Modelagem UML

Prof. Dr. Rodrigo Piva

Modelagem UML

- Introdução.
- Motivação.
- Histórico.
- Exemplos.

Modelagem UML

- UML, ou Unified Modeling Language, é uma linguagem visual usada para modelar sistemas de software.
- É uma linguagem de modelagem padrão amplamente utilizada na engenharia de software para visualizar, projetar e documentar sistemas de software.

Modelagem UML

A UML consiste em um conjunto de notações gráficas, chamadas de diagramas UML, que representam diferentes aspectos de um sistema de software. Esses diagramas incluem:

Modelagem UML

- Diagramas de classes: usados para representar as classes e seus relacionamentos em um sistema.
- Diagramas de objetos: usados para representar instâncias de classes e seus relacionamentos.
- Diagramas de casos de uso: utilizados para representar as interações entre um sistema e seus usuários.
- Diagramas de atividades: usados para representar o fluxo de atividades em um sistema.
- Diagramas de seqüência: usados para representar as interações entre objetos em um sistema.
- Diagramas de estado: usados para representar os diferentes estados de um objeto em um sistema.

- A UML é amplamente utilizada na engenharia de software e é suportada por muitas ferramentas de desenvolvimento de software.
- É uma ferramenta valiosa para comunicação e colaboração entre desenvolvedores de software, bem como para documentação de sistemas de software.

- A motivação por trás da criação da modelagem UML foi fornecer uma maneira padronizada de representar e documentar sistemas de software, bem como melhorar a comunicação entre desenvolvedores de software, designers e partes interessadas.
- Antes da UML, havia muitas notações e metodologias diferentes usadas para modelagem de software, o que dificultava o compartilhamento e a compreensão do design de software entre diferentes interessados. A UML foi criada para resolver esse problema, fornecendo uma notação única e padronizada que pode ser usada para modelar e documentar sistemas de software.

- Outra motivação para a criação da UML foi melhorar a qualidade do desenvolvimento de software, fornecendo uma estrutura para projeto de software baseada em princípios de programação orientada a objetos.
- Usando UML para modelar sistemas de software, os desenvolvedores podem garantir que o sistema foi projetado de forma modular e escalável, com clara separação de interesses entre diferentes componentes.

- Além disso, a UML foi projetada para ser independente de linguagem, o que significa que ela pode ser usada para modelar sistemas de software independentemente da linguagem de programação ou plataforma que está sendo usada.
- Isso facilitou o compartilhamento e a compreensão do design de software entre diferentes equipes de desenvolvimento e organizações.
- No geral, a motivação por trás da criação da UML foi fornecer uma linguagem de modelagem padronizada, abrangente e flexível que pudesse melhorar a comunicação, o projeto e o desenvolvimento de sistemas de software.

- A história da UML remonta ao início da década de 1990, quando três proeminentes engenheiros de software, Grady Booch, James Rumbaugh e Ivar Jacobson, estavam trabalhando em suas respectivas análises orientadas a objetos e metodologias de projeto. Booch estava trabalhando no método Booch, Rumbaugh estava trabalhando em Object Modeling Technique (OMT) e Jacobson estava trabalhando em Object-Oriented Software Engineering (OOSE).

- Em 1994, os três engenheiros uniram forças para criar uma nova metodologia unificada que combinasse os melhores aspectos de suas abordagens individuais. Essa metodologia foi chamada de Método Unificado e serviu como precursora da UML.
- Em 1995, os três engenheiros enviaram uma proposta ao Object Management Group (OMG), uma organização de padrões que promove padrões para programação orientada a objetos. A proposta sugeria que o OMG adotasse o Método Unificado como padrão para análise e projeto orientado a objetos.

- A UML foi desenvolvida em meados da década de 1990 por um grupo de engenheiros de software, liderados por Grady Booch, James Rumbaugh e Ivar Jacobson.
- O objetivo da UML era criar uma linguagem padrão que pudesse ser usada para descrever, projetar e documentar sistemas de software, independentemente da linguagem de programação ou plataforma usada.

- O OMG aceitou a proposta e formou o consórcio UML Partners, que incluía IBM, Microsoft, Oracle e outras empresas de software. O consórcio trabalhou em conjunto para desenvolver a especificação UML, lançada pela primeira vez em 1997.
- Desde então, a UML passou por diversas revisões e atualizações, sendo a última versão a UML 2.5. A UML tornou-se amplamente aceita e usada na comunidade de engenharia de software e é considerada uma linguagem de modelagem padrão para sistemas de software.

Os conceitos básicos da UML incluem:

- Objetos: Objetos são instâncias de classes em um sistema de software. Na UML, os objetos são representados por formas retangulares com o nome do objeto dentro.
- Classes: As classes são os blocos de construção de um sistema de software e representam a estrutura e o comportamento do sistema. Na UML, as classes são representadas por formas retangulares com o nome da classe na parte superior e os atributos e operações listados abaixo.

3. Relacionamentos: Os relacionamentos descrevem como objetos e classes estão relacionados entre si em um sistema de software. Existem vários tipos de relacionamentos em UML, incluindo associação, agregação e herança.

4. Diagramas UML: Diagramas UML são representações gráficas de diferentes aspectos de um sistema de software. Existem vários tipos de diagramas UML, incluindo diagramas de classes, diagramas de casos de uso, diagramas de atividades, diagramas de sequência e diagramas de estado.

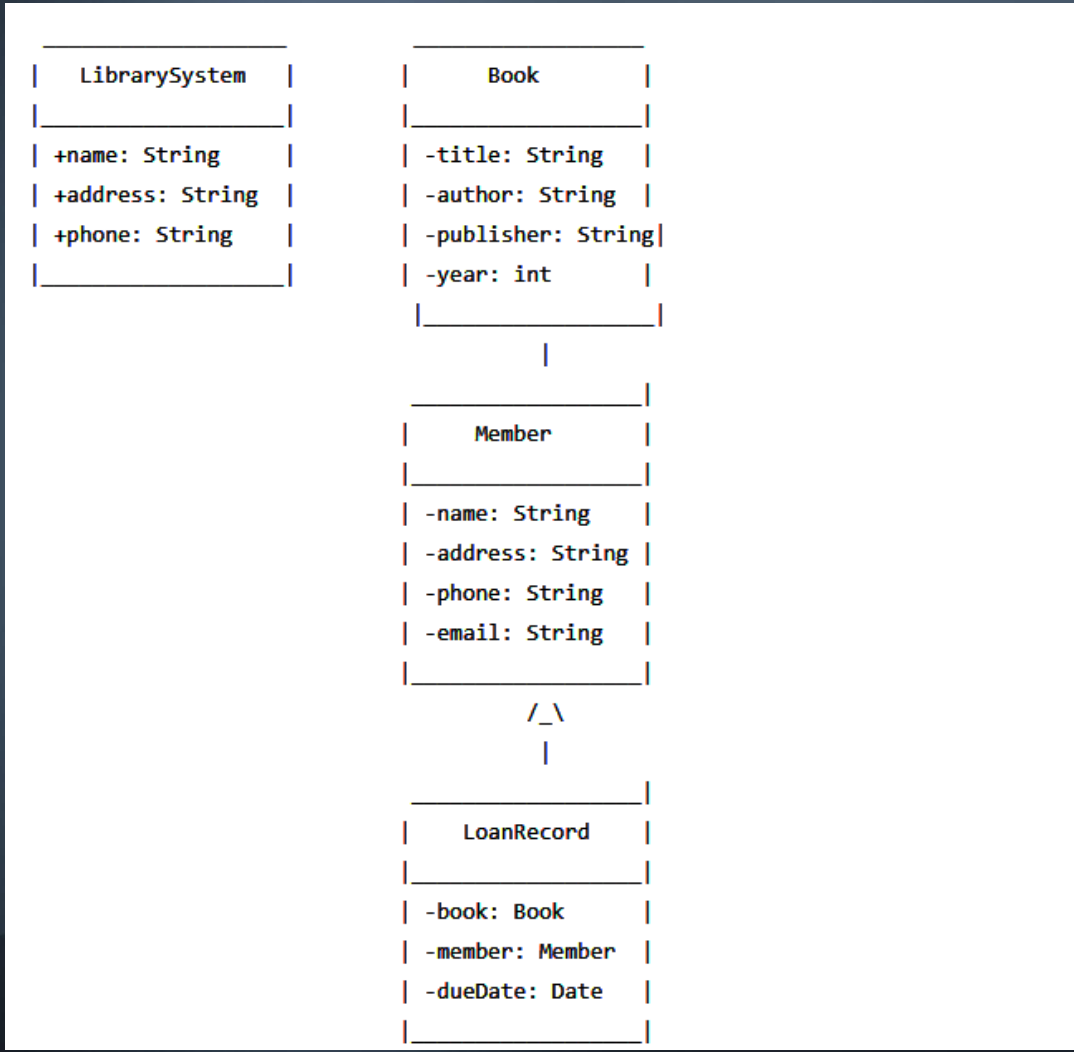
- Comportamento: O comportamento descreve como um sistema de software responde a eventos e interage com seu ambiente. Na UML, o comportamento é representado por meio de diagramas de atividades, diagramas de estado e diagramas de sequência.
- Encapsulamento: O encapsulamento é o princípio de ocultar os detalhes de implementação de uma classe do restante do sistema. Em UML, o encapsulamento é representado usando modificadores de acesso em atributos e operações de classe.

