



# Software aus dem Baukasten

Die Software-Entwicklung spielt im Maschinenbau eine zunehmend wichtige Rolle. Durchgängig, modular, übersichtlich und wiederverwendbar soll sie sein, um die Kosten des Engineerings niedrig zu halten. Darauf ist die objektorientierte Programmierung ausgerichtet.

**TEXT:** Ingrid Traintiner, Sigmatek **FOTO:** imagetwelve, Sigmatek  [www.AuD24.net/PDF/AD11941310](http://www.AuD24.net/PDF/AD11941310)

Der Automatisierungsspezialist Sigmatek hat sich bereits vor langer Zeit vom sogenannten Spaghetti-Code verabschiedet und setzt seit zwölf Jahren auf objektorientierte Programmierung (OOP). Mit der Entwicklungsumgebung Lasal stellt das Unternehmen dem Anwender ein ausgereiftes und für viele Maschinenapplikationen passendes Komplettwerkzeug zur Verfügung, mit dem sich neben der eigentlichen Steuerungsprogrammierung auch Visualisierungs-, Motion Control-, Safety- sowie Service- und Fernwartungs-Aufgaben realisieren lassen.

## Vorteile der Objektorientierung

Die Idee der OOP ist, Code und Daten in logische Einheiten oder Objekte zusammenzufassen. Hinter einem Objekt steht jeweils eine sogenannte Klasse, die der Bauplan für ein

Objekt ist und daher den Programmcode und die Datenelemente definiert. Jede Klasse übernimmt eine bestimmte Aufgabe, wie die Messung und Auswertung einer Temperatur oder die Ansteuerung eines Antriebs. Bei Lasal wird der eigentliche Programmcode eines Objekts in den gebräuchlichen Sprachen der IEC 61131-3 implementiert.

## Wiederverwendbare Module

Mit den Lasal-Tools gestaltet sich die Entwicklung einfach und die Software ist durchgängig, modular, übersichtlich und wiederverwendbar. Die einzelnen Module sind nach außen abgeschottet, die Schnittstellen klar definiert und die Dokumentation übersichtlich, was die Wartung vereinfacht. Diese Faktoren helfen dem Maschinenbauer dabei, die Entwicklungs- aber auch die Gesamtkosten der Maschine über

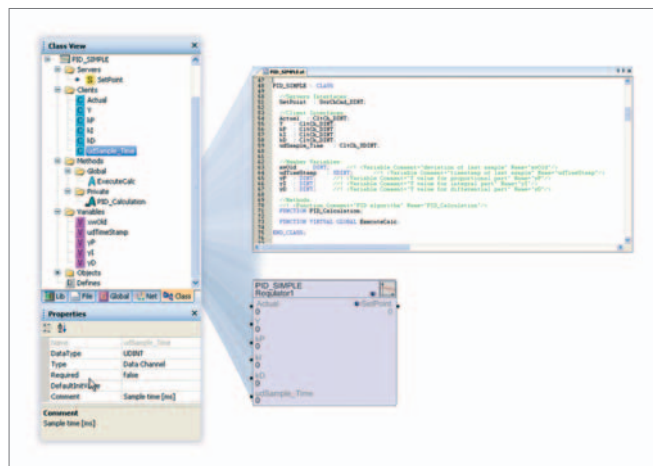


den kompletten Lebenszyklus zu verringern und die Entwicklungszeiten zu reduzieren.

Die Module der Software lassen sich wie in einem Baukastensystem zusammensetzen. Durch die modulare Struktur können bei der OOP einmal erstellte Applikationsteile wieder verwendet werden, ohne diese noch einmal prüfen zu müssen. Software wird auf diese Weise nachhaltig, das ist gerade bei komplexen Applikationen ein wichtiger Faktor. Durch die Strukturiertheit der OOP ist es möglich, bestehende Klassen zu erweitern beziehungsweise zu verändern – und zwar ohne dazu in die ursprüngliche Klasse eingreifen zu müssen. Die Änderungen sind somit immer nachvollziehbar und einfach zu erkennen. Getestete Bausteine werden in sogenannten Klassenbibliotheken abgelegt und lassen sich dann in unterschiedlichen Projekten oder Systemteilen wiederverwenden. Funktionsklassen für verschiedene Anwendungskategorien stehen in Standardbibliotheken zur Verfügung.

## Überblick garantiert

Die grafische Darstellung von Lasal trägt auch zur Vereinfachung bei. Dabei werden die von Klassen erzeugten Objekte (Maschinenmodule) in sogenannten Netzwerken dargestellt.



Über die Baumstruktur von Lasal Class wird die Klasse mit ihren Eigenschaften, Methoden und Schnittstellen definiert – Deklaration und grafische Darstellung übernimmt das Tool automatisch

So erkennt der Anwender auf einen Blick die Eigenschaften eines Maschinenteils sowie die Kommunikation mit anderen Objekten.

## Zeit und Kosten sparen

Mit dem Simulations-Tool Lars kann der Code ohne angeschlossene Steuerung auf einem Windows-Rechner simuliert und getestet werden. So ist es möglich, die Funktionalität der Maschine zu überprüfen und den Quellcode zu debuggen, noch bevor die mechanischen Komponenten fertig sind. Eventuelle Designfehler lassen sich frühzeitig erkennen und Programmierzeiten verringern.

ein Software-Baustein hat die Aufgabe eine Temperatur zu regeln. Es sind tausende Maschinen in unzähligen Varianten am Markt und dann stellt man fest, dass der Temperaturregler einen schwerwiegenden Fehler enthält. Klassischerweise müsste der Hersteller nun für jede Maschine den spezifischen Software-Stand korrigieren und an der Maschine updaten. Lasal erstellt nun auf Knopfdruck einen USB-Stick mit der korrigierten Klasse, in unserem Beispiel dem korrekten Reglerbaustein. Dieser kann überall per Stick oder Online-Fernwartung eingespielt werden, ohne dafür die tatsächlichen Programme auf den Maschinen kennen zu müssen.

## Flexible Updates

Die gekapselten Objekte kommunizieren über Schnittstellen mit der Außenwelt. Unsauberkeiten wie Zugriffe auf Daten an X-Stellen im Projekt sind somit erst gar nicht möglich. Dank dieser klar definierten Schnittstellen sind Objekte später einfach gegen andere austauschbar. Zudem eröffnet Lasal mit der Kapselung flexible Möglichkeiten bei Programm-Updates oder Fehlerkorrekturen. Dazu ein Beispiel: Angenommen

## Code automatisch erstellen

Basierend auf den gekapselten Software-Bausteinen lassen sich mit Hilfe der standardisierten Script-Sprache Python Applikationen automatisch erstellen oder abändern. Oft gibt es für eine bestimmte Maschinenbaureihe ein einziges Basisprojekt. Von diesem können dann automatisch verschiedene Varianten für die kundenspezifischen Maschinen derselben Baureihe abgeleitet werden. □

**> MORE@CLICK AD11941310**