklung einer OOP-Schale	251
2.1	Aufgabenstellung	251
2.2	Idee	252

2.3	Umsetzung OOP-Schale	253
2.3.1	Schritt 1: Implementierung der objektorientierten Bestandsbibliothek	256
2.3.2	Schritt 2: Anpassung/Erweiterung der applikations-spezifischen Teile.....	261
P3	Erstellung und Verwendung von Interfaces – Pattern 3 zur Simulation .	266
3.1	Aufgabenstellung	266
3.2	Idee	267
3.3	Umsetzung mit Interfaces 3	268
3.3.1	Schritt 1: Projektaufbau	268
3.3.2	Schritt 2: Applikationsspezifische Teile	268
3.3.3	Schritt 3: Wiederverwendbare Teile	272
P4	Erstellung und Verwendung einer Ablaufsteuerung als editierbare Sequenz	284
4.1	Aufgabenstellung	284
4.2	Idee	285
4.3	Umsetzung der Ablaufsteuerung	285
4.3.1	Schritt 1: Wiederverwendbare Implementierung der editierbaren Sequenz	286

4.3.2	Schritt 2: Erweiterung der applikationsspezifischen Teile des Projekts	292
Teil 4 – Weiterführende Konzepte zur Qualitätssicherung und Wiederholung der klassischen IEC 61131-3		295
G4 Architektur von Steuerungssoftware		297
4.1	Einflussfaktoren auf Steuerungssoftware von Produktionssystemen	299
4.2	Softwarequalität nach ISO 25010	301
4.3	Standards und Normen für reife Softwarearchitektur.....	303
4.3.1	Steuerungsarchitektur gemäß der ISA 88 aus der Prozesstechnik	303
4.3.2	Softwarearchitektur für die Intralogistik gemäß der SAIL-Richtlinie.....	305
4.3.3	PackML als Standard für herstellerübergreifende, zustandsbasierte Steuerung.....	305
4.4	Typische Hierarchieebenen in industrieller Steuerungssoftware und die extrafunktionale Aufgabe der Fehlermeldung	309
4.4.1	Die extrafunktionale Aufgabe „Fehlerbehandlung“ mit Bezug zu den typischen Hierarchieebenen	311
G5 Vorgehensweise bei der Erstellung von Automatisierungssoftware		315
5.1	Modulbasierte Vorgehen zur Entwicklung und Wiederverwendung von SPS-Software	317
5.2	Modularer, interdisziplinärer Engineering-Ansatz.....	318
5.3	Vorlagenbasierter Konfigurationsansatz.....	319
5.4	Ansatz der parameterbasierten Konfiguration	320
5.5	Nutzung von Entwurfsmustern zur Wiederverwendung reifer Lösungen	321
5.6	Praktische Relevanz der vorgestellten Ansätze	322
G6 Refaktorisierung und Codeanalyse zur Qualitätsüberwachung und -verbesserung.....		325
6.1	Reverse Engineering unterstützt durch statische Codeanalyse und Metriken	327
6.2	Industrielle Anwendbarkeit der statischen Codeanalyse und Softwaremetriken zur Bewertung und Verbesserung der Softwarequalität.....	329
G7 Wiederholung der klassischen IEC 61131-3		332
7.1	Struktur eines SPS-Programms.....	341
Stichwortverzeichnis.....		345