- Abstração: Abstração é o princípio de focar nas características essenciais de um sistema de software e ignorar os detalhes que não são relevantes para o problema em questão. Na UML, a abstração é representada por meio de interfaces, classes abstratas e relacionamentos de generalização.
- Esses conceitos básicos formam a base da UML e são usados para modelar sistemas de software de maneira padronizada e compreensível.

Existem conceitos mais básicos em UML. Aqui estão alguns adicionais:

- Estereótipos: Os estereótipos são usados para estender elementos de modelagem UML com propriedades e características adicionais. Eles são representados usando a notação «estereótipo», por exemplo, «ator» ou «banco de dados».
- Pacotes: Os pacotes são usados para organizar elementos de modelagem UML em agrupamentos lógicos. Eles são representados por formas retangulares com o nome do pacote na parte superior

Exemplo:


LibrarySystem
+name: String
+address: String
+phone: String
Book
-title: String
-author: String
-publisher: String
-year: int
Member
-name: String
-address: String
-phone: String
-email: String
LoanRecord
-book: Book
-member: Member
-dueDate: Date

Exemplo:

Neste exemplo, o diagrama de classes mostra três classes: LibrarySystem, Book e Member. A classe LibrarySystem possui atributos para o nome, endereço e número de telefone da biblioteca. A classe Book tem atributos para título, autor, editora e ano de publicação. A classe Member tem atributos para o nome, endereço, número de telefone e endereço de e-mail dos membros da biblioteca.

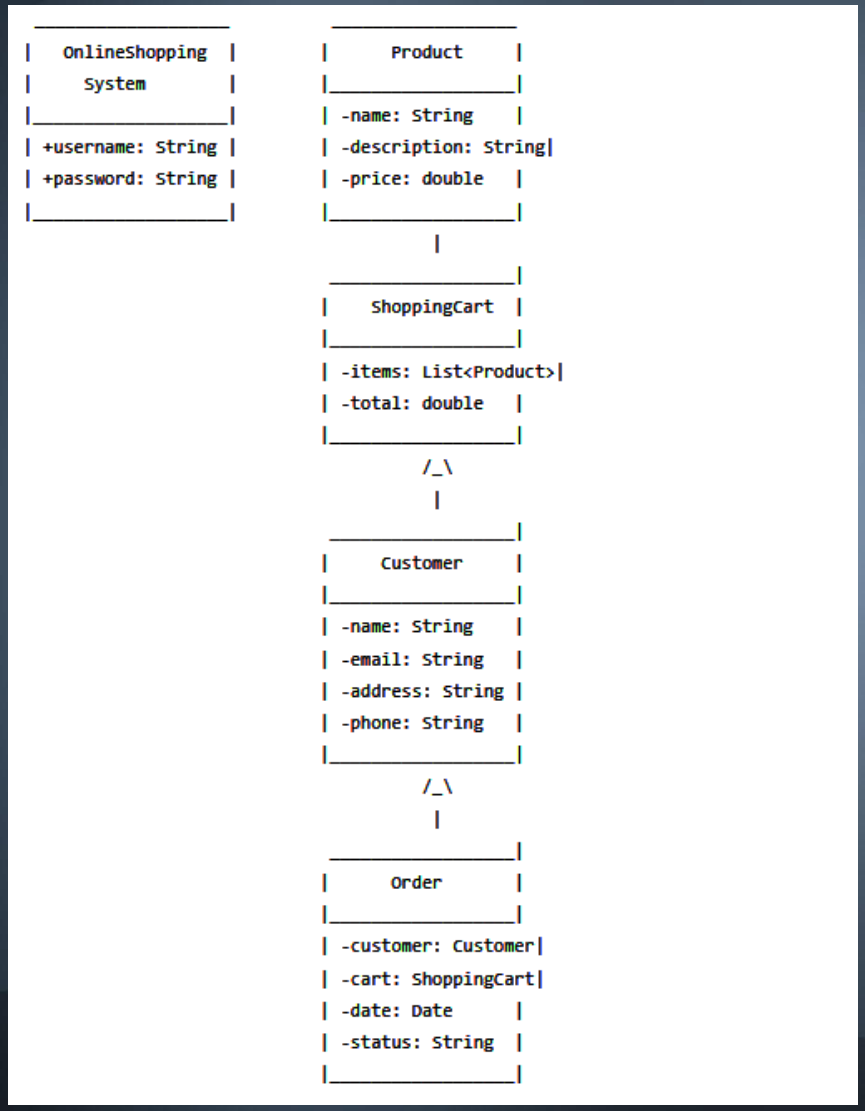
Há também uma classe LoanRecord que representa um registro de empréstimo de um livro, que possui atributos para o livro que está sendo emprestado, o membro que está pegando o livro emprestado e a data de vencimento do livro.

