

UML 系统建模

The Unified Modeling Language(UML) is a general-purpose visual modeling language that is used to specify, visualize, construct, and document the artifacts of a software system. It captures decisions and understanding about systems that must be constructed. It is used to understand, design, browse, configure, maintain, and control information about such systems. It is intended for use with all development methods, lifecycle stages, application domains, and media. The modeling language is intended to unify past experience about modeling techniques and to incorporate current software best practices into a standard approach. UML includes semantic concepts, notation, and guidelines. It has static, dynamic, environmental, and organizational parts. It is intended to be supported by interactive visual modeling tools that have code generators and report writers. The UML specification does not define a standard process but is intended to be useful within an iterative development process. It is intended to support most existing object-oriented development processes.

The UML captures information about the static structure and dynamic behavior of a system. A system is modeled as a collection of discrete objects that interact to perform work that ultimately benefits an outside user. The static structure defines the kinds of objects important to a system and to its implementation, as well as the relationships among the objects. Dynamic behavior defines the history of objects over time and the communications among objects to accomplish goals. Modeling a system from several separate but related viewpoints permits it to be understood for different purposes.

The UML also contains organizational constructs for arranging model into packages that permit software teams to partition large systems into workable pieces, to understand and control dependencies among the packages, and to manage the versioning of model units in a complex development environment. It contains constructs for representing implementation decisions and for organizing run-time elements into components.

UML is not a programming language. Tools can provide code generators from UML into a variety of programming languages, as well as construct reverse-engineered models from existing programs. The UML is not a highly formal language intended for theorem proving. There are a number of such languages, but they are not easy to understand or to use for most purposes. The UML is a general-purpose modeling language. For specialized domains, such as GUI layout, VLSI circuit design, or rule-based artificial intelligence, a more specialized tool with a special language might be appropriate. UML is a discrete modeling language. It is not intended to model continuous systems such as those found in engineering and physics. UML is intended to be a universal general-purpose modeling language for discrete systems such as those made of software, firmware, or digital logic.

UML（统一建模语言）是一种用于软件工程中的可视化建模语言，它可以帮助开发人员在设计阶段对软件系统进行建模和分析。UML 提供了多种图形化的建模方法，包括用例图、类图、序列图、状态图等，这些图形化方法可以帮助开发人员清晰地描述软件系统的结构和行为。

在学习 UML 知识时，我们需要了解 UML 的基本概念和各种图形化建模方法。例如，用例图可以帮助我们描述系统的功能需求，类图可以帮助我们描述系统的结构，序列图可以帮助我们描述系统中对象之间的交互。通过学习这些知识，我们可以更好地理解如何使用 UML 对软件系统进行建模。

此外，UML 还提供了一些高级建模方法，例如组合结构图、组件图、部署图等。这些高级建模方法可以帮助我们更好地描述软件系统的复杂性。例如，组合结构图可以帮助我们描述系统中各个部分之间的关系，组件图可以帮助我们描述系统中各个组件之间的依赖关系，部署图可以帮助我们描述系统在运行时的部署情况。

在实际应用中，我们可以使用 UML 来设计软件系统。例如，在设计阶段，我们可以使用用例图来描述系统的功能需求，然后使用类图来设计系统的结构。在开发阶段，我们可以使用序列图来描述对象之间的交互，以便更好地实现系统的功能。此外，我们还可以使用高级建模方法来描述软件系统的复杂性。总之，UML 是一种非常有用的工具，它可以帮助我们更好地设计和开发软件系统。