

Aula 2

Modelagem de software

Professor: Jonny I. Beckert

Ementa da disciplina

- Visão geral do conceito de processo de software.
- Fundamentos de requisitos e visão geral de técnicas para elicitação de requisitos.
 - Tipos de requisitos:
 - funcionais e não funcionais.
 - Prototipação não funcional.
- **Análise e projeto orientado a objetos.**
- Notação UML (*Unified Modeling Language*).
- Modelo de casos de uso (diagrama e especificação).
 - *User stories*.
 - Diagrama de classes.
 - Diagrama de sequência.
 - Diagrama de estados.
 - Diagrama de atividades.
 - Diagrama de componentes.
 - Diagrama de implantação.
- Análise e projeto de banco de dados.
 - Modelo entidade relacionamento.
 - Modelo relacional e normalização.
 - Modelo lógico e físico de banco de dados.
 - Operações matemáticas básicas com vetores e matrizes.
- Teoria de conjuntos.
 - Álgebra.
 - Geometria.
- Raciocínio lógico booleano.

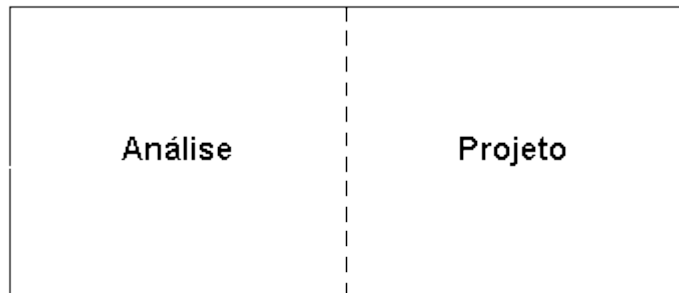
Análise e projeto orientado a objetos

- Saber uma linguagem de programação orientada a objeto (OO) não é suficiente para criar sistemas OO
 - Tem que saber Análise e Projeto OO (APOO)
 - Isto é, Análise e Projeto usando uma perspectiva de objetos
- Nesta disciplina, vamos nos concentrar na fase de Projeto, supondo que a Análise já foi feita
 - Na prática, a análise e o projeto são feitos em ciclos
 - Em cada ciclo, a análise vem *antes* do projeto
 - Ensinaamos Projeto primeiro porque Análise requer mais maturidade e será feita em semestre posterior
- Usaremos a linguagem UML (*Unified Modeling Language*) para criar modelos (de análise e de projeto)
 - Um *modelo* é uma representação abstrata dos aspectos essenciais de um sistema
 - O que é "essencial" depende do momento da modelagem
 - A UML usa uma representação principalmente gráfica para representar os modelos
 - UML é *muito* popular hoje em dia para modelar sistemas
 - Não vamos ensinar UML nessa disciplina

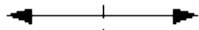
ANALISE & PROJETO

- Analise:
 - *o que* deve ser feito
- Projeto:
 - *como* pode ser feito

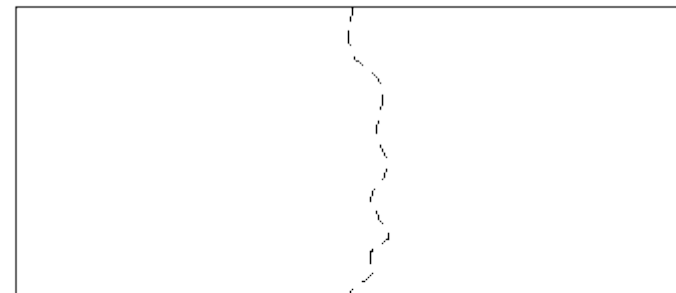
ANÁLISE & PROJETO



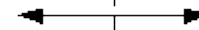
Modelagem
do problema
(entender)



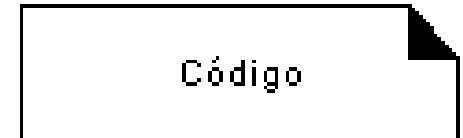
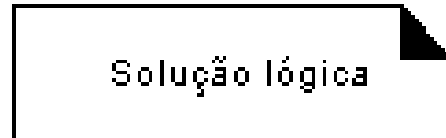
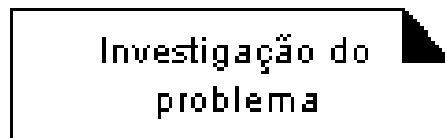
Modelagem
da solução
(criar)



