

Glossar

über alle Themengebiete

Copyright © 1999 by Bernd Oestereich.

Dieses Glossar darf in vollständiger Form und unverändert jederzeit kopiert und kostenlos weitergegeben werden. Der Hinweis auf die Originalquelle <http://www.oose.de/glossar> muß ebenso wie dieser Copyright-Hinweis stets angegeben werden.

Es ist nicht zulässig das Glossar kommerziell zu vertreiben, gegen Entgelt weiterzugeben oder Inhalte zu verändern. Im Rahmen nicht-kommerzieller Verwendungen, beispielsweise Diplomarbeiten, darf das Glossar gerne übernommen werden. Die Verwendung in kommerziellen Zusammenhängen, beispielsweise in öffentlichen oder internen Schulungen, firmen-internen Netzwerken, Publikationen, Produkten etc. ist prinzipiell gestattet, wenn eine entsprechende Meldung an boe@oose.de gesendet wird.

Die Weitergabe ist sowohl in elektronischer als auch gedruckter Form zulässig. Im Internet zugängliche Kopien sind ebenfalls zu melden.

Fehlerhinweise, Verbesserungs- und Ergänzungsvorschläge sind willkommen und zu senden an boe@oose.de (Ergänzungsvorschläge werden jedoch kritisch geprüft).

Letzte Änderung am: 26.08.99 Anzahl der Einträge: 376

Das Gesamtglossar enthält Einträge folgender Einzelglossare:

- **[UML]**
UML-Glossar von B. Oestereich
Siehe auch *Objektorientierte Softwareentwicklung*, B. Oestereich, Oldenbourg Wissenschaftsverlag, 1999.
- **[OEP]**
Spezifische Begriffe des Object Engineering Process (OEP) sowie allgemeine Begriffe zu Vorgehensmodellen,
siehe <http://www.oose.de/oep>.
- **[EmOO]**
Glossareinträge aus dem Buch *Erfolgreich mit Objektorientierung, Vorgehensmodelle und Managementpraktiken für objektorientierte Softwareentwicklung*, B. Oestereich et. al., Oldenbourg Wissenschaftsverlag, 1999.
- **[engl.]**
Englischsprachige OO- und UML-Begriffe.
Entnommen aus *Objektorientierte Softwareentwicklung*, B. Oestereich, Oldenbourg Wissenschaftsverlag, 1999 und *Developing Software with UML*, B. Oestereich, Addison Wesley,
- **[OO allg.]**
Allgemeines OO-Glossar von B. Oestereich
Siehe auch *Objektorientierte Softwareentwicklung*, B. Oestereich, Oldenbourg Wissenschaftsverlag, 1999.

A Ä B C D E F G H I J K L M N O Ö P Q R S T U Ü V W X Y Z

A

Abgeleitete Assoziation [UML]

Eine abgeleitete Assoziation ist eine Assoziation, deren konkrete Objektbeziehungen jederzeit aus den Werten anderer Objektbeziehungen und ihrer Objekte abgeleitet (berechnet) werden können.

Siehe auch [Assoziation \[UML\]](#) [Abgeleitetes Element \[UML\]](#)

Abgeleitetes Attribut [UML]

Ein abgeleitetes Attribut wird aus den Werten anderer Attribute berechnet. Abgeleitete Attribute können nicht direkt geändert werden und werden durch eine Berechnungsoperation implementiert oder gesetzt.

Siehe auch [Abgeleitetes Element \[UML\]](#) [Attribut \[OO allg.\]](#)

Abgeleitetes Element [UML]

Synonym/Akronym/Übersetzung: [derived element](#) [engl.]

Ein Modellelement, das jederzeit durch ein anderes Element berechnet werden kann und nur der Klarheit wegen gezeigt wird oder für Designzwecke zugefügt wird, ohne daß es jedoch weitere semantische Information zugefügt.

Abhängigkeitsbeziehung [UML]

Eine Abhängigkeitsbeziehung ist eine Beziehung zwischen zwei Modellelementen, die zeigt, daß eine Änderung in dem einen (unabhängigen) Element eine Änderung in dem anderen (abhängigen) Element notwendig macht.

Siehe auch [Depends on-Beziehung \[UML\]](#) [Beziehung \[UML\]](#)

Ablauforganisation [OEP]

Ablauforganisation ist die zeitliche und räumliche Ordnung betrieblicher Prozesse innerhalb des durch die Aufbauorganisation geschaffenen Rahmens.

abstract class [engl.]

Synonym/Akronym/Übersetzung: [Abstrakte Klasse](#) [OO allg.]

Abstrakte Klasse [OO allg.]

Von einer abstrakten Klasse werden niemals Objektexemplare erzeugt; sie ist bewußt unvollständig und bildet somit die Basis für weitere Unterklassen, die Exemplare haben können.

Siehe auch [Virtuelle Klasse \[OO allg.\]](#) [Klasse \[OO allg.\]](#)

Abstrakte Operation [OO allg.]

Eine Operation, für die nur eine Signatur, jedoch keine Anweisungsfolge definiert ist, d.h. Die Operation ist definiert, aber noch nicht implementiert. Sie wird in einer abgeleiteten Klassen implementiert.

Siehe auch [Virtuelle Operation \[OO allg.\]](#) [Operation \[UML\]](#)

Abstrakter Datentyp [OO allg.]

Das Konzept des abstrakten Datentyps ähnelt dem der Klasse. Unter einem abstrakten Datentyp versteht man die Zusammenfassung von Daten und der mit ihnen ausführbaren Operationen.

Siehe auch [Klasse \[OO allg.\]](#)

Abstraktion [OO allg.]

Abstraktion ist eine Methode, bei der unter einem bestimmten Gesichtspunkt die wesentlichen Merkmale eines Gegenstandes oder Begriffes herausgesondert werden.

action state [engl.]

Synonym/Akronym/Übersetzung: [Aktivität](#) [UML]

activity diagram [engl.]

Synonym/Akronym/Übersetzung: [Aktivitätsdiagramm](#) [UML]

activity graph [engl.]

Synonym/Akronym/Übersetzung: [Aktivitätsdiagramm](#) [UML]

Letzte Änderung von boe am 26.08.99

actor [engl.]

Synonym/Akronym/Übersetzung: [Akteur](#) [UML]

Aggregat [UML]

Die an einer Aggregation (Ganze-Teile-Beziehung) beteiligte Klasse, die das "Ganze" darstellt.

Siehe auch [Aggregation](#) [UML]

aggregation [engl.]

Synonym/Akronym/Übersetzung: [Aggregation](#) [UML]

Aggregation [UML]

Eine Aggregation ist eine Sonderform der Assoziation, bei der die beteiligten Klassen keine gleichwertige Beziehung führen, sondern eine Ganzes-Teile-Hierarchie darstellen. Eine Aggregation beschreibt, wie sich etwas Ganzes aus seinen Teilen zusammensetzt.

Siehe auch [Komposition](#) [UML] [Assoziation](#) [UML]

Akteur [UML]

Synonym/Akronym/Übersetzung: [actor](#) [engl.]

Ein Akteur ist ein außerhalb des Systems agierende beteiligter, der an der in einem Anwendungsfall beschriebenen Interaktion beteiligt ist. Akteure nehmen in der Interaktion gewöhnlich eine definierte Rolle ein. Es sind Benutzer des Systems oder andere Systeme. Ein Akteur ist eine stereotypisierte Klasse.

Siehe auch [Stereotyp](#) [UML] [Klasse](#) [OO allg.] [Anwendungsfall](#) [UML]

Akteur [OEP]

Synonym/Akronym/Übersetzung: [Rolle](#) [OEP]

Ein Akteur ist jemand der an der Durchführung einer Aktivität mitwirkt.

Siehe auch [Aktivität](#) [OEP]

Aktive Klasse [UML]

Eine Klasse, deren Instanzen nebenläufig ausgeführt werden und ihren eigenen Kontrollfokus (Thread) besitzen.

Siehe auch [Aktives Objekt](#) [UML]

Aktives Objekt [UML]

Instanz einer aktiven Klasse.

Siehe auch [Aktive Klasse](#) [UML]

Aktivität [UML]

Ein Zustand mit einer internen Aktion und einer oder mehreren ausgehenden Transitionen, die automatisch dem Abschluß der internen Aktion folgen. Eine Aktivität ist ein einzelner Schritt in einem Ablauf. Eine Aktivität kann mehrere ausgehende Transitionen haben, wenn diese durch Bedingungen unterschieden werden können.

Siehe auch [Aktivitätsdiagramm](#) [UML] [Transition](#) [UML] [Zustand](#) [UML]

Aktivität [OEP]

Eine Aktivität ist die konkrete Durchführung von definierten Aktionen innerhalb eines Entwicklungsprozesses und Ausprägung eines [Aktivitätstyps](#).

Siehe auch [Aktivitätstyp](#) [OEP]

Aktivitätsdiagramm [UML]

Synonym/Akronym/Übersetzung: [activity diagram](#) [engl.]

Ein Aktivitätsdiagramm ist eine spezielle Form des [Zustandsdiagramms](#), das überwiegend oder ausschließlich Aktivitäten enthält. Es kann die Abläufe einer einzelnen Operation näher spezifizieren oder einen Ablauf, an dem verschiedene Objekte beteiligt sind (Workflow).

Siehe auch [Operation](#) [UML] [Workflow](#) [OO allg.] [Zustandsdiagramm](#) [UML] [Aktivität](#) [OEP]

Aktivitätstyp [OEP]

Ein Aktivitätstyp ist eine abstrakte Beschreibung von Tätigkeiten, um ein oder mehrere definierte Ergebnisse zu erzeugen. Ein Aktivitätstyp ist somit eine Art Arbeitsanleitung.

Amigos [OO allg.]

Grady Booch, James Rumbaugh und Ivar Jacobson sind die Initiatoren der Unified Modeling Language (UML) und des Unified Process und werden häufig einfach "die Amigos" genannt.

Siehe auch [Unified Process](#) [OEP] [Unified Modeling Language](#) [UML]

Analyse [OEP]

Mit (objektorientierter) Analyse werden alle Aktivitäten im Rahmen des Softwareentwicklungsprozesses bezeichnet, die der Ermittlung, Klärung und Beschreibung der Anforderungen an das System dienen (d.h. die Klärung, was das System leisten soll). Im Gegensatz zur [Anforderungsanalyse](#) findet dies in der Sprache der Softwareentwickler statt (z.B. UML).

Siehe auch [Anforderungsanalyse](#) [OEP]

Anforderungsanalyse [OEP]

Teil des Softwareentwicklungsprozesses und Menge von Entwicklungsaktivitäten, die der Identifizierung, Analyse und Beschreibung von Anforderungen an die zu erstellende Software dienen.

Siehe auch [Analyse](#) [OEP]

Annotation [OO allg.]

Synonym/Akronym/Übersetzung: [Notiz](#) [UML]

Anwendungsarchitektur [OO allg.]

Die fachliche und technische [Architektur](#) einer Software-Anwendung.

Siehe auch [Architektur](#) [OO allg.]

Anwendungsfall [UML]

Synonym/Akronym/Übersetzung: [use case](#) [engl.]

Ein Anwendungsfall beschreibt in Klartext eine Menge konsistenter und zielgerichteter Interaktionen von Akteuren mit einem System, an deren Ende ein definiertes Ergebnis entstanden ist. Ein Anwendungsfall wird stets durch einen Akteur initiiert und ist eine komplette, unteilbare Beschreibung. Anwendungsfälle beschreiben das gewünschte externe Systemverhalten aus der Sicht und in der Sprache der Anwender und somit Anforderungen, die das System zu erfüllen hat. Beschrieben wird, was es leisten muß, aber nicht wie es dies leisten soll.

Anwendungsfälle

- eine IST-Situation beschreiben (Ist-Anwendungsfall)
- einen SOLL-Zustand beschreiben (Soll-Anwendungsfall)
- eine Essenz beschreiben (Essentielle Anwendungsfälle)
- nur die durch Software zu unterstützenden Sachverhalte beschreiben (System-Anwendungsfälle)
- auch außerhalb der Software und allgemeine geschäftliche Anwendungsfälle darstellen (Geschäfts-Anwendungsfälle)
- ganz einfach verwendet werden (Name + Kurzbeschreibung)
- semi-formal und detailliert verwendet werden (bis zu aktorspezifischen Schrittbeschreibungen)

Siehe auch [Essentieller Anwendungsfall](#) [OO allg.] [Anwendungsfalldiagramm](#) [UML] [Akteur](#) [UML]

Anwendungsfalldiagramm [UML]

Synonym/Akronym/Übersetzung: [use case diagram](#) [engl.]

Ein Diagramm, das die Beziehungen zwischen Akteuren und Anwendungsfällen zeigt.

Siehe auch [Anwendungsfall](#) [UML] [Akteur](#) [UML]

Anwendungsfallmodell [UML]

Ein Modell, das die funktionalen Anforderungen an ein System in Form von Anwendungsfällen beschreibt.

Siehe auch [Anwendungsfall](#) [UML]

Arbeitsauftrag [OEP]

Beschreibt für eine Person oder ein Team den Auftrag, ein definiertes Ergebnis herzustellen. Rahmenbedingungen wie geschätzter bzw. geplanter Aufwand, Start- und Endtermin, Form und Qualität des Ergebnisses, ggf. einzusetzende Werkzeuge u. ä. werden beschrieben.

Arbeitspaket [OEP]

Eine Menge von konkreten Arbeitsaufträgen oder die Beschreibung einer Aufgabe, die in mehrere konkrete Arbeitsaufträge zerlegbar ist.

Siehe auch [Arbeitsauftrag](#) [OEP]

Architektur [OO allg.]

Architektur ist die Spezifikation der grundlegenden Struktur eines Systems.

Siehe auch [Anwendungsarchitektur](#) [OO allg.]

Architektur-Prototyp [OEP]

Ein [Prototyp](#), mit dem die prinzipielle Brauchbarkeit und Funktionsfähigkeit einer technischen oder fachlichen [Architektur](#) nachgewiesen werden soll.

Siehe auch [Architektur](#) [OO allg.] [Prototyp](#) [OEP]

Architekturzentriert [OEP]

Im Kontext des Entwicklungsprozesses bedeutet dies, daß bereits der Entwicklungsprozess die besonderen Gegebenheiten der [Architektur](#) (Konzepte, Abstraktionen, Artefakte u. ä.) berücksichtigt.

Siehe auch [Architektur](#) [OO allg.]

Argument [OO allg.]

Konkreter Wert eines [Parameters](#).

Siehe auch [Parameter](#) [UML]

Artefakt [OEP]

Ein Artefakt ist ein Synonym für Produkt. Als Artefakte werden jegliche Produkte bezeichnet, die im Rahmen des Softwareentwicklungsprozesses erzeugt werden (Klassen, Testfälle, Projektpläne, Code etc.)

artifact [engl.]