Exemplo:

O diagrama também mostra os relacionamentos entre as classes. As classes Book e Member possuem um relacionamento de associação com a classe LoanRecord, indicando que um registro de empréstimo envolve um book e um membro. A classe LoanRecord possui um relacionamento de composição com as classes Book e Member, indicando que um registro de empréstimo é composto por um book e um membro.

Este exemplo demonstra como a UML pode ser usada para representar a estrutura e os relacionamentos de classes em um sistema de software.

Exemplo:


OnlineShopping System
+username: String
+password: String
Product
-name: String
-description: String
-price: double
ShoppingCart
-items: List<Product>
-total: double
Customer
-name: String
-email: String
-address: String
-phone: String
Order
-customer: Customer
-cart: ShoppingCart
-date: Date
-status: String

Exemplo:

Neste exemplo, o diagrama de classes mostra quatro classes: OnlineShoppingSystem, Product, ShoppingCart e Customer. A classe OnlineShoppingSystem possui atributos para o nome de usuário e senha do usuário.

A classe Product tem atributos para o nome, descrição e preço do produto. A classe ShoppingCart possui atributos para a lista de itens adicionados ao carrinho e o preço total do carrinho.

A classe Cliente tem atributos para o nome, e-mail, endereço e número de telefone do cliente. A classe Order tem atributos para o cliente que fez o pedido, o carrinho de compras contendo os itens solicitados, a data do pedido e o status do pedido.

Exemplo:

O diagrama também mostra os relacionamentos entre as classes. A classe ShoppingCart possui um relacionamento de associação com a classe Product, indicando que o carrinho pode conter um ou mais produtos. A classe Order possui um relacionamento de composição com as classes ShoppingCart e Customer, indicando que um pedido é composto por um carrinho de compras e um cliente.

Este exemplo demonstra como a UML pode ser usada para representar a estrutura e os relacionamentos de classes em um sistema de compras online.

Exercicio:

A tabela abaixo é uma tabela em Excel do usuário Dollynaldo, que controla os gastos mensais com sua conta de luz.

De cada conta de luz, Dollynaldo cadastra a data em que foi feita a leitura do relógio de luz, seu número da leitura, a quantidade de Kw mensal gasto, o valor a pagar, a data do pagamento e sua média de consumo.

Mensalmente, são realizadas as seguintes pesquisas:

- verificação do mês de menor consumo;
- verificação do mês de maior consumo.

LISTA DE ACOMPANHAMENTO DE GASTO DE LUZ						
Data Leitura	n° Leitura	kw gasto	valor a pagar	data pagto	media consumo	
04/09/2022	4166	460	206,43	15/09/2022	15,33	
05/11/2022	4201	350	157,07	15/11/2022	12,06	
Menor Consumo			460	206,43		
Maior Consumo			350	157,07		

Identifique as classes, atributos e métodos desse cenário

Exercicio:

O desenvolvedor Valdisney decidiu criar uma classe que permita mover um boneco na tela, Esse boneco deve ter nome, posição da coordenada X, posição da coordenada Y e direção atual (cima, baixo, direita, esquerda).

Identifique as classes, atributos e métodos desse cenário

Dúvidas???