

Disciplina: Processo de desenvolvimento de software

Aula 3: Atividade de análise no processo de desenvolvimento de softwares



Apresentação

Nesta aula, iremos definir o conceito de análise para o processo de desenvolvimento de software.

A fase de análise tem como objetivo fazer uma modelagem dos agentes, separando-os em objetos, classes e atributos.

A análise pode ser estrutural ou comportamental.

Objetivos

- Conhecer as atividades de análise de desenvolvimento de software;
- Entender os relacionamentos dos objetos;
- Modelar os relacionamentos dos objetos.

Conceito de Modelagem

Modelagem

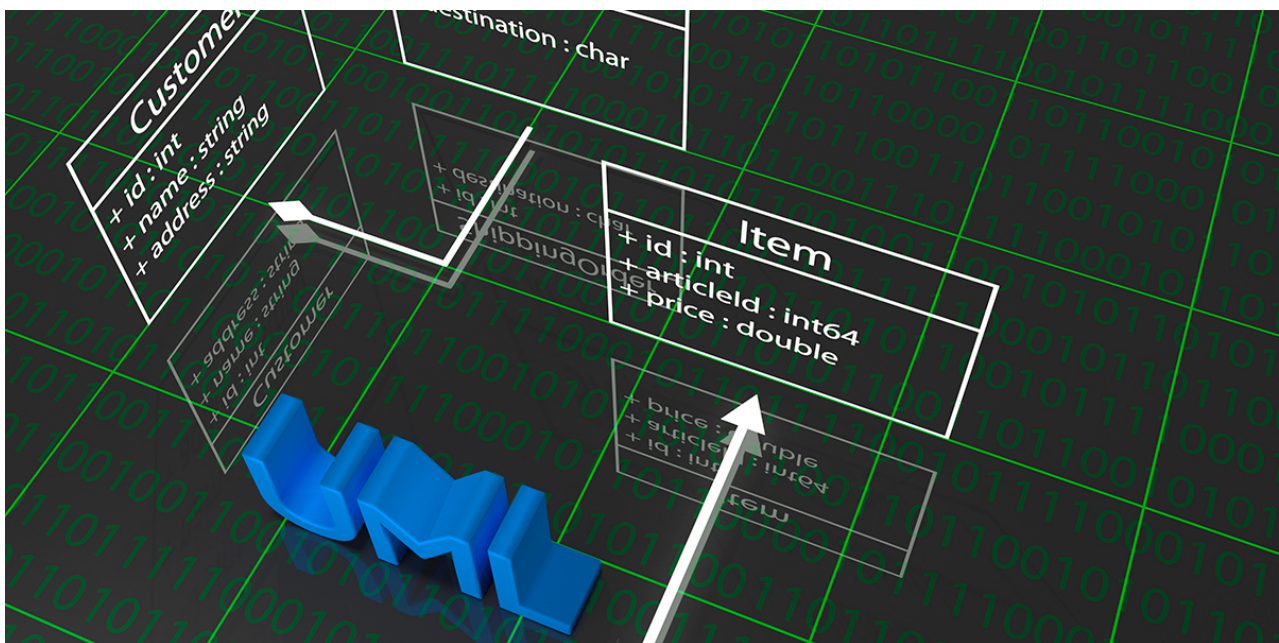
Serve para verificar a qualidade dos requisitos, estudados na aula anterior, que se tornarão precisos e detalhados o suficiente para as atividades do próximo passo no processo de desenvolvimento de software.

Análise

Atividade que utiliza o conceito de orientação a objeto, utilizando a UML como notação. Tem como objetivo modelar o problema, não a solução.

UML

Unified Modeling Language, linguagem de modelagem unificada, utilizada em engenharia de software para visualizar o desenho do sistema e a intercomunicação entre objetos.



UML (Fonte: Ton Snoei/Shutterstock).

Objeto e classe

Objeto

Estrutura de dados encapsulada por procedimentos. Essa estrutura são os atributos e operações.

Classe

Conjunto de objetos similares agrupados em que a etapa de análise está mais voltada para sua realização.

Vejamos um exemplo para entender melhor.

Classe: Pessoa.

Objetos: Pessoas.



 Mosaico de pessoas (Fonte: Etraveler/Shutterstock).

Tipos de análise

Análise Estrutural



Tem como objetivo modelar aspectos estáticos de um problema, utilizando o modelo orientado a objeto. É utilizada em conjunto com detalhamento de requisitos para visualizar e fornecer base para identificar soluções para os

requisitos apresentados.

Vejamos as atividades dentro da análise estruturada:

Identificação de classes

Identificar quais são as classes chaves. Fazer o levantamento com base em suas responsabilidades e colaborações. Utiliza-se em larga escala o cartão CRC (*Class-Responsability-Collaborator*).

Organização das classes

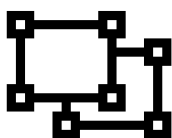
Organizar as classes em três tipos:

Entidade
Representa conceitos do domínio do problema herdada dos modelos de negócio.
Fronteira
Representa interfaces externas que estão dentro do produto, como interface de usuário e conexão com outros sistemas. Facilita o desenho das interfaces.
Controle
Organização que não pertence à entidade e nem à fronteira. Normalmente é associada a um caso de uso.

Identificação dos relacionamentos

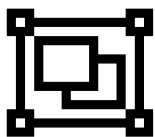
Ajuda a filtrar e refinar as classes.

Pode se por:



1. Associação

Indica a relação entre duas classes em que o objeto de uma classe consegue obter informações da outra a que foi associado.



2. Agregação

Indica uma associação, mas com a classe se apropriando das informações de um objeto da outra.

Identificação dos atributos

A cada classe é atribuída uma característica responsável por tomar alguma ação.

Análise comportamental

Aplicada depois que os requisitos forem detalhados, validando-os e indicando as dificuldades de implementação no plano de conceito.

Diagrama de interação

Mensagens que são trocadas, ao longo do tempo, para execução de alguma tarefa.

Mensagens e Operações: representam um mecanismo de interação, ou seja, um objeto só poderá receber uma mensagem invocada por uma classe.

A mensagem tem as seguintes partes:

Receptor	Operação	Parâmetro
----------	----------	-----------

Interação: como as mensagens tráfegarão para a execução de uma tarefa.

Diagrama de sequência: ordem temporal das ações que serão executadas.

Identificação das operações

Todas as mensagens devem ser mapeadas para executarem alguma operação. Podem ser: Incluir, Alterar, Excluir, dentre outras.

Podem ser: Incluir, Alterar, Excluir, dentre outras.

Notas

Referências

GUSTAFSON, Davis A. **Engenharia de software**. 8. ed. São Paulo: Pearson Education, 2007. cap. 8 e 13.

PAULA FILHO, Wilson de. **Engenharia de software**: fundamentos, métodos e padrões. 3. ed. São Paulo: LTC, 2009. cap. 1, 5 e 21.

SOMMERVILLE, Ian. **Engenharia de software**. 1. ed. Porto Alegre: Artmed, 2003. cap. 10.

Próximos Passos

- Etapa de desenho onde se trabalha e modela os requisitos para se obter uma estrutura para auxiliar no desenho da solução.

Explore Mais

Pesquise na internet sites, vídeos e artigos relacionados ao conteúdo visto.

Em caso de dúvidas, converse com seu professor online por meio dos recursos disponíveis no ambiente de aprendizagem.