Synonym/Akronym/Übersetzung: [Artefakt](#) [OEP]

assertion [engl.]

Synonym/Akronym/Übersetzung: [Zusicherung](#) [UML]

Assertions sind boolesche Ausdrücke, die niemals unwahr werden sollen und anderenfalls auf Fehler hinweisen. Typischerweise sind Assertions nur zur Entwicklungszeit aktiviert.

association [engl.]

Synonym/Akronym/Übersetzung: [Assoziation](#) [UML]

association class [engl.]

Synonym/Akronym/Übersetzung: [Assoziationsklasse](#) [UML]

association end [engl.]

Synonym/Akronym/Übersetzung: [Assoziationsende](#) [UML]

Letzte Änderung von boe am 26.08.99

association role [engl.]

Synonym/Akronym/Übersetzung: [Assoziationsrolle](#) [UML]

Assoziation [UML]

Eine Assoziation beschreibt eine Relation zwischen Klassen, d.h. die gemeinsame Semantik und Struktur einer Menge von Objektbeziehungen. Es werden gerichtete Assoziationen (nur einseitig direkt navigierbar) und bidirektionale Assoziationen (beidseitig direkt navigierbar) unterschieden. Die beiden Enden einer Assoziation sind Assoziationsrollen.

Siehe auch [Klasse](#) [OO allg.] [Objektverbindung](#) [UML]

Assoziationsende [UML]

Siehe auch [Assoziationsrolle](#) [UML]

Assoziationsklasse [UML]

Ein Modellelement, das sowohl über die Eigenschaften einer Klasse als auch über die einer Assoziation verfügt. Es kann gesehen werden als Assoziation mit zusätzlichen Klasseneigenschaften (Attributierte Assoziation) oder als Klasse mit zusätzlichen Assoziationseigenschaften (Assoziationsklasse).

Siehe auch [Attributierte Assoziation](#) [UML]

Assoziationsrolle [UML]

Die Rolle, die ein Typ oder eine Klasse in einer Assoziation spielt. D.h. eine Rolle repräsentiert eine Klasse in einer Assoziation. Gewöhnlich befinden sich Assoziationen zwischen zwei Klassen. Eine Klasse kann aber auch eine Assoziation zu sich selbst haben kann; in diesem Fall sind die beiden Enden der Assoziation nur durch die Rollenangabe zu unterscheiden.

Siehe auch [Assoziation](#) [UML]

Attribut [OO allg.]

Eine benannte Eigenschaft eines Typs. Ein Attribut ist ein Datenelement, das in jedem Objekt einer Klasse gleichermaßen enthalten ist und von jedem Objekt mit einem individuellen Wert repräsentiert wird. Im Gegensatz zu Objekten haben Attribute außerhalb des Objektes, von dem sie Teil sind, keine eigene Identität. Attribute sind vollständig unter der Kontrolle der Objekte, von denen sie Teil sind.

Siehe auch [Identität](#) [OO allg.] [Klasse](#) [OO allg.] [Typ](#) [UML] [Objekt](#) [OO allg.]

attribute [engl.]

Synonym/Akronym/Übersetzung: [Attribut](#) [OO allg.]

Attributierte Assoziation [UML]

Eine Assoziation die über eigene Attribute verfügt.

Siehe auch [Assoziation \[UML\]](#)

Audit [OEP]

Ein Audit ist eine Untersuchung einer Organisationseinheit oder eines Projektes, um die Einhaltung und Wirksamkeit von Regelungen, Verfahren und Standards zu ermitteln und zu dokumentieren.

Siehe auch [Inspektion \[OEP\]](#) [Review \[OEP\]](#)

Aufgebrochene Assoziation [UML]

Eine attributierte Assoziation, bei der die Attribute zu einer gewöhnlichen Klasse überführt und die attributierte Assoziation unter Einbeziehung dieser neuen Klasse in zwei gewöhnliche Assoziationen umgeformt wurde.

Siehe auch [Assoziation \[UML\]](#) [Attributierte Assoziation \[UML\]](#)

Aufwand [EmOO]

Aufwand ist eine Größe der Dimension [Zahl von Personen] * [Zeit]. Der Aufwand in einem Projekt ergibt sich als Integral der [Projektbesetzung](#) über eine jeweils vorgegebene Zeit. Der Gesamtaufwand des Projektes ist auch der [Herstellungsaufwand](#) des jeweiligen Produktes.

Siehe auch [Herstellungsaufwand \[EmOO\]](#) [Projektbesetzung \[EmOO\]](#)

Ä**Ähnlichkeitsbeziehung [UML]**

Synonym/Akronym/Übersetzung: [Resemble-Beziehung \[UML\]](#)

Änderungsmanagement (Organisationseinheit) [OEP]

Entscheidungs- und Steuerungsgremium für das Änderungsmanagement.

Änderungsmanagement (Tätigkeit) [OEP]

Management zur Aufnahme, Verfolgung und Durchführung von Änderungen, Erweiterungen und Fehlerbeseitigungen.

B**Basisklasse [OO allg.]**

Synonym/Akronym/Übersetzung: [Oberklasse \[OO allg.\]](#)

Behälterklasse [OO allg.]

Synonym/Akronym/Übersetzung: [Sammlung \[OO allg.\]](#)

Beziehung [UML]

UML-Modellelemente können verschiedene Arten von Beziehungen untereinander haben. Klassen können beispielsweise Spezialisierungsbeziehungen, Assoziationen, Realisierungs- und Abhängigkeitsbeziehungen haben.

Siehe auch [Realisierungsbeziehung \[UML\]](#) [Abhängigkeitsbeziehung \[UML\]](#) [Vererbung \[OO allg.\]](#) [Assoziation \[UML\]](#)

Bidirektionale Assoziation [UML]

Eine bidirektionale Assoziation ist eine beidseitig direkt navigierbare Assoziation, d.h. eine Assoziation, bei der von beiden beteiligten Assoziationsrollen zur jeweils anderen direkt navigiert werden kann.

Siehe auch [Navigation, Navigierbarkeit \[OO allg.\]](#) [Assoziationsrolle \[UML\]](#) [Assoziation \[UML\]](#)

Binäre Assoziation [UML]

Eine Assoziation zwischen zwei [Assoziationsrollen](#).

Siehe auch [Assoziationsrolle \[UML\]](#) [Assoziation \[UML\]](#) [Mehrgliedrige Assoziation \[UML\]](#)

Letzte Änderung von boe am 26.08.99

binary association [engl.]

Synonym/Akronym/Übersetzung: [Binäre Assoziation \[UML\]](#)

Bindung [OO allg.]

Synonym/Akronym/Übersetzung: [Dynamische Bindung \[OO allg.\]](#)

Botschaft [OO allg.]

Synonym/Akronym/Übersetzung: [Nachricht \[OO allg.\]](#)

Build [OEP]

Eine gewöhnlich unvollständige und vorübergehende aber ausführbare Version des Systems.

Siehe auch [Version \[OEP\]](#)

C**Change Management [OEP]**

Synonym/Akronym/Übersetzung: [Änderungsmanagement \(Tätigkeit\) \[OEP\]](#)

class [engl.]

Synonym/Akronym/Übersetzung: [Klasse \[OO allg.\]](#)

class diagram [engl.]

Synonym/Akronym/Übersetzung: [Klassendiagramm \[UML\]](#)

collaboration [engl.]

Synonym/Akronym/Übersetzung: [Kollaboration \[UML\]](#)

collaboration diagram [engl.]

Synonym/Akronym/Übersetzung: [Kollaborationsdiagramm \[UML\]](#)

collection [engl.]

Synonym/Akronym/Übersetzung: [Sammlung \[OO allg.\]](#)

component [engl.]

Synonym/Akronym/Übersetzung: [Komponente \[UML\]](#)

component diagram [engl.]

Synonym/Akronym/Übersetzung: [Komponentendiagramm \[UML\]](#)

composition [engl.]

Synonym/Akronym/Übersetzung: [Komposition \[UML\]](#)

concrete class [engl.]

Synonym/Akronym/Übersetzung: [Konkrete Klasse](#) [OO allg.]

concurrency [engl.]

Synonym/Akronym/Übersetzung: [Nebenläufigkeit](#) [OO allg.]

constraint [engl.]

Synonym/Akronym/Übersetzung: [Einschränkung](#) [UML]

Container-Klasse [OO allg.]

Synonym/Akronym/Übersetzung: [Behälterklasse](#) [OO allg.]

CRC-Karten [OO allg.]

Karteikarten, auf denen der Name der Klasse (Class), ihre Aufgaben (Responsibilities) und ihre Beziehungen (Collaborations) beschrieben werden.

D**Datenabstraktion [OO allg.]**

Hierunter versteht man das Prinzip, nur die auf ein Objekt anwendbaren Operationen nach außen sichtbar zu machen. Die tatsächliche innere Realisierung der Operationen und die innere Struktur des Objektes werden verborgen, d.h. man betrachtet abstrakt die eigentliche Semantik und läßt die tatsächliche Implementierung außer acht.

Default-Implementierung [OO allg.]

Synonym/Akronym/Übersetzung: [Standard-Implementierung](#) [OO allg.]

Degenerierte Assoziationsklasse [UML]

Die (namenlose) Assoziationsklasse einer attributierten Assoziation.

Siehe auch [Attributierte Assoziation](#) [UML] [Assoziationsklasse](#) [UML]

Delegation [OO allg.]

Delegation ist ein Mechanismus, bei dem ein Objekt eine Nachricht nicht (vollständig) selbst interpretiert, sondern an ein anderes Objekt weiterleitet (propagiert).

Siehe auch [Nachricht](#) [OO allg.] [Objekt](#) [OO allg.]

dependency [engl.]

Synonym/Akronym/Übersetzung: [Abhängigkeitsbeziehung](#) [UML]

Depends on-Beziehung [UML]

Eine Beziehung zwischen Anwendungsfällen mit dem Stereotyp «depends on». Eine Depends on-Beziehung zwischen Anwendungsfällen besagt, daß ein Anwendungsfall ganz allgemein oder indirekt von einem anderen abhängig ist, d.h. erst in Zusammenhang mit diesem ein sinnvolles Gesamtsystem bzw. -modell existiert.

Siehe auch [Abhängigkeitsbeziehung](#) [UML] [Anwendungsfall](#) [UML]

deployment diagram [engl.]

Synonym/Akronym/Übersetzung: [Einsatz- und Verteilungsdiagramm](#) [UML]

Design [OEP]

Mit (objektorientiertem) Design werden alle Aktivitäten im Rahmen des Softwareentwicklungsprozesses bezeichnet, mit denen ein Modell logisch und physisch strukturiert wird und die dazu dienen zu beschreiben, wie das System die in der Analyse beschriebenen Anforderungen erfüllt.

Siehe auch [Analyse](#) [OEP]

design pattern [engl.]

Synonym/Akronym/Übersetzung: [Entwurfsmuster](#) [OO allg.]

Diskriminator [UML]

Ein Diskriminator ist ein Unterscheidungsmerkmal für die Strukturierung der Modellsemantik in Generalisierungs- und Spezialisierungsbeziehungen.

Siehe auch [Spezialisierung](#) [OO allg.] [Vererbung](#) [OO allg.] [Generalisierung](#) [OO allg.]

domain [engl.]

Synonym/Akronym/Übersetzung: [Domäne](#) [OEP]

Domäne [OEP]

Synonym/Akronym/Übersetzung: [Problembereich](#) [OEP]

Domänenmodell [OEP]

Ein Modell, beispielsweise ein Klassenmodell, welches die fachlich relevanten Sachverhalte repräsentiert, nicht aber technisch oder fachfremd motivierte Sachverhalte.

Siehe auch [Fachklassenmodell](#) [OO allg.]

Dynamische Bindung [OO allg.]

Hierunter ist zu verstehen, daß eine Nachricht erst zur Programmlaufzeit einer konkreten Operation zugeordnet wird, die diese Nachricht dann interpretiert.

Dynamische Klassifikation [OO allg.]

Ein Objekt ist nacheinander Instanz unterschiedlicher Klassen, d.h. es kann seine Klassenzugehörigkeit während seiner Lebenszeit ändern.

Siehe auch [Instantiierung](#) [OO allg.] [Klasse](#) [OO allg.] [Instanz](#) [OO allg.]

E**Eigenschaftswert [UML]**

Synonym/Akronym/Übersetzung: [tagged value](#) [engl.]

Eigenschaftswerte sind benutzerdefinierte, sprach- oder werkzeugspezifische Schlüsselwort-Wert-Paare, die die Semantik einzelner Modellelemente um spezielle Eigenschaften erweitern. Der Unterschied zum [Stereotyp](#) besteht darin, daß durch ein Stereotyp das Metamodell um ein neues Element erweitert wird. Mit Eigenschaftswerten hingegen können einzelne Ausprägungen bestehender Modellelemente (z.B. eine bestimmte Operation) um bestimmte Eigenschaften erweitern.

Siehe auch [Stereotyp](#) [UML]

Einfachvererbung [OO allg.]

Bei der Einfachvererbung erbt jede Unterklasse nur von einer direkten Oberklasse.

Siehe auch [Unterklasse](#) [OO allg.] [Oberklasse](#) [OO allg.] [Vererbung](#) [OO allg.]

Einsatz- und Verteilungsdiagramm [UML]

Synonym/Akronym/Übersetzung: [deployment diagram](#) [engl.]

Ein Diagramm, welches die Konfiguration der zur Laufzeit vorhandenen (eingesetzten) [Knoten](#) und ihrer [Komponenten](#), Prozesse und Objekte zeigt.

Siehe auch [Knoten \[UML\]](#) [Komponente \[UML\]](#)

Einsatzdiagramm [UML]

Synonym/Akronym/Übersetzung: [Einsatz- und Verteilungsdiagramm \[UML\]](#)

Einschränkung [UML]

Synonym/Akronym/Übersetzung: [Zusicherung \[UML\]](#)

Enthalten-Beziehung [UML]

Synonym/Akronym/Übersetzung: [Include-Beziehung \[UML\]](#)

Entwicklungsprozeß [OEP]

Repräsentiert die zeitliche Perspektive und die Abfolge der Aktivitäten zur Erstellung eines Produktes.

Siehe auch [Vorgehensmodell \[OEP\]](#)

Entwurfsmuster [OO allg.]

Entwurfsmuster sind generalisierte und bewährte Lösungsansätze zu immer wiederkehrenden Entwurfsproblemen. Sie sind keine fertig codierten Lösungen, sie beschreiben lediglich den Lösungsweg.

Enumeration [OO allg.]