Análise =
Informação
importante
para o
cliente
discutir e
aprovar



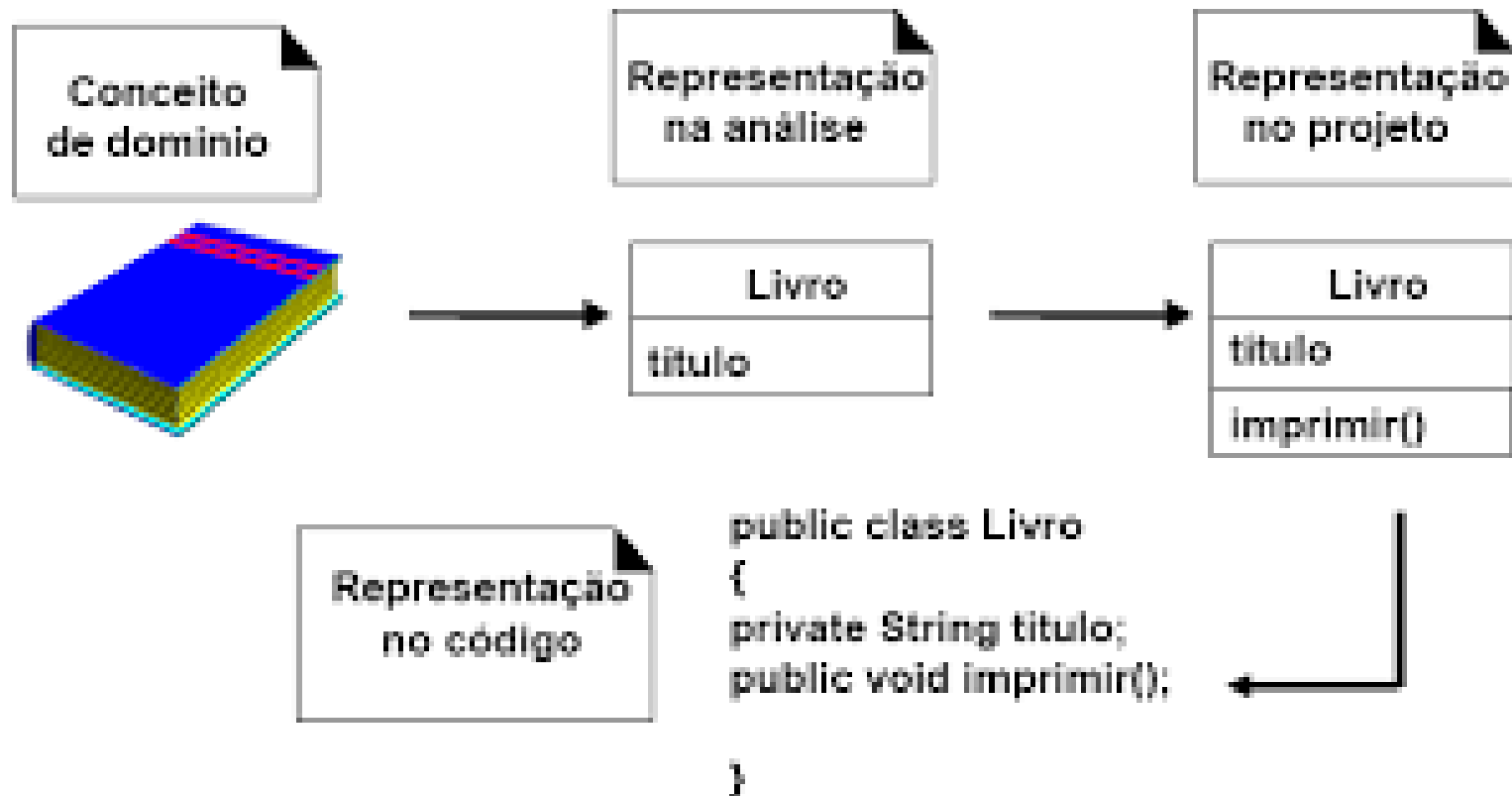
Projeto =
Informação
importante
para o
programador
apenas



O que é Análise e Projeto Orientados a Objeto (APOO)?

- A perspectiva empregada é de *objetos* (coisas, conceitos ou entidades).
- