



Programação orientada a objeto

Aula 4



Em programação, um diagrama de classes é uma representação da estrutura e relações das classes que servem de modelo para objetos. Podemos afirmar de maneira mais simples que seria um conjunto de objetos com as mesmas características, assim saberemos identificar objetos e agrupá-los, de forma a encontrar suas respectivas classes. Na Unified Modeling Language (UML) em diagrama de classe, uma classe é representada por um retângulo com três divisões, são elas: O nome da classe, seus atributos e por fim os métodos. Vejam na Figura 1 sua representação.

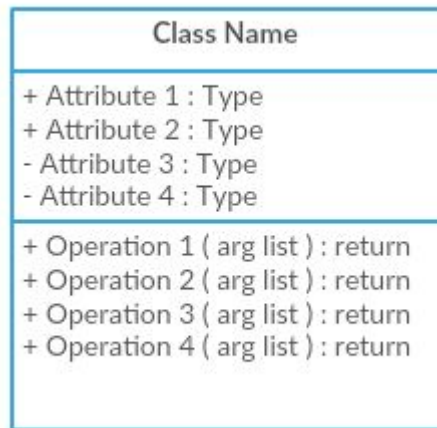




Representando uma classe

Para representarmos uma classe, adicionaremos um retângulo com três compartimentos, sendo eles:

- O Identificador da classe;
- Seus atributos;
- Seus métodos;

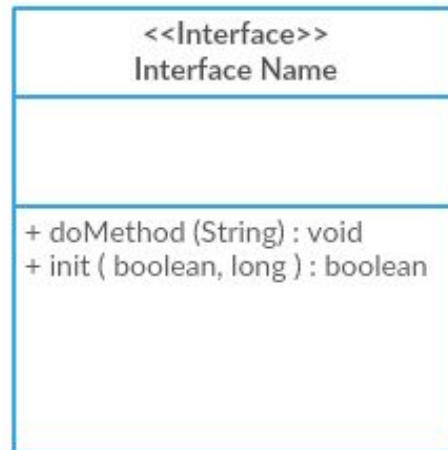




Representando uma interface

Para representarmos uma interface, adicionaremos um retângulo com três compartimentos, sendo eles:

- O Identificador da classe;
- Seus atributos (em branco);
- Seus métodos;





Representando o encapsulamento em nossas classes e interfaces

Para representar o encapsulamento dos atributos e métodos, utilize a seguinte notação:







- + público - visível em qualquer classe
- # protegido - qualquer descendente pode usar
- - privado - visível somente dentro da classe











Representando relacionamentos entre as classes

Para representar relacionamento entre as classes, utilizamos as notações a seguir:

Class Diagram Relationship Type	Notation
Association	
Inheritance	
Realization/ Implementation	
Dependency	
Aggregation	
Composition	



- Associação 
- Herança 
- Implementação 
- Dependência 
- Agregação 
- Composição 



- Os diagramas de classes tendem a ficar incoerentes à medida que se expandem e crescem. É melhor evitar criar diagramas grandes e dividi-los em diagramas menores, que podem ser ligados entre si mais tarde. Você pode muito facilmente fazer isso com Creately. Ajuda-o a melhorar a legibilidade dos seus diagramas;
- Usando a simples notação de classe, você pode criar rapidamente uma síntese de alto nível do seu sistema. Um diagrama detalhado pode ser criado separadamente, conforme necessário, e até mesmo ligado ao primeiro para fácil referência;
- Quanto mais linhas se sobrepõem nos seus diagramas de classes, mais desordenada se torna. O leitor só vai ficar confuso ao tentar encontrar o caminho. Certifique-se que não há duas linhas que se cruzam;
- Use cores para agrupar módulos comuns. Cores diferentes em classes diferentes ajudam o leitor a diferenciar entre os vários grupos;



Vamos criar uma rede social. Para isso, precisaremos controlar um sistema de postagens. Desenvolva um diagrama que ilustrará a seguinte estrutura:

- Um usuário pode efetuar uma postagem;
- Esse post pode ser somente um texto, uma imagem ou um vídeo;
- O post de vídeo e imagem pode possuir também um texto descritivo;





Crie um diagrama que especifique um sistema para agências de viagens. O diagrama deve detalhar os seguintes requisitos:

- Um cliente poderá comprar uma viagem;
- A viagem possuirá um destino, vinculado a um pacote;
- Essa viagem possuirá uma cobrança;
- A viagem possuirá data de partida, data de chegada e destino;



Crie um diagrama que exemplifique a estrutura a ser desenvolvida para uma agenda de contatos. Os requisitos dessa agenda são os seguintes:

- Cada contato terá um nome diferente;
- Cada contato na agenda pode conter vários telefones e vários e-mails;
- Para cada telefone e/ou cada e-mail, deverá existir um método no objeto para contatar a pessoa através dessa informação;