Synonym/Akronym/Übersetzung: [Wertemenge \[OO allg.\]](#)

Enumerationen sind Aufzählungstypen. Diese Aufzählungstypen können fest programmiert sein oder durch (ggf. datenbankgestützte) konfigurierbare Wertemengen realisiert sein. Die Wertemengen ändern sich typischerweise nur selten. Beispiel: *Familienstand = {ledig, verheiratet, geschieden, verwitwet}*.

Ereignis [OO allg.]

Ein Geschehen, das in einem gegebenen Kontext eine Bedeutung hat und sich räumlich und zeitlich lokalisieren läßt.

Ergebnistyp [OEP]

Ein Ergebnistyp ist ein abstrahiertes Ergebnis, d. h. es wird damit definiert, welche Inhalte, Struktur und Form ein bestimmter Typ von Ergebnissen hat (syntaktische und semantische Beschreibung). Zur Beschreibung des Ergebnistyps gehören auch Aussagen zur Qualität und Granularität.

Erweiterungsbeziehung [UML]

Synonym/Akronym/Übersetzung: [Extend-Beziehung \[UML\]](#)

Essentieller Anwendungsfall [OO allg.]

Ein essentieller Anwendungsfall ist ein [Anwendungsfall](#), der

- vereinfacht
- generalisiert
- abstrakt
- technologieunabhängig
- implementierungsunabhängig

beschrieben ist.

Siehe auch [Anwendungsfall \[UML\]](#)

event [engl.]

Synonym/Akronym/Übersetzung: [Ereignis \[OO allg.\]](#)

Evolutionäres Vorgehen [OEP]

Vorgehen, bei dem ausgehend von zunächst unvollständigen Anforderungen ein Ergebnis in mehreren aufeinander aufbauenden Zwischenschritten erstellt wird.

Exemplar [OO allg.]

Synonym/Akronym/Übersetzung: [Instanz \[OO allg.\]](#)

Siehe auch [Objekt \[OO allg.\]](#)

Extend-Beziehung [UML]

Synonym/Akronym/Übersetzung: [Erweiterungsbeziehung \[UML\]](#)

Eine Beziehung zwischen Anwendungsfällen mit dem Stereotyp «extend». Eine Extend-Beziehung zwischen Anwendungsfällen besagt, daß ein Anwendungsfall unter bestimmten Umständen bzw. an einer bestimmten Stelle (dem sog. "extension point") durch einen anderen erweitert wird.

Siehe auch [Anwendungsfall \[UML\]](#)

F

Fachabteilung [OEP]

Eine Organisationseinheit, die dem eigentlichen Geschäftszweck einer Organisation dient. Sie ist Partner der Softwareentwicklung, artikuliert Bedarf, Anforderungen und Änderungen für Anwendungen und nutzt Anwendungen.

Fachklasse [OO allg.]

Gewöhnliche fachlich motivierte Klasse, die einen Begriff aus dem Problembereich repräsentiert. Wird zur Unterscheidung von Businessklassen verwendet.

Siehe auch [Klasse \[OO allg.\]](#)

Fachklassenmodell [OO allg.]

Ein Klassenmodell, das ausschließlich oder vorwiegend Fachklassen enthält.

Siehe auch [Domänenmodell \[OEP\]](#)

Fachliche Architektur [OEP]

Ein Modell, daß die grundsätzlichen fachlichen Zusammenhänge eines Anwendungsbereiches repräsentiert.

Siehe auch [Architektur \[OO allg.\]](#)

Fassade [UML]

Eine Fassade ist ein stereotypisiertes Paket, daß ausschließlich Modellelemente referenziert, die in anderen Paketen beheimatet sind.

Siehe auch [Fassade \[OO allg.\]](#) [Paket \[UML\]](#)

Fassade [OO allg.]

Name eines [Entwurfsmusters](#).

Siehe auch [Entwurfsmuster \[OO allg.\]](#)

Fehler [OO allg.]

Ein Fehler ist die Nicht-Erfüllung einer festgelegten Anforderung.

Siehe auch [Test \[OEP\]](#)

Fehlerrate [EmOO]

Die Zahl der während einer spezifischen Aktivität auftretenden Fehler. Die spezifische Aktivität ist z. B. eine Stunde Test, eine Stunde Benutzung, die Durchführung eines Testfalls

Fertigstellungsgrad [EmOO]

Der Fertigstellungsgrad ist der bereits erbrachte Anteil vom Herstellungsaufwand. Ein Fertigstellungsgrad von 70% besagt, daß noch 30% des Herstellungsaufwandes zu leisten sind, bevor das Produkt vollständig hergestellt ist.

Forward-Engineering [OO allg.]

beschreibt die Überführung eines Modells in eine spezielle Programmiersprache, d. h. gewöhnlich die Erzeugung von Code.

Siehe auch [Roundtrip-Engineering \[OO allg.\]](#) [Reverse-Engineering \[OO allg.\]](#)

framework [OO allg.]

Synonym/Akronym/Übersetzung: [Rahmenwerk \[OO allg.\]](#)

Funktionspunkte [EmOO]

Funktionspunkte sind ein extrinsisches [Maß](#) für den von außen sichtbaren Funktionsumfang und damit die Größe eines Softwareprodukts, das keine wesentliche verborgene Funktionalität enthält. Funktionspunkte sind von der International Function Points Users Group [IFPUG94] standardisiert. Vgl.auch [Widgetpunkte](#).

Siehe auch [Widgetpunkte \[EmOO\]](#) [Maß \[EmOO\]](#)

G**Ganze-Teile-Beziehung [UML]**

Synonym/Akronym/Übersetzung: [Aggregation \[UML\]](#)

Gebundenes Element [UML]

Synonym/Akronym/Übersetzung: [Parametrisierte Klasse \[UML\]](#)

Siehe auch [bound element](#) [engl.]

Generalisierung [OO allg.]

Eine Generalisierung (bzw. Spezialisierung) ist eine taxonomische Beziehung zwischen einem allgemeinen und einem speziellen Element (bzw. umgekehrt), wobei das speziellere weitere Eigenschaften hinzufügt, die Semantik erweitert und sich kompatibel zum allgemeinen verhält. Generalisierung und Spezialisierung sind Abstraktionsprinzipien zur hierarchischen Strukturierung der Modellsemantik unter einem diskriminierenden Aspekt.

Die Generalisierung bzw. Spezialisierung wird vor allem auf Klassen und Anwendungsfälle angewendet.

Siehe auch [Anwendungsfall \[UML\]](#) [Klasse \[OO allg.\]](#) [Diskriminator \[UML\]](#) [Vererbung \[OO allg.\]](#) [Spezialisierung \[OO allg.\]](#)

generalization [engl.]

Synonym/Akronym/Übersetzung: [Generalisierung \[OO allg.\]](#)

Generische Klasse [OO allg.]

Synonym/Akronym/Übersetzung: [Parametrisierbare Klasse \[UML\]](#)

Generische Programmierung [OO allg.]

Anwendung von Schablonen (engl. templates), parametrisierbaren Klassen u.ä. bei der Programmierung.

Generisches Design [OO allg.]

Anwendung von Schablonen oder Makros zum Design (in CASE-Tools).

Geordnete Assoziation [UML]

Eine Assoziation, bei der die Objektverbindungen in bestimmter Weise geordnet sind.

Siehe auch [Ordnungszusicherung \[UML\]](#) [Objektverbindung \[UML\]](#) [Assoziation \[UML\]](#)

Gerichtete Assoziation [UML]

Eine Assoziation, bei der von der einen beteiligten Assoziationsrolle zur anderen direkt navigiert werden kann, nicht aber umgekehrt.

Siehe auch [Navigation, Navigierbarkeit \[OO allg.\]](#) [Assoziationsrolle \[UML\]](#) [Assoziation \[UML\]](#)

Geschäftsfall [OO allg.]

Synonym/Akronym/Übersetzung: [Geschäftsvorfall \(Vorgang\) \[OO allg.\]](#)

Geschäftsprozeß [OO allg.]

Ein Geschäftsprozeß ist eine Zusammenfassung von organisatorisch evtl. verteilten, fachlich jedoch zusammenhängenden Aktivitäten, die notwendig sind, um einen Geschäftsvorfall (z.B. einen konkreten Antrag) ergebnisorientiert zu bearbeiten. Die Aktivitäten eines Geschäftsprozesses stehen gewöhnlich in zeitlichen und logischen Abhängigkeiten zueinander. Ein Geschäftsvorfall entsteht gewöhnlich durch ein Ereignis (z.B.

Siehe auch [Geschäftsvorfall \(Vorgang\) \[OO allg.\]](#) [Workflow \[OO allg.\]](#)

Geschäftsvorfall (Vorgang) [OO allg.]

Ein Geschäftsvorfall ist ein (Geschäfts-) Objekt (z.B. ein konkreter Vertrag), das durch ein Ereignis ausgelöst (z.B. Antragseingang) durch die innerhalb eines Geschäftsprozesses beschriebenen Aktivitäten bearbeitet wird.

Siehe auch [Geschäftsprozeß \[OO allg.\]](#)

GUI [OO allg.]

Graphical User Interface, Grafische Benutzeroberfläche.

H**Herstellungsaufwand [EmOO]**

Der [Aufwand](#), der erforderlich ist, um ein bestimmtes Produkt herzustellen.

Siehe auch [Aufwand \[EmOO\]](#)

Herstellungszeit [EmOO]

Die Zeitspanne zwischen dem Beginn der Produkterstellung und ihrem Ende, auch Projektdauer.

Heuristik [OEP]

Eine Heuristik ist eine mit einer Wahrscheinlichkeit behaftete Regel ("Daumenregel").

Hilfsmittelklasse [UML]

Synonym/Akronym/Übersetzung: [utility](#) [engl.]

Hilfsmittelklassen sind Sammlungen von globalen Variablen und Funktionen, die zu einer Klasse zusammengefaßt und dort als Klassenattribute und -operationen definiert sind. Insofern sind Hilfsmittelklassen keine echten Klassen. Das [Stereotyp](#) «utility» kennzeichnet eine Klasse als Hilfsmittelklasse.

Siehe auch [Stereotyp \[UML\]](#) [Klasse \[OO allg.\]](#)

I

Identität [OO allg.]

Synonym/Akronym/Übersetzung: [Objektidentität \[OO allg.\]](#)

Implementierungsvererbung [OO allg.]

Vererbung einer Implementierung an ein spezialisiertes Element. Enthält Schnittstellenvererbung.

Siehe auch [Vererbung \[OO allg.\]](#) [Schnittstellenvererbung \[OO allg.\]](#)

Include-Beziehung [UML]

Synonym/Akronym/Übersetzung: [Enthalten-Beziehung \[UML\]](#)

Eine Beziehung zwischen Anwendungsfällen mit dem Stereotyp «include». Eine Include-Beziehung zwischen Anwendungsfällen besagt, daß innerhalb eines Anwendungsfalles ein anderer vorkommt. Dieses Konstrukt dient dazu, Abschnitte, die in mehreren Anwendungsfällen gleichermaßen vorkommen, zu extrahieren, um so Redundanz zu vermeiden.

Siehe auch [Anwendungsfall \[UML\]](#)

inheritance [engl.]

Synonym/Akronym/Übersetzung: [Vererbung \[OO allg.\]](#)

Inkrement [OEP]

Ein Inkrement ist die Erweiterung eines Produktes. Ein Inkrement ist gewöhnlich gekennzeichnet durch die Differenz zwischen zwei \$Builds.

Siehe auch [Build \[OEP\]](#)

Inkrementelles Vorgehen [OEP]

Eine Vorgehensweise, bei der ein Produkt schrittweise in wachsenden Zwischenprodukten entsteht.

Siehe auch [Inkrement \[OEP\]](#)

Inspektion [OEP]

Eine Sitzung zur Überprüfung einzelner Dokumente und Ergebnisse. Vgl. [Review](#).

Siehe auch [Audit \[OEP\]](#) [Review \[OEP\]](#)

instance [engl.]

Synonym/Akronym/Übersetzung: [Instanz \[OO allg.\]](#)

Instantiierung [OO allg.]

Instantiierung ist das Erzeugen eines Exemplars aus einer Klasse.

Siehe auch [Objekt \[OO allg.\]](#) [Klasse \[OO allg.\]](#)

Instanz [OO allg.]

Für den Hausgebrauch können Instanz, Objekt und Exemplar synonym betrachtet werden.

Siehe auch [Exemplar \[OO allg.\]](#) [Objekt \[OO allg.\]](#)

Integration [OEP]

Zusammenfügen von Teilsystemen oder Komponenten zu einem Gesamtsystem.

Integrationstest [OEP]

Überprüfung, ob die durch eine [Integration](#) zusammengeführten Einzelteile korrekt zusammenarbeiten und als Gesamtheit funktionieren.

Siehe auch [Test \[OEP\]](#) [Integration \[OEP\]](#)

interaction diagram [engl.]

Synonym/Akronym/Übersetzung: [Interaktionsdiagramm \[UML\]](#)

Interaktionsdiagramm [UML]

Synonym/Akronym/Übersetzung: [interaction diagram \[engl.\]](#)

Sammelbegriff für [Sequenzdiagramm](#), [Kollaborationsdiagramm](#), [Aktivitätsdiagramm](#).

Siehe auch [Aktivitätsdiagramm \[UML\]](#) [Kollaborationsdiagramm \[UML\]](#) [Sequenzdiagramm \[UML\]](#)

interface [engl.]

Synonym/Akronym/Übersetzung: [Schnittstelle \[OO allg.\]](#)

Invariante [OO allg.]

Eine Eigenschaft oder ein Ausdruck, der über den gesamten Lebenszeitraum eines Elementes, bspw. eines Objektes gegeben sein muß.

Iteration [OEP]

Eine Iteration entsteht durch die zeitliche Einteilung des Entwicklungsprozesses in mehrere gleichartige Zeitabschnitte (Iterationen). Innerhalb jeder Iteration wird gewöhnlich ein Mikroprozeß durchlaufen (Analyse-Design-Implementierung-Test).

Siehe auch [Inkrement \[OEP\]](#)

Iteratives Vorgehen [OEP]

Ein Vorgehen, bei dem der Entwicklungsprozesses in mehrere gleichartige Zeitabschnitte (Iterationen) geteilt wird.

Siehe auch [Iteration \[OEP\]](#)

K**Kardinalität [OO allg.]**

Anzahl der Elemente.

Siehe auch [Multiplizität \[UML\]](#)

Klasse [OO allg.]

Eine Klasse ist die Definition der Attribute, Operationen und der Semantik für eine Menge von Objekten. Alle Objekte einer Klasse entsprechen dieser Definition.

Siehe auch [Objekt \[OO allg.\]](#) [Operation \[UML\]](#) [Attribut \[OO allg.\]](#)

Klassenattribut [OO allg.]

Klassenattribute gehören nicht einem einzelnen Objekt, sondern sind Attribut einer Klasse und damit für alle Objekte dieser Klasse zugänglich. Einige Programmiersprachen unterstützen Klassenattribute direkt (z. B. Smalltalk), in anderen kann ein vergleichbarer Effekt erzielt werden (z. B. Static-Deklaration in C++).

Siehe auch [Objekt \[OO allg.\]](#) [Attribut \[OO allg.\]](#) [Klasse \[OO allg.\]](#)

Klassenbibliothek [OO allg.]

Eine Klassenbibliothek ist eine Sammlung von Klassen.

Klassendiagramm [UML]

Synonym/Akronym/Übersetzung: [class diagram](#) [engl.]

Ein Klassendiagramm zeigt eine Menge statischer Modellelemente, vor allem Klassen und ihre Beziehungen.

Siehe auch [Klasse](#) [OO allg.]

Klassenkarte [OO allg.]

Synonym/Akronym/Übersetzung: [CRC-Karten](#) [OO allg.]

Wird häufig gleichgesetzt mit CRC-Karte, kann jedoch von dieser dadurch unterschieden werden, daß Klassenkarten eher dem Lösungsraum zugeordnet sind und die Auflistung von Funktionalität (Operationen) und Daten (Attribute) in den Vordergrund stellt, während CRC-Karten eher dem Problemraum zuzuordnen sind, fachliche Begrifflichkeiten definieren und Verantwortlichkeiten und Beteiligte beschreiben.

Klassenmethode [OO allg.]

Synonym/Akronym/Übersetzung: [Klassenoperation](#) [OO allg.]

Klassenoperation [OO allg.]

Klassenoperationen sind Operationen, die nicht auf einem Objekt, sondern auf einer Klasse operieren. Dies ist sinnvoll für Sprachen, in denen Klassen wiederum durch normale Objekte repräsentiert werden, z. B. Smalltalk.

Siehe auch [Objekt](#) [OO allg.] [Operation](#) [UML] [Klasse](#) [OO allg.]

Klassenschablone [OO allg.]

Synonym/Akronym/Übersetzung: [Parametrisierbare Klasse](#) [UML]

Klassenvariable [OO allg.]

Synonym/Akronym/Übersetzung: [Klassenattribut](#) [OO allg.]

Klassenvorlage [OO allg.]

Synonym/Akronym/Übersetzung: [Parametrisierbare Klasse](#) [UML]

Klassifikation [OO allg.]

Einteilung in Klassen.

Siehe auch [Klasse](#) [OO allg.]

Knoten [UML]

Synonym/Akronym/Übersetzung: [node](#) [engl.]

Ein Knoten ist ein physisches Laufzeit-Objekt, das über Rechnerleistung (Prozessor, Speicher) verfügt. Laufzeitobjekte und Komponenten können auf Knoten residieren.

Kollaboration [UML]

Synonym/Akronym/Übersetzung: [collaboration](#) [engl.]

Eine Kollaboration ist der Kontext einer Menge von Interaktionen.

Siehe auch [Kollaborationsdiagramm](#) [UML]

Kollaborationsdiagramm [UML]

Synonym/Akronym/Übersetzung: [collaboration diagram](#) [engl.]

Eine Kollaborationsdiagramm zeigt eine Menge von Interaktionen zwischen einer Menge ausgewählter Objekte in einer bestimmten begrenzten Situation (Kontext) unter Betonung der Beziehungen zwischen den Objekten und ihrer Topographie. Ähnlich dem

Siehe auch [Sequenzdiagramm](#) [UML]

Komponente [UML]

Synonym/Akronym/Übersetzung: [component](#) [engl.]

Eine Komponente ist ein ausführbares Softwaremodul mit eigener Identität und wohldefinierten Schnittstellen (Sourcecode, Binärcode, DLL oder ausführbares Programm). Außerhalb der UML wird Komponente häufig anders, mehr im Sinne eines [Paketes](#) definiert.

Siehe auch [Paket](#) [UML]

Komponentendiagramm [UML]

Synonym/Akronym/Übersetzung: [component diagram](#) [engl.]

Ein Komponentendiagramm zeigt die Organisation und Abhängigkeiten von [Komponenten](#).

Siehe auch [Komponente](#) [UML]

Komposition [UML]

Synonym/Akronym/Übersetzung: [composition](#) [engl.]

Eine Komposition ist eine strenge Form der Aggregation, bei der die Teile vom Ganzen existenzabhängig sind.

Siehe auch [Aggregation](#) [UML]

Konfiguration [EmOO]

Eine Konfiguration ist die Gesamtheit zusammenpassender Software-Elemente.

Siehe auch [Konfigurationsmanagement](#) [OEP]

Konfigurationsdiagramm [UML]

Synonym/Akronym/Übersetzung: [Einsatz- und Verteilungsdiagramm](#) [UML]

Konfigurationseinheit [EmOO]

Eine Konfigurationseinheit nennt man jeden einzeln versionierten und verwalteten Teil einer Software. In der Regel ist ein Entwickler für eine Konfigurationseinheit verantwortlich.

Siehe auch [Konfigurationsmanagement](#) [OEP]

Konfigurationskontrollsystem [EmOO]

Konfigurationskontrollsystem nennt man die Menge der Verfahren, Mechanismen und Werkzeuge, mit denen Konfigurationseinheiten verwaltet werden.

Siehe auch [Konfigurationseinheit](#) [EmOO]

Konfigurationsmanagement [OEP]

Vorgehensweise zur Überwachung und Kontrolle der unvermeidlichen Programmänderungen. Sie erfordert eine Festlegung der System- und Programmbestandteile einschließlich der Dokumentation von Änderungsprozeduren, Programmversionen und Freigabeprozeduren sowie eine laufende Verfolgung des jeweiligen Programmstatus

Siehe auch [Konfigurationsverwaltung](#) [EmOO]

Konfigurationsverwaltung [EmOO]

Die Konfigurationsverwaltung von Software ist die Gesamtheit der Verfahren zur eindeutigen Kennzeichnung der Konfiguration eines Softwaresystems zu bestimmten Zeitpunkten des Software-Lebenszyklus mit dem Zweck, alle Änderungen dieser Konfiguration systematisch zu überwachen, die Konsistenz des Softwaresystems sicherzustellen und jederzeit die Möglichkeit der Rückverfolgung anzubieten.

Siehe auch [Konfigurationsmanagement](#) [OEP]

Konkrete Klasse [OO allg.]

Eine Klasse, die Objekte instantieren kann. Vgl. [Abstrakte Klasse](#).

Siehe auch [Abstrakte Klasse \[OO allg.\]](#) [Objekt \[OO allg.\]](#) [Klasse \[OO allg.\]](#)

Konkretisierung [OO allg.]

Synonym/Akronym/Übersetzung: [Spezialisierung \[OO allg.\]](#)

Konsistenzsicherung [OO allg.]

Synonym/Akronym/Übersetzung: [Zusicherung \[UML\]](#)

Eine Zusicherung zwischen mehreren Assoziationen, die teilweise redundante Sachverhalte repräsentieren. Die Zusicherung gibt die Konsistenzbedingung an.

Siehe auch [Assoziation \[UML\]](#)

L**Legacy System [OO allg.]**

Synonym für ein schwer wartbares, schwer anpaßbares oder inkompatibles Alt-System.

Lines of Code (LOC) [EmOO]

LOC ist ein intrinsisches [Maß](#) für die Größe eines Software-Programms. LOC mißt die Zeilen der textuellen Quelle (nicht dagegen die Zahl der Anweisungen). Wichtig ist noch die Unterscheidung in Brutto-LOC (inklusive aller Kommentare und Leerzeilen) und Netto-LOC (exklusive aller Kommentare und Leerzeilen).

Siehe auch [Maß \[EmOO\]](#)

link [engl.]

Synonym/Akronym/Übersetzung: [Objektbeziehung \[UML\]](#)

M**Makroschätzung [EmOO]**

Die [Schätzung](#) des Herstellungsaufwandes anhand globaler Parameter von Projekt und Produkt. Sie benutzt wesentlich die Produktgröße, um die Herstellungszeit und/oder die notwendige Projektbesetzung zu prognostizieren. Eine Makroschätzung setzt also eine Produktschätzung voraus.

Siehe auch [Schätzung \[EmOO\]](#)

Maß [EmOO]

Ein Maß dient dazu, Größen und Mengen festzustellen. Bei der [Messung](#) der [Produktgröße](#) von Software unterscheidet man zwischen intrinsischen und extrinsischen Maßen. Intrinsische Maße beziehen sich auf innere, von der Software-Produktionsumgebung abhängige Eigenschaften und sind deshalb nicht zwischen verschiedenen Umgebungen übertragbar. Extrinsische oder äußere Maße abstrahieren davon.

Mehrfachvererbung [OO allg.]

Synonym/Akronym/Übersetzung: [Multiple Vererbung \[OO allg.\]](#)

Siehe auch [Vererbung \[OO allg.\]](#)

Mehrgliedrige Assoziation [UML]

Eine Assoziation, an der mehr als zwei Assoziationsrollen beteiligt sind.

Siehe auch [Binäre Assoziation \[UML\]](#) [Assoziationsrolle \[UML\]](#) [Assoziation \[UML\]](#)

Meilenstein [OEP]

Ein Meilenstein definiert einen Termin, zu dem eine Menge von Ergebnissen in vorgegebener Detaillierung und Vollständigkeit nachprüfbar und formal dokumentiert vorliegen soll. Ein Meilenstein ist ein Hilfsmittel zur Planung und Überwachung eines Projektes.

Siehe auch [Phase \[OEP\]](#)

Merkmal [UML]

Synonym/Akronym/Übersetzung: [Eigenschaftswert \[UML\]](#)

message [engl.]

Synonym/Akronym/Übersetzung: [Nachricht \[OO allg.\]](#)

Messung [EmOO]

Die Bestimmung einer Größe oder Menge durch Vergleich mit einem [Maß](#). Das Ergebnis einer Messung wird als Produkt von Maßzahl und Maß angegeben.

Siehe auch [Maß \[EmOO\]](#)

meta class [engl.]

Synonym/Akronym/Übersetzung: [Metaklasse \[OO allg.\]](#)

Metaklasse [OO allg.]

Eine Metaklasse ist eine Klasse, deren Exemplare wiederum Klassen sind. Dieses Konzept existiert nur in einigen objektorientierten Sprachen (z. B. in Smalltalk).

Siehe auch [Klasse \[OO allg.\]](#)

Metamodell [OO allg.]

Ein Modell, das die Sprache definiert, mit der ein Modell definiert werden kann.

Metatyp [UML]

Ein Metatyp ist ein Typ (eine Klasse), dessen Instanzen Untertypen (Unterklassen) eines anderen Typs (einer anderen Klasse) darstellen.

method [engl.]

Synonym/Akronym/Übersetzung: [Methode \[UML\]](#)

Siehe auch [Methode \[OEP\]](#)

Methode [OEP]

Eine Methode ist eine Handlungsvorschrift die beschreibt, wie ein Ziel bzw. Ergebnis unter gegebenen Bedingungen erreicht werden kann.

Methode [UML]

In der UML wird eine Methode als Implementierung einer Operation definiert. Für die Praxis ist es unkritisch, Methode und Operation synonym zu verwenden. In Smalltalk werden Operationen Methoden genannt.

Siehe auch [Operation \[UML\]](#)

Methodik [EmOO]

Synonym/Akronym/Übersetzung: [Methode \[OEP\]](#)

Bedeutsam klingendes Synonym für Methode.

Siehe auch [Methode \[OEP\]](#)

Methodologie [EmOO]

Methodologie ist die Lehre von den Methoden.

Siehe auch [Methodik \[EmOO\]](#) [Methode \[OEP\]](#)

Metrik [EmOO]

Eine Meßmethode. Zu einer Metrik oder Meßmethode gehört ein [Maß](#) und eine Anleitung, wie bei der Messung zu verfahren ist.

Mikroschätzung [EmOO]

Die Schätzung eines [Herstellungsaufwandes](#) anhand detaillierter Eigenschaften von Projekt und Produkt. Zu den Eigenschaften zählen sowohl die Summe aller Aktivitäten im Projekt, die [Projektbesetzung](#) und Struktur als auch eine Zerlegung des Produkts in solche Teile, die innerhalb einzelner Aktivitäten erstellt werden können.

Siehe auch [Projektbesetzung \[EmOO\]](#) [Herstellungsaufwand \[EmOO\]](#)

Multiple Klassifikation [OO allg.]

Ein Objekt ist zur gleichen Zeit Instanz mehrerer Klassen (nicht möglich in C++, Java, Smalltalk).

Siehe auch [Klassifikation \[OO allg.\]](#) [Instantiierung \[OO allg.\]](#) [Klasse \[OO allg.\]](#) [Objekt \[OO allg.\]](#) [Instanz \[OO allg.\]](#)

Multiple Vererbung [OO allg.]

Synonym/Akronym/Übersetzung: [Mehrfachvererbung \[OO allg.\]](#)

Multiplizität [UML]

Bereich erlaubter [Kardinalitäten](#).

Siehe auch [Schnittstelle \[OO allg.\]](#)

n**n-ary association [UML]**

Synonym/Akronym/Übersetzung: [Mehrgliedrige Assoziation \[UML\]](#)

Nachbedingung [OO allg.]

Eine Nachbedingung beschreibt einen Zustand, der nach Abschluß einer Operation o.ä. gegeben sein muß.

Siehe auch [Zusicherung \[UML\]](#)

Nachricht [OO allg.]

Nachrichten sind ein Mechanismus, mit dem Objekte untereinander kommunizieren können. Eine Nachricht überbringt einem Objekt die Information darüber, welche Aktivität von ihm erwartet wird, d.h. eine Nachricht fordert ein Objekt zur Ausführung einer Operation auf. Eine Nachricht besteht aus einem Selektor (einem Namen), einer Liste von Argumenten und geht genau an einem Empfänger. Der Sender einer Nachricht erhält ggf. ein Antwort-Objekt zurück. Durch [Polymorphismus](#) kann eine Nachricht zum Aufruf einer von mehreren gleichlautenden [Operation](#)en führen.

Siehe auch [Methode \[OEP\]](#) [Operation \[UML\]](#)

Navigation, Navigierbarkeit [OO allg.]

Navigation ist die Betrachtung von Zugriffsmöglichkeiten auf Objekte (und ihre Attribute und Operationen) innerhalb eines Objektnetzes. Direkt navigierbar werden solche Zugriffe genannt, die ohne Umwege möglich sind.

Siehe auch [Navigationsangaben \[OO allg.\]](#)

Navigationsangaben [OO allg.]

Navigationsangaben sind Spezifikationen zur Navigation, d.h. Beschreibungen von Zugriffspfaden und -einschränkungen und der daraus resultierenden Zugriffsergebnisse (beispielsweise mit Hilfe der [OCL](#)).

Siehe auch [Object Constraint Language \(OCL\) \[UML\]](#)

Nebenläufigkeit [OO allg.]

Zwei oder mehr Aktivitäten werden zeitgleich (parallel) ausgeführt.

node [engl.]

Synonym/Akronym/Übersetzung: [Knoten \[UML\]](#)

Norm [OEP]

Eine Norm ist die Vorgabe eines Handlungsmusters, daß befolgt, oder einer Qualität, die eingehalten werden muß. Sie dient dazu, unabhängig voneinander entstehende Artefakte mit einheitlichen Eigenschaften oder Qualitäten herzustellen. Vgl. [Richtlinie](#).

Siehe auch [Richtlinie \[OEP\]](#)

Notiz [UML]

Kommentare bzw. Annotationen zu einem Diagramm oder einem oder mehreren beliebigen Modellelementen ohne semantische Wirkung.

O**Oberklasse [OO allg.]**

Eine Oberklasse ist eine Verallgemeinerung ausgewählter Eigenschaften ihrer [Unterklasse\(n\)](#).

Siehe auch [Unterklasse \[OO allg.\]](#) [Generalisierung \[OO allg.\]](#)

Oberzustand [UML]

Ein Oberzustand enthält andere Zustände bzw. Unterzustände.

Siehe auch [Zustand \[UML\]](#)

Object Constraint Language (OCL) [UML]

Die OCL definiert eine Sprache zu Beschreibung von Zusicherungen, Invarianten, Vor- und Nachbedingungen und Navigation innerhalb von UML-Modellen.

Siehe auch [Navigation, Navigierbarkeit \[OO allg.\]](#) [Nachbedingung \[OO allg.\]](#) [Vorbedingung \[OO allg.\]](#) [Invariante \[OO allg.\]](#) [Zusicherung \[UML\]](#)

object diagram [engl.]

Synonym/Akronym/Übersetzung: [Objektdiagramm \[UML\]](#)

Object Engineering Process (OEP) [OEP]

Ein Leitfaden zur Planung und Durchführung objektorientierter, UML-basierter Softwareentwicklungsprojekte. Kommerzielles Produkt der Fa. oose.de GmbH.

object lifeline [engl.]

Synonym/Akronym/Übersetzung: [Objekt-Lebenslinie \[UML\]](#)

Objekt [OO allg.]

Ein Objekt ist eine konkret vorhandene und agierende Einheit mit eigener Identität und definierten Grenzen das Zustand und Verhalten kapselt. Der Zustand wird repräsentiert durch die [Attribute](#) und Beziehungen, das Verhalten durch Operationen bzw. Methoden. Jedes Objekt ist ein Exemplar (Synonym: Instanz) einer Klasse. Das definierte Verhalten gilt für alle Objekte einer Klasse gleichermaßen, ebenso die Struktur ihrer Attribute. Die Werte der Attribute sind jedoch individuell für jedes Objekt. Jedes Objekt hat eine eigene, von seinen Attributwerten unabhängige, nicht veränderbare Identität.

Siehe auch [Objektidentität \[OO allg.\]](#) [Objektverbindung \[UML\]](#) [Operation \[UML\]](#) [Attribut \[OO allg.\]](#)

Objekt-Lebenslinie [UML]

Eine Linie in einem Sequenzdiagramm, die den Lebenszeitraum eines Objektes darstellt.

Siehe auch [Sequenzdiagramm \[UML\]](#)

Objektbasiert [OO allg.]

Eine Programmiersprache, Datenbank o. a. wird als objektbasiert bezeichnet, wenn sie das Konzept der Datenabstraktion unterstützt, weitergehende Konzepte wie Klassen, Vererbung, Polymorphismus etc. aber teilweise oder vollständig fehlen.

Siehe auch [Datenabstraktion \[OO allg.\]](#) [Polymorphismus \[OO allg.\]](#) [Vererbung \[OO allg.\]](#) [Klasse \[OO allg.\]](#)

Objektbeziehung [UML]

Synonym/Akronym/Übersetzung: [Objektverbindung \[UML\]](#)

Objektdiagramm [UML]

Synonym/Akronym/Übersetzung: [object diagram](#) [engl.]

Ein Diagramm, das Objekte und ihre Beziehungen untereinander zu einem bestimmten Zeitpunkt zeigt. Gewöhnlich ein [Kollaborationsdiagramm](#) oder eine spezielle Variante des [Klassendiagramms](#).

Siehe auch [Klassendiagramm \[UML\]](#) [Kollaborationsdiagramm \[UML\]](#)

Objektidentität [OO allg.]

Objektidentität ist eine Eigenschaft, die ein Objekt von allen anderen unterscheidet, auch wenn es möglicherweise die gleichen Attributwerte besitzt.

Objektorientierte Programmiersprache [OO allg.]

Objektorientierte Programmiersprachen stellen u. a. folgende Basiskonzept bereit:

- Objekte sind abstrakte Einheiten.
- Objekte sind Exemplare einer Klasse, d.h. von einer Klasse abgeleitet.
- Klassen vererben ihre Eigenschaften und bilden so eine Vererbungshierarchie.
- Auf Objekte wird dynamisch verwiesen, d. h. die Bindung ist dynamisch und ermöglicht so Polymorphismus.

Siehe auch [Dynamische Bindung \[OO allg.\]](#) [Polymorphismus \[OO allg.\]](#) [Vererbung \[OO allg.\]](#) [Klasse \[OO allg.\]](#) [Objekt \[OO allg.\]](#)

Objektverbindung [UML]

Synonym/Akronym/Übersetzung: [link](#) [engl.]

Eine konkrete Beziehung zwischen zwei Objekten, d.h. die Instanz einer [Assoziation](#). Ein Objekt hat eine Beziehung zu einem anderen Objekt, wenn es eine Referenz darauf besitzt. Implementiert werden diese Referenzen gewöhnlich durch [Attribute](#), was für die Modellierung jedoch unerheblich ist.

Siehe auch [Attribut \[OO allg.\]](#) [Assoziation \[UML\]](#)

OCL [UML]

Synonym/Akronym/Übersetzung: [Object Constraint Language \(OCL\) \[UML\]](#)

Oder-Zusicherung [UML]

Eine Zusicherung zwischen Assoziationen, die alle von einer gemeinsamen Klasse zu verschiedenen anderen führen und für die definiert ist, daß ein Objekt der gemeinsamen Klasse stets nur Objektverbindungen zu genau einer der übrigen Klassen (präziser: Assoziationsrollen) unterhalten (exklusives Oder).

Siehe auch [Assoziationsrolle \[UML\]](#) [Objekt \[OO allg.\]](#) [Klasse \[OO allg.\]](#) [Assoziation \[UML\]](#) [Zusicherung \[UML\]](#)

OEP [OEP]

Synonym/Akronym/Übersetzung: [Object Engineering Process \(OEP\) \[OEP\]](#)

OMG [OO allg.]

OMG ist die Abkürzung für Object Management Group.

OO [OO allg.]

OO ist die Abkürzung für Objektorientierung.

Operation [UML]

Operationen sind Dienstleistungen, die von einem Objekt mit einer Nachricht angefordert werden können, um ein bestimmtes Verhalten zu bewirken. Sie werden implementiert durch Methoden. In der Praxis werden Operation und Methode häufig synonym verwendet.

Siehe auch [Methode \[UML\]](#) [Nachricht \[OO allg.\]](#)

operation [engl.]

Synonym/Akronym/Übersetzung: [Operation \[UML\]](#)

Ordnungszusicherung [UML]

Eine Zusicherung zu einer Assoziation, die angibt, daß ihre Elemente (Objektverbindungen) in bestimmter Weise geordnet sind.

Siehe auch [Geordnete Assoziation \[UML\]](#) [Assoziation \[UML\]](#) [Objektverbindung \[UML\]](#) [Zusicherung \[UML\]](#)

Organisationseinheit [OEP]

Eine Gruppe von Personen innerhalb eines Unternehmens, die eine definierte Rolle wahrnehmen. Die Organisationseinheit ist ein Element der Aufbauorganisation des Unternehmens. Die unterste Ebene ist gewöhnlich die Abteilung.

p

package [engl.]

Synonym/Akronym/Übersetzung: [Paket \[UML\]](#)

Paket [UML]

Synonym/Akronym/Übersetzung: [package](#) [engl.]

Pakete sind Ansammlungen von Modellelementen beliebigen Typs, mit denen das Gesamtmodell in kleinere überschaubare Einheiten gegliedert wird. Ein Paket definiert einen Namensraum, d.h. innerhalb eines Paketes müssen die Namen der enthaltenen Elemente sein. Jedes Modellelement kann in anderen Paketen referenziert werden, gehört aber zu genau einem (Heimat-) Paket. Pakete können wiederum Pakete beinhalten. Das oberste Paket beinhaltet das gesamte System.

Parameter [UML]

Ein Parameter ist die Spezifikation einer Variablen, die Operationen, Nachrichten oder Ereignissen mitgegeben, von diesen verändert oder zurückgegeben wird. Ein Parameter kann aus einem Namen, einem Typ (einer Klasse) und einer Übergaberichtung (in, out, inout) bestehen.

parameterized class [engl.]

Synonym/Akronym/Übersetzung: [Parametrisierbare Klasse](#) [UML]

Parametrisierbare Klasse [UML]

Eine parametrisierbare Klasse ist eine mit generischen formalen Parametern versehene Schablone, mit der gewöhnliche (d.h. nicht-generische) Klassen erzeugt werden können. Die generischen Parameter dienen als Stellvertreter für die aktuellen Parameter, die Klassen oder einfache Datentypen repräsentieren.

Parametrisierte Klasse [UML]

Als parametrisierte Klasse wird die Instanz einer parametrisierbaren Klasse bezeichnet, d. h. das Ergebnis einer konkreten Parametrisierung.

Siehe auch [Parametrisierbare Klasse](#) [UML]

Partition [UML]

Eine Partition ist die Gesamtheit der Subklassen, die auf dem selben Diskriminator beruhen.

Siehe auch [Subklasse](#) [OO allg.] [Diskriminator](#) [UML]

pattern [engl.]

Synonym/Akronym/Übersetzung: [Entwurfsmuster](#) [OO allg.]

Persistentes Objekt [OO allg.]

Persistente Objekte (persistent: lat. "anhaltend") sind solche, deren Lebensdauer über die Laufzeit einer Programmsitzung hinausreicht. Die Objekte werden hierzu auf nichtflüchtigen Speichermedien (z.B. Datenbanken) gehalten.

Siehe auch [Transientes Objekt](#) [OO allg.]

Phase [OEP]

Eine Phase ist ein zeitlicher bzw. sachlogischer Gliederungsabschnitt eines Projektes. Eine Phase faßt eine Menge von Aktivitäten und Ergebnissen zu einer Planungs- und Kontrolleinheit zusammen. Am Ende jeder Phase steht ein [Meilenstein](#), der die in der Phase zu erzielenden Inhalte definiert.

Siehe auch [Iteration](#) [OEP] [Meilenstein](#) [OEP]

Pilot, Pilotprojekt [OEP]

Ein Pilot ist ein mit einer bestimmten Verfahrensweise oder Architektur erstmals erstelltes vollständiges Ergebnis, mit dem gewöhnlich die Brauchbarkeit nachgewiesen werden soll. Während [Prototypen](#) nur ausgewählte Aspekte des endgültigen Produktes repräsentieren, ist ein Pilot ein vollständiges und rahmenbedingungs-konformes Ergebnis. Ein Pilotprojekt ist ein Projekt, mit dem ein Pilot erstellt werden soll.

Siehe auch [Prototyp](#) [OEP]

Polymorphismus [OO allg.]

Polymorphismus (Vielgestaltigkeit) heißt, daß gleichlautende Nachrichten an kompatible Objekte unterschiedlicher Klassen ein unterschiedliches Verhalten bewirken können. Beim dynamischen Polymorphismus wird eine Nachricht nicht zur Compilierzeit, sondern erst beim Empfang zur Programmaufzeit einer konkreten Operation zugeordnet. Voraussetzung hierfür ist das dynamische Binden.

postcondition [engl.]

Synonym/Akronym/Übersetzung: [Nachbedingung](#) [OO allg.]

precondition [engl.]

Synonym/Akronym/Übersetzung: [Vorbedingung](#) [OO allg.]

Primitive Datentypen [OO allg.]

Primitive Datentypen fassen fachlich neutrale Basisfunktionalität zusammen, die in verschiedenen fachlichen Kontexten verwendet werden kann, beispielsweise Zeitraum, Geld oder Adresse. Es kann sich hierbei um solche handeln, die bereits durch die Programmiersprache und Entwicklungsumgebung bereitgestellt werden (String, Integer etc.) oder um solche, die unternehmens- bzw. projektspezifisch benötigt werden.

Primitive Datentypen unterscheiden sich von sonstigen Objekten dadurch, daß sie per Wert und nicht per Referenz weitergeleitet werden.

Problembereich [OEP]

Anwendungsgebiet bzw. Problembereich, innerhalb dessen die fachliche Modellierung stattfindet. Als Problembereichsmodell (Domänenmodell) wird in der Regel der Teil des Gesamtmodells verstanden, der sich auf den eigentlichen fachlichen Problembereich bezieht (auch fachliches Modell genannt). Technische, querschnittliche u. ä. Aspekte gehören nicht dazu.

Siehe auch [Domäne](#) [OEP]

Produktcontrolling [EmOO]

Produktcontrolling ist das Nachprüfen und die daraus abgeleitete Prognose qualitativer Produkteigenschaften. Zu den qualitativen Produkteigenschaften zählen neben anderen Qualitätsmaßen die [Fehlerrate](#) und die [Produktgröße](#).

Siehe auch [Produktgröße](#) [EmOO] [Fehlerrate](#) [EmOO]

Produktgröße [EmOO]

Produktgröße wird entweder intrinsisch, dann in LOC, oder extrinsisch, dann meist in [Funktionspunkten](#), gemessen. Siehe auch [Maß](#). In diesem Buch werden die [Widgetpunkte](#) als weiteres extrinsisches Maß für die Produktgröße vorgeschlagen.

Siehe auch [Widgetpunkte](#) [EmOO] [Maß](#) [EmOO] [Funktionspunkte](#) [EmOO] [Lines of Code \(LOC\)](#) [EmOO]

Produktivität [EmOO]

Produktivität ist der Quotient aus Produktgröße und Herstellungsaufwand. Da die Produktgröße extrinsisch oder intrinsisch angegeben werden kann, gibt es auch zwei "Arten" von Produktivität: Die intrinsische oder LOC-Produktivität und die "richtige" extrinsische Produktivität, die eine Messung der Produktgröße in Funktionspunkten oder Widgetpunkten voraussetzt.

Siehe auch [Widgetpunkte](#) [EmOO] [Lines of Code \(LOC\)](#) [EmOO]

Projekt [OEP]

Ein Projekt ist ein einmaliges Vorhaben, daß zeitlich, finanziell und personell begrenzt ist und zur Erreichung eines definierten Zieles geschaffen wird. Ein Projekt verfügt über eine projektspezifische Organisation.

Projektbesetzung [EmOO]

Projektbesetzung ist eine Kurve über der Zeit, die angibt, wieviele Personen zur Zeit in einem Projekt arbeiten. Der Aufwand eines Projektes, z. B. auch der [Herstellungsaufwand](#) eines Produktes, ergibt sich als Integral der Projektbesetzung über die [Herstellungszeit](#).

Siehe auch [Herstellungszeit](#) [EmOO] [Herstellungsaufwand](#) [EmOO]

Projektcontrolling [EmOO]

Projektcontrolling ist das Nachprüfen des erbrachten [Aufwands](#), der Vergleich mit dem [Fertigstellungsgrad](#) und die daraus abgeleitete Prognose für den weiteren Verlauf des Projekts. Es liefert damit sozusagen die Positionsbestimmung des Projektleiters.

Siehe auch [Fertigstellungsgrad](#) [EmOO] [Aufwand](#) [EmOO]

Projektmanagement [OEP]

Gesamtheit der Aufgaben und Techniken zur Führung und Organisation eines Projektes.

Projektorganisation [OEP]

Gesamtheit der Organisationseinheiten eines Projektes. Gewöhnlich setzt sich die Projektorganisation aus Bestandteilen der vorhandenen Betriebsorganisation zusammen.

Projektplan [OEP]

Ein Plan der den grundsätzlichen "Fahrplan" des Projektes darstellt, er enthält [Meilensteine](#) und beschreibt die vorgesehenen [Phasen](#) und [Iterationen](#) des Projektes.

Siehe auch [Softwareentwicklungsplan \[OEP\]](#) [Iteration \[OEP\]](#) [Phase \[OEP\]](#) [Meilenstein \[OEP\]](#)

Propagation [OO allg.]

Ausdehnung der Eigenschaften einer Klasse durch Verwendung von Operationen anderer Klassen.

Siehe auch [Delegation \[OO allg.\]](#)

property [engl.]

Synonym/Akronym/Übersetzung: [Eigenschaftswert](#) [UML]

Protokoll [OO allg.]

Eine Menge von [Signaturen](#).

Siehe auch [Signatur \[OO allg.\]](#)

Prototyp [OEP]

Ein Prototyp ist ein vorläufiges oder temporäres Produkt, mit dem ausgewählte Eigenschaften oder Aspekte des zu entwickelnden endgültigen Produktes erfahrbar und beurteilbar gemacht werden sollen. Prototypen können explorativ sein, d. h. zur Erforschung eines Sachverhaltes dienen, sie können experimentell sein, d. h. zur Überprüfung der Machbarkeit oder Funktionsfähigkeit dienen, oder evolutionär sein, d. h. vorab ein Teilprodukt bereitstellen.

Prototyping [OEP]

Herstellung eines [Prototyps](#).

Siehe auch [Prototyp \[OEP\]](#)

Prozeßmodell [OEP]

Synonym/Akronym/Übersetzung: [Vorgehensmodell \[OEP\]](#)

Prozeßmodell kann gewöhnlich mit Vorgehensmodell gleichgesetzt werden, wobei der Begriff Vorgehensmodell häufiger für die Durchführung von Aktivitäten durch Menschen gebraucht wird und Prozeßmodell häufiger für die automatisierte Durchführung von Aktivitäten.

Q**Qualifizierendes Attribut [UML]**

Das Attribut, über welches in einer Assoziation der zugriff auf die gegenüberliegende Seite erfolgt. Das qualifizierende Attribut ist definiert als Teil der Assoziation, jedoch muß in der Klasse, auf die darüber zugegriffen wird, dieses Attribut definiert sein.

Siehe auch [Qualifizierte Assoziation \[UML\]](#)

Qualifizierte Assoziation [UML]

Eine qualifizierte Assoziation ist eine Assoziation, bei der die referenzierte Menge der Objekte durch qualifizierende Attribute in Partitionen unterteilt wird, wobei vom Ausgangsobjekt aus betrachtet jede Partition nur einmal vorkommen kann.

Siehe auch [Qualifizierendes Attribut \[UML\]](#)

Qualitätssicherung [OEP]

Maßnahmen, die zur Erlangung, Steigerung oder Kontrolle von Qualität beitragen. Dazu gehören planerische Maßnahmen, konstruktive Maßnahmen (Werkzeuge, Richtlinien, Konventionen etc.), analytische Maßnahmen (Test, Validierung, Verifizierung).

R**Rahmenwerk [OO allg.]**

Ein Rahmenwerk ist eine Menge kooperierender Klassen, die unter Vorgabe eines Ablaufes ("Don't call the framework, the framework calls you") eine generische Lösung für eine Reihe ähnlicher Aufgabenstellungen bereitstellen.

Rational Unified Process (RUP) [OO allg.]

Ein objektorientiertes, UML-basiertes Vorgehensmodell, siehe [Kruchten98]. Kommerzielles Produkt der Fa. Rational Software.

Realisierungsbeziehung [UML]

Eine Realisierungsbeziehung ist eine Beziehung zwischen einer Schnittstelle und ihrer Implementierung.

Siehe auch [Verfeinerungsbeziehung \[UML\]](#)

Refaktorisierung [EmOO]

Umstrukturierung eines Systems zugunsten besserer Wart- oder Erweiterbarkeit ohne Änderung der Funktionalität.

Referentielle Integrität [OO allg.]

Regel, die die Integrität von Objektbeziehungen beschreibt, vor allem für den Fall, daß eines der beteiligten Objekte oder die Objektverbindung selbst gelöscht werden sollen.

refinement [engl.]

Synonym/Akronym/Übersetzung: [Verfeinerungsbeziehung](#) [UML]

Regressionstest [EmOO]

Hiermit wird die Wiederholung früherer Tests bezeichnet. Er dient dazu, nachzuweisen, daß bereits zu einem früheren Zeitpunkt korrekt vorhandene Systemfunktionalität immer noch vorhanden ist, auch wenn zwischenzeitlich neue Funktionalität hinzugekommen ist.

Siehe auch [Test](#) [OEP]

relationship [engl.]

Synonym/Akronym/Übersetzung: [Beziehung](#) [UML]

Release [OEP]

Ein [Build](#), das an externe Anwender ausgeliefert werden kann.

Siehe auch [Build \[OEP\]](#)

Resemble-Beziehung [UML]

Synonym/Akronym/Übersetzung: [Ähnlichkeitsbeziehung \[UML\]](#)

Eine Beziehung zwischen Anwendungsfällen mit dem Stereotyp «resemble». Eine Resemble-Beziehung zwischen Anwendungsfällen besagt, daß zwei Anwendungsfälle in ihrer Beschreibung viele Ähnlichkeiten besitzen.

Siehe auch [Anwendungsfall \[UML\]](#)

Restaufwandsschätzung [EmOO]

Die Schätzung des noch verbleibenden [Herstellungsaufwandes](#) zu einem fortgeschrittenen Zeitpunkt im Projekt, das ein bestimmtes Produkt herstellt.

Siehe auch [Herstellungsaufwand \[EmOO\]](#)

Reverse-Engineering [OO allg.]

Hiermit bezeichnet man einen Vorgang, bei dem aus vorhandenem Programmiersprachen-Code ein (visuelles) Modell erzeugt werden soll.

Siehe auch [Roundtrip-Engineering \[OO allg.\]](#)

Review [OEP]

Ein Review ist eine Sitzung zur kritischen Begutachtung des aktuellen Projektstands auf Basis repräsentativer (Teil-)Ergebnisse mit dem Ziel, den Status zu ermitteln und mögliche Beiträge zur weiteren Problemlösung aufzunehmen.

Siehe auch [Inspektion \[OEP\]](#) [Audit \[OEP\]](#)

Richtlinie [OEP]

Eine Richtlinie ist die Vorgabe eines Handlungsmusters, daß befolgt, oder einer Qualität, die eingehalten werden sollte. Sie dient dazu, unabhängig voneinander entstehende [Artefakte](#) mit einheitlichen, ähnlichen oder vergleichbaren Eigenschaften oder Qualitäten herzustellen. Vgl. [Norm](#).

Siehe auch [Artefakt \[OEP\]](#) [Norm \[OEP\]](#)

role [engl.]

Synonym/Akronym/Übersetzung: [Rolle](#) [UML]

Rolle [UML]

Synonym/Akronym/Übersetzung: [Assoziationsrolle](#) [UML]

Rolle [OEP]

Synonym/Akronym/Übersetzung: [Akteur \[OEP\]](#)

Eine Rolle ist ein abstrakter Platzhalter für jemanden, der an der Durchführung einer Aktivität mitwirkt, wobei ausgedrückt wird, in welcher Rolle er dabei mitwirkt. Der Begriff Rolle hat sich innerhalb von Vorgehensmodellen etabliert, während im Rahmen der Anwendungsfallanalyse eher von Akteuren die Rede ist.

Siehe auch [Akteur](#) [UML] [Aktivität \[OEP\]](#)

Roundtrip-Engineering [OO allg.]

Hiermit bezeichnet man den Versuch, durch abwechselndes Forward- und Reverse-Engineering Code und Modell permanent oder regelmäßig konsistent zu halten.

Siehe auch [Reverse-Engineering \[OO allg.\]](#) [Forward-Engineering \[OO allg.\]](#)

RUP [OEP]

RUP ist die Abkürzung für Rational Unified Process.

Siehe auch [Rational Unified Process \(RUP\)](#) [OO allg.]

S**Sammlung [OO allg.]**

Sammlungen sind Objekte, die eine Menge anderer Objekte referenzieren und die Operationen bereitstellen,

um auf diese Objekte zuzugreifen.

Siehe auch [Objekt \[OO allg.\]](#)

Schablone [OO allg.]

Synonym/Akronym/Übersetzung: [Parametrisierbare Klasse](#) [UML]

Schätzung [EmOO]

Eine Schätzung ist eine Prognose. Man kann die Größe eines Produktes schätzen oder mit einer Projektschätzung den erforderlichen Aufwand zur Herstellung des Produktes.

Schnittstelle [OO allg.]

Schnittstellen beschreiben einen ausgewählten Teil des extern sichtbaren Verhaltens von Modellelementen (hauptsächlich von Klassen und Komponenten), d.h. eine Menge von Signaturen.

Siehe auch [Schnittstellenklassen](#) [UML]

Schnittstellenklassen [UML]

Schnittstellenklassen sind abstrakte Klassen (genauer: Typen), die ausschließlich abstrakte Operationen definieren. Schnittstellenklassen sind Klassen, die mit dem [Stereotyp](#) «interface» gekennzeichnet sind. Sie sind Spezifikationen des extern sichtbaren Verhaltens von Klassen und beinhalten eine Menge von [Signaturen](#) für Operationen, die Klassen, die diese Schnittstelle bereitstellen wollen, implementieren müssen.

Siehe auch [Operation \[UML\]](#) [Stereotyp \[UML\]](#) [Klasse](#) [OO allg.]

Schnittstellenvererbung [OO allg.]

Innerhalb einer Spezialisierungsbeziehung wird lediglich eine [Schnittstelle](#) vererbt.

Siehe auch [Spezialisierung \[OO allg.\]](#)

Selbstdelegation [OO allg.]

Zur Ausführung einer Operation wird eine Teilaufgabe an eine andere Operation der selben Klasse delegiert (d. h. ein Objekt sendet sich selbst eine Nachricht).

Siehe auch [Objekt \[OO allg.\]](#) [Klasse \[OO allg.\]](#) [Nachricht \[OO allg.\]](#) [Operation](#) [UML]

self [OO allg.]

Self (Smalltalk) und *this* (Java, C++) sind vordefinierte Programmiersprachen-Schlüsselwörter. Mit *self* bzw. *this* kann ein Objekt sich selbst eine Nachricht senden, d. h. es ruft eine seiner eigenen Methoden auf. Nachrichten, die eine Objekt sich auf diesen Weg selbst sendet, werden genauso behandelt, wie solche von

Siehe auch [Methode](#) [UML] [Nachricht \[OO allg.\]](#) [Objekt \[OO allg.\]](#)

sequence diagram [engl.]

Synonym/Akronym/Übersetzung: [Sequenzdiagramm](#) [UML]

Sequentielles Vorgehen [OEP]

Vorgehensweise, bei der einzelne Aktivitäten nacheinander und nicht gleichzeitig bzw. nebeneinander durchgeführt werden.

Sequenzdiagramm [UML]

Synonym/Akronym/Übersetzung: [sequence diagram](#) [engl.]

Eine Sequenzdiagramm zeigt eine Menge von Interaktionen zwischen einer Menge ausgewählter Objekte in einer bestimmten begrenzten Situation (Kontext) unter Betonung der zeitlichen Abfolge. Ähnlich dem [Kollaborationsdiagramm](#).

Siehe auch [Sequenzdiagramm \[UML\]](#)

SEU [OO allg.]

Software-Entwicklungsumgebung

Signatur [OO allg.]

Die Signatur einer Operation setzt sich zusammen aus dem Namen der Operation, ihrer Parameterliste und der Angabe eines evtl. Rückgabetyps.

Siehe auch [Operation \[UML\]](#)

Software-Element [EmOO]

Ein Software-Element ist die kleinste unteilbar zu behandelnde und eindeutig zu identifizierende Einheit, die einem Änderungs- und Freigabeverfahren unterworfen werden kann.

Siehe auch [Konfigurationsmanagement \[OEP\]](#)

Software-Entwicklungsumgebung (SEU) [OO allg.]

Eine (meistens integrierte) Umgebung, die eine Reihe von Werkzeugen enthält, die im Rahmen der Softwareentwicklung notwendig oder hilfreich sind, beispielsweise Editor, Compiler, Debugger, Inspektoren etc.

Softwareentwicklungsplan [OEP]

Zusammenfassung aller zur Projekt- und Iterationsplanung notwendigen Unterlagen und Ergebnisse.

Späte Bindung [OO allg.]

Synonym/Akronym/Übersetzung: [Dynamische Bindung \[OO allg.\]](#)

Spezialisierung [OO allg.]

Eine Generalisierung (bzw. Spezialisierung) ist eine taxonomische Beziehung zwischen einem allgemeinen und einem speziellen Element (bzw. umgekehrt), wobei das speziellere weitere Eigenschaften hinzufügt, die Semantik erweitert und sich kompatibel zum allgemeinen verhält. Generalisierung und Spezialisierung sind Abstraktionsprinzipien zur hierarchischen Strukturierung der Modellsemantik unter einem diskriminierenden Aspekt.

Siehe auch [Vererbung \[OO allg.\]](#) [Diskriminator \[UML\]](#) [Generalisierung \[OO allg.\]](#)

Standard [OEP]

Eine [Richtlinie](#) oder Konvention, um bestimmte Mindesteigenschaften sicherzustellen.

Siehe auch [Richtlinie \[OEP\]](#)

Standard-Implementierung [OO allg.]

Konkrete Implementierung einer eigentlich abstrakten Operation, um für Subklassen ein Standardverhalten bereitzustellen.

Siehe auch [Vererbung \[OO allg.\]](#) [Operation \[UML\]](#)

state [engl.]

Synonym/Akronym/Übersetzung: [Zustand \[UML\]](#)

state diagram [engl.]

Synonym/Akronym/Übersetzung: [Zustandsdiagramm \[UML\]](#)

Statische Klassifikation [OO allg.]

Ein Objekt ist und bleibt Instanz genau einer Klasse, d. h. es kann seine Klassenzugehörigkeit während seiner Lebenszeit nicht ändern.

Siehe auch [Dynamische Klassifikation \[OO allg.\]](#)

Stereotyp [UML]

Synonym/Akronym/Übersetzung: [stereotype \[engl.\]](#)

Stereotypen dienen zur (werkzeug-, projekt-, unternehmens- oder methodenspezifischen) Erweiterung vorhandener Modellelemente der UML, d.h. ihres Metamodells. Entsprechend der mit der Erweiterung definierten Semantik wird das Modellierungselement, auf das es angewendet wird, direkt semantisch beeinflusst. In der Praxis geben Stereotypen häufig mögliche Verwendungszusammenhängen einer Klasse, einer Beziehung oder eines Paketes an. Andere erweiternde Mechanismen in der UML sind [Eigenschaftswerte](#) und [Zusicherungen](#). (Duden: *das* Stereotyp)

Siehe auch [Eigenschaftswert \[UML\]](#) [Zusicherung \[UML\]](#) [Metamodell \[OO allg.\]](#)

stereotype [engl.]

Synonym/Akronym/Übersetzung: [Stereotyp \[UML\]](#)

subclass [engl.]

Synonym/Akronym/Übersetzung: [Unterklasse \[OO allg.\]](#)

Subklasse [OO allg.]

Synonym/Akronym/Übersetzung: [Unterklasse \[OO allg.\]](#)

Subsystem [UML]

Ein Subsystem ist eine spezielle Form von Paket, daß gewöhnlich eine größere Zahl von Modellelementen enthält und der fachlichen Systempartitionierung dient.

Siehe auch [Paket \[UML\]](#)

super [OO allg.]

Super (Smalltalk, Java) ist ein Programmiersprachen-Schlüsselwort. Es bewirkt, daß die Nachricht immer an die nächsthöhere Klasse geht, die über die genannte Operation verfügt.

Siehe auch [Operation \[UML\]](#) [Oberklasse \[OO allg.\]](#) [Klasse \[OO allg.\]](#) [Nachricht \[OO allg.\]](#)

super state [engl.]

Synonym/Akronym/Übersetzung: [Oberzustand \[UML\]](#)

superclass [engl.]

Synonym/Akronym/Übersetzung: [Oberklasse \[OO allg.\]](#)

Superklasse [OO allg.]

Synonym/Akronym/Übersetzung: [Oberklasse \[OO allg.\]](#)

swimlane [engl.]

Synonym/Akronym/Übersetzung: [Verantwortlichkeitsbereich \[UML\]](#)

System Bauhaus [EmOO]

Das System Bauhaus ist ein Netzwerk international bekannter und im deutschsprachigen Raum tätiger unabhängiger Fachleute für objektorientierte Systementwicklung.

Systemtest [OEP]

Synonym/Akronym/Übersetzung: [Integrationstest \[OEP\]](#)

Szenario [UML]

Ein Szenario ist eine spezifische Folge von Aktionen. Beispielsweise ein konkreter Ablaufpfad in einem Anwendungsfall (sozusagen eine Instanz des Anwendungsfalls). Vgl. [Sequenzdiagramm](#).

Siehe auch [Anwendungsfall \[UML\]](#) [Sequenzdiagramm \[UML\]](#)

t

tagged value [engl.]

Synonym/Akronym/Übersetzung: [Eigenschaftswert \[UML\]](#)

template [engl.]

Synonym/Akronym/Übersetzung: [Parametrisierbare Klasse \[UML\]](#)

Ternäre Assoziation [UML]

Eine Assoziation, an der drei Assoziationsrollen beteiligt sind.

Siehe auch [Assoziationsrolle \[UML\]](#) [Assoziation \[UML\]](#)

Test [OEP]

Testen ist eine Tätigkeit mit der Absicht, Fehler zu finden. Ein Test kann die Existenz eines Fehlers beweisen. Die Abwesenheit von Fehlern ist nicht beweisbar.

Siehe auch [Fehler \[OO allg.\]](#)

this [OO allg.]

Synonym/Akronym/Übersetzung: [self \[OO allg.\]](#)

Transientes Objekt [OO allg.]

Ein Objekt, das nur während der Ausführungszeit eines Prozesses existiert.

Siehe auch [Persistentes Objekt \[OO allg.\]](#)

Transition [UML]

Eine Transition ist ein Zustandsübergang, häufig ausgelöst durch ein Ereignis.

Siehe auch [Zustand \[UML\]](#)

Typ [UML]

Definition einer Menge von Operationen und Attributen. Andere Elemente sind typkonform, wenn sie über die durch den Typen definierten Eigenschaften verfügen. Wird in der Praxis häufig gleichgesetzt mit der Beschreibung von [Schnittstellen](#).

Siehe auch [Attribut \[OO allg.\]](#) [Operation \[UML\]](#) [Schnittstelle \[OO allg.\]](#) [Klasse \[OO allg.\]](#)

type [engl.]

Synonym/Akronym/Übersetzung: [Typ \[UML\]](#)

U

UML [UML]

Synonym/Akronym/Übersetzung: [Unified Modeling Language \[UML\]](#)

Ungerichtete Assoziation [UML]

Eine Assoziation, deren Richtung nicht festgelegt, d. h. unterspezifiziert ist. In der UML wird die ungerichtete Assoziation häufig auch mit der bidirektionalen Assoziation gleichgesetzt (leider).

Siehe auch [Bidirektionale Assoziation \[UML\]](#) [Gerichtete Assoziation \[UML\]](#) [Assoziation \[UML\]](#)

Unidirektionale Assoziation [UML]

Synonym/Akronym/Übersetzung: [Gerichtete Assoziation \[UML\]](#)

Unified Modeling Language [UML]

Unified Modeling Language. Eine von der Object Management Group (OMG) standardisierte Notation und Semantik zur Visualisierung, Konstruktion und Dokumentation von Modellen für die objektorientierte Softwareentwicklung.

Unified Process [OEP]

Synonym/Akronym/Übersetzung: [Unified Software Development Process \(USDP\) \[OO allg.\]](#)

Unified Software Development Process (USDP) [OO allg.]

Vorgehensmodell der drei "Amigos". I. Jacobson, G. Booch, J. Rumbaugh: The Unified Software Development Process, Addison Wesley, Reading, MA, 1999.

Siehe auch [Rational Unified Process \(RUP\) \[OO allg.\]](#) [Unified Process \[OEP\]](#)

Unterklasse [OO allg.]

Eine Unterklasse ist die Spezialisierung einer Oberklasse und erbt alle Eigenschaften der Oberklasse.

Untermengenzusicherung [UML]

Eine Zusicherung bzw. Abhängigkeit zwischen zwei Assoziationen. Die Elemente (Objektverbindungen) der einen Assoziation müssen Teil der Elemente der anderen Assoziation sein.

Siehe auch [Abhängigkeitsbeziehung \[UML\]](#) [Zusicherung \[UML\]](#) [Assoziation \[UML\]](#)

UP [OEP]

UP ist die Abkürzung für [Unified Process](#).

Siehe auch [Unified Process \[OEP\]](#)

USDP [OO allg.]

USDP ist die Abkürzung für Unified Software Development Process

Siehe auch [Unified Software Development Process \(USDP\) \[OO allg.\]](#)

use case [engl.]

Synonym/Akronym/Übersetzung: [Anwendungsfall \[UML\]](#)

use case diagram [engl.]

Synonym/Akronym/Übersetzung: [Anwendungsfalldiagramm \[UML\]](#)

V

V-Modell [OO allg.]

Vorgehensmodell. Meistens wird damit das erstmalig 1992 in Form des "Softwareentwicklungsstandard der Bundeswehr" veröffentlichte Vorgehensmodell

Siehe auch [Vorgehensmodell \[OEP\]](#)

Validierung [OEP]

Überprüfung, ob das richtige Ergebnis bzw. Produkt hergestellt wurde.

Siehe auch [Verifizierung](#) [OO allg.]

Verantwortlichkeit [OO allg.]

Verantwortlichkeit (einer Klasse) umfaßt die Menge der Attribute und der interpretierbaren Nachrichten eines Objektes.

Siehe auch [Nachricht](#) [OO allg.] [Attribut](#) [OO allg.]

Verantwortlichkeitsbereich [UML]

Synonym/Akronym/Übersetzung: [swimlane](#) [engl.]

Durch Linien getrennte Bereiche in [Aktivitätsdiagrammen](#), die die Verantwortlichkeit der im Diagramm enthaltenen Elemente beschreiben.

Siehe auch [Aktivitätsdiagramm](#) [UML]

Vererbung [OO allg.]

Vererbung ist ein Programmiersprachenkonzept für die Umsetzung einer Relation zwischen einer Ober- und einer Unterklasse, wodurch Unterklassen die Eigenschaften ihrer Oberklassen mitbenutzen können. Vererbung implementiert normalerweise Generalisierungs- und Spezialisierungsbeziehungen. Alternativen sind [Delegation](#), [Aggregation](#), generisches Design, generische Programmierung.

Siehe auch [Generalisierung](#) [OO allg.] [Klasse](#) [OO allg.] [Mehrfachvererbung](#) [OO allg.]

Verfeinerungsbeziehung [UML]

Synonym/Akronym/Übersetzung: [refinement](#) [engl.]

Verfeinerungsbeziehungen sind Beziehungen zwischen gleichartigen Elementen unterschiedlichen Detaillierungs- bzw. Spezifikationsgrades. Verfeinerungsbeziehungen sind Stereotypisierungen von [Abhängigkeitsbeziehungen](#).

Siehe auch [Realisierungsbeziehung](#) [UML]

Verifizierung [OO allg.]

Überprüfung, ob ein Ergebnis bzw. Produkt richtig ist, d. h. gegebene Anforderungen erfüllt.

Siehe auch [Validierung](#) [OEP]

Verteilungsdiagramm [UML]

Synonym/Akronym/Übersetzung: [Einsatz- und Verteilungsdiagramm](#) [UML]

Virtuelle Klasse [OO allg.]

Synonym/Akronym/Übersetzung: [Abstrakte Klasse](#) [OO allg.]

Virtuelle Operation [OO allg.]

Synonym/Akronym/Übersetzung: [Abstrakte Operation](#) [OO allg.]

Vorbedingung [OO allg.]

Eine Vorbedingung beschreibt einen Zustand, der vor dem Ablauf einer Operation o.ä. gegeben sein muß.

Siehe auch [Zusicherung](#) [UML]

Vorgehensmodell [OEP]

Modellhafte und häufig formale Beschreibung, wie in Projekten vorgegangen werden soll.

W

Wartung [EmOO]

Wartung ist derjenige Teil des Software-Lebenszyklus, der nach dem Ende der Garantiezeit liegt. Hierunter werden auch Tätigkeiten verstanden, die eine bestehende Anwendung verbessern, optimieren, reparieren oder überprüfen mit dem Ziel, die Anwendung weiterhin bzw. besser nutzen zu können.

Wertemenge [OO allg.]

Synonym/Akronym/Übersetzung: [Enumeration](#) [OO allg.]

Widget [EmOO]

Ein Widget ist ein vorgefertigtes Oberflächenelement in den Programmierungsumgebungen für moderne grafische Benutzungsoberflächen. Der Programmierer benutzt es wie ein Bauteil, entweder deklarativ in einem Kompositionseditor oder durch prozeduralen Aufruf.

Siehe auch [Widgetpunkte](#) [EmOO]

Widgetpunkte [EmOO]

Widgetpunkte sind ein extrinsisches [Maß](#) für den Funktionsumfang und damit die die Größe eines Softwareprodukts. Siehe auch [Funktionspunkte](#).

Siehe auch [Widget](#) [EmOO] [Funktionspunkte](#) [EmOO] [Maß](#) [EmOO]

worker [OEP]

Synonym/Akronym/Übersetzung: [Rolle](#) [OEP]

Bezeichnung für Rollen im [Unified Process](#).

Siehe auch [Unified Process](#) [OEP]

Workflow [OO allg.]

Ein Workflow ist die computergestützte Automatisierung und Unterstützung eines Geschäftsprozesses oder eines Teils davon.

Siehe auch [Geschäftsprozess](#) [OO allg.]

Z

Zusicherung [UML]

Synonym/Akronym/Übersetzung: [constraint](#) [engl.]

Eine Zusicherung ist ein Ausdruck, der die möglichen Inhalte, Zustände oder die Semantik eines Modellelementes einschränkt und der stets erfüllt sein muß. Bei dem Ausdruck kann es sich um ein [Stereotyp](#) oder [Eigenschaftswerte](#) handeln, um eine freie Formulierung oder einen [OCL](#)-Ausdruck. Zusicherungen in Form reiner boolescher Ausdrücke werden auch Assertions genannt.

Siehe auch [assertion](#) [engl.] [Stereotyp](#) [UML] [Object Constraint Language \(OCL\)](#) [UML]

Zustand [UML]

Synonym/Akronym/Übersetzung: [state](#) [engl.]

Ein Zustand ist eine Abstraktion der möglichen Attributwerte eines Objektes. Ein Zustand gehört zu genau einer Klasse und stellt eine Abstraktion bzw. Zusammenfassung einer Menge von möglichen Attributwerten dar, die die Objekte dieser Klasse einnehmen können. In der UML ist ein Zustand eine Bedingung bzw. Situation im Leben eines Objektes, während der eine bestimmte Bedingung erfüllt ist, Aktivitäten ausgeführt werden oder auf ein Ereignis gewartet wird.

Zustandsdiagramm [UML]

Synonym/Akronym/Übersetzung: [state diagram](#) [engl.]

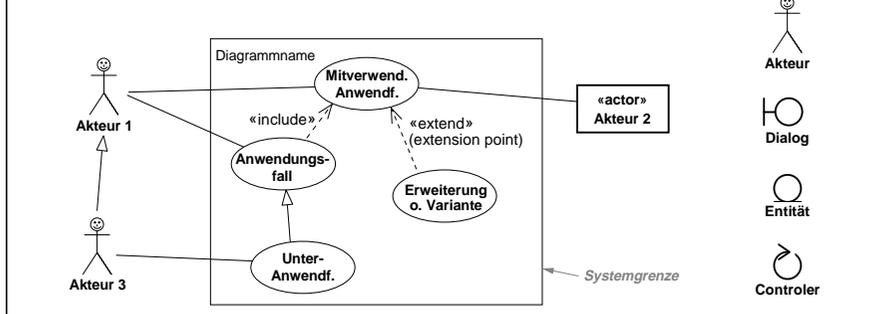
Ein Zustandsdiagramm zeigt eine Folge von Zuständen, die ein Objekt im Laufe seines Lebens einnehmen kann und aufgrund welcher Stimuli Zustandsänderungen stattfinden. Ein Zustandsdiagramm beschreibt eine hypothetische Maschine (Endlicher Automat), die sich zu jedem Zeitpunkt in einer Menge endlicher Zustände

befindet. Sie besteht aus:

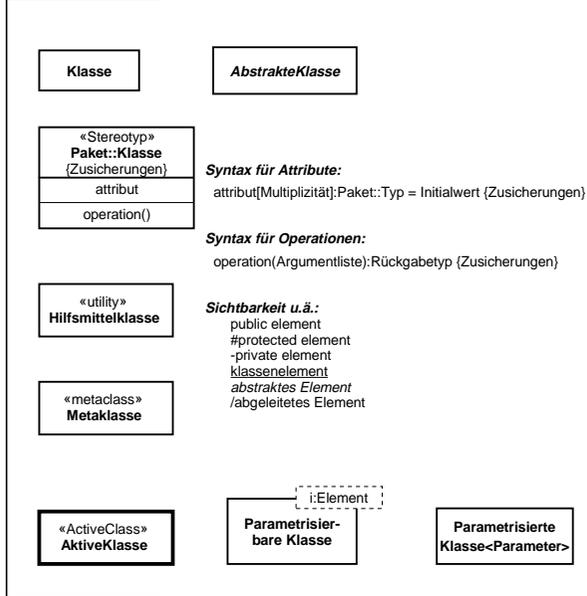
- einer endlichen, nicht-leeren Menge von Zuständen
- einer endlichen, nicht-leeren Menge von Eingabesymbolen (Ereignissen)
- Funktionen, die den Übergang von einem Zustand in den nächsten beschreiben
- einen Anfangszustand
- einer Menge von Endzuständen

Ende des Glossars

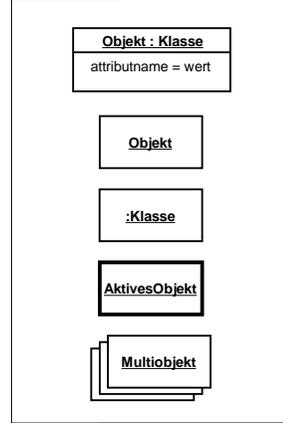
Anwendungsfalldiagramm



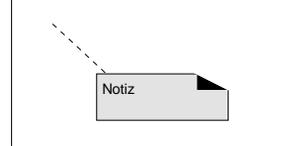
Klassen



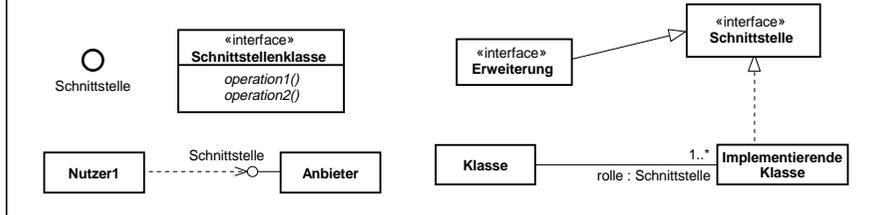
Objekte



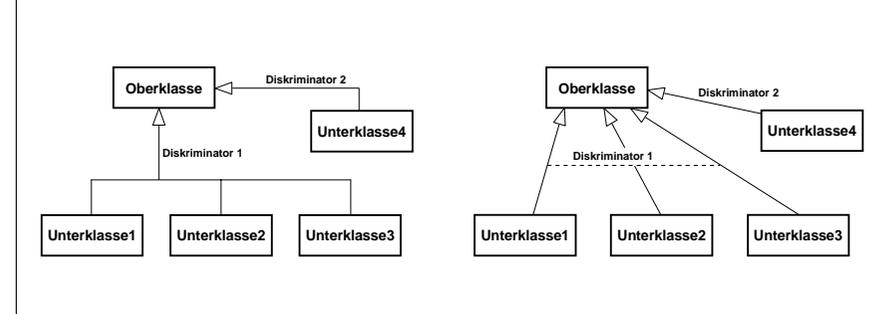
Notiz



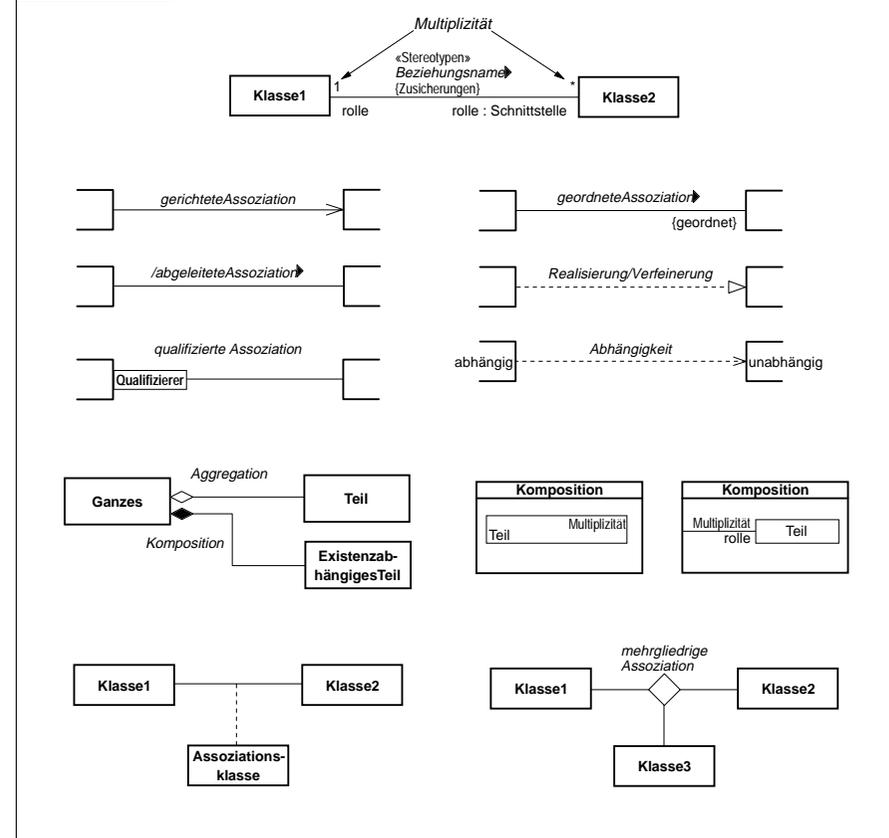
Schnittstellen



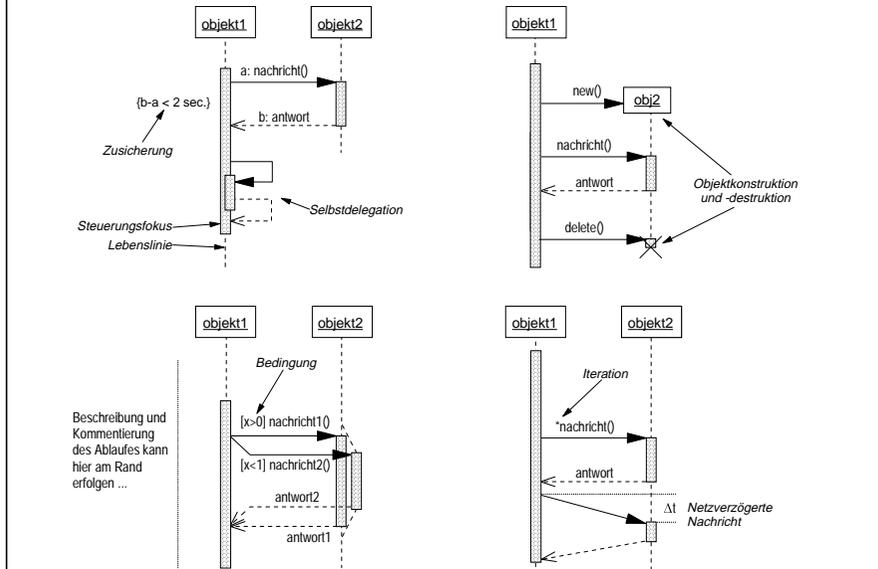
Vererbung



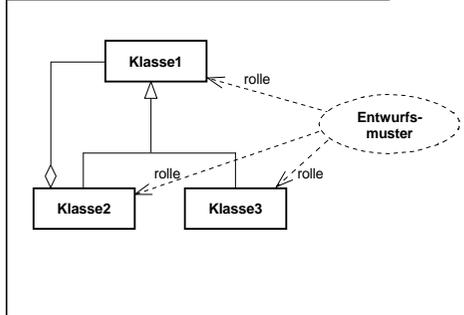
Assoziationen



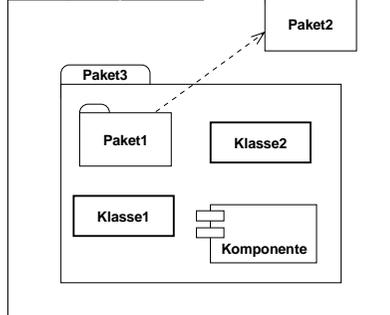
Sequenzdiagramme



Zusammenarbeits-/ Entwurfsmuster-Notation



Pakete, Subsysteme



Zusicherung

{Freiformulierter Text}
{OCL-Ausdruck}
{vertrag.summe>500}

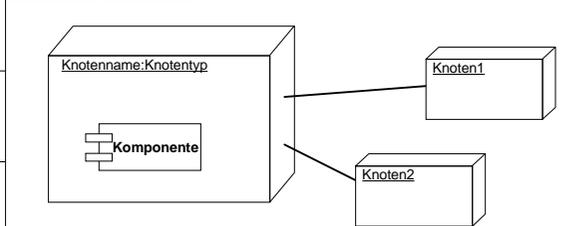
Eigenschaftswert

{schlüssel = wert}
{abstrakt = true}
{abstrakt}

Stereotyp

«stereotyp»
«interface»

Einsatzdiagramm

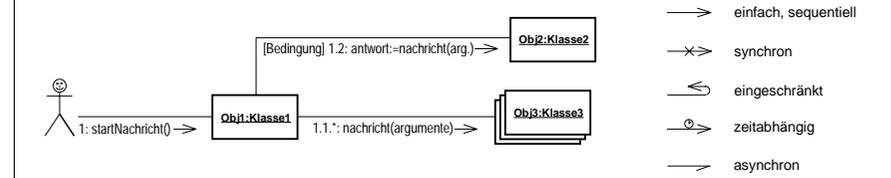


Aktuelle Fassung,
Info und Download: <http://www.oose.de/uml>

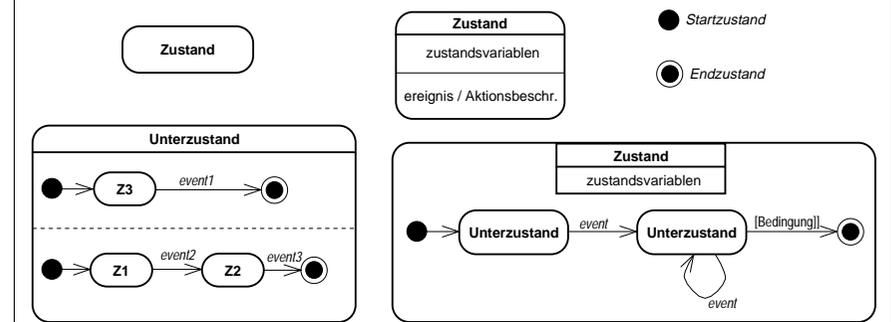
Unified Modeling Language (UML 1.3)

Notationsübersicht
Teil 3/4

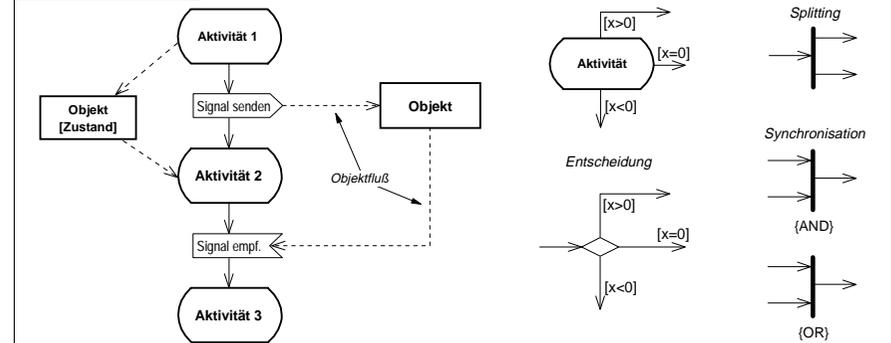
Kollaborationsdiagramme



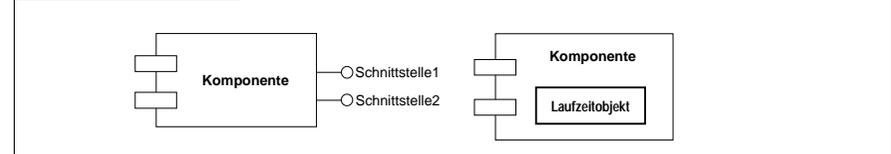
Zustandsdiagramme



Aktivitätsdiagramme



Komponentendiagramme



Aktuelle Fassung,
Info und Download: <http://www.oose.de/uml>

Unified Modeling Language (UML 1.3)

Notationsübersicht
Teil 4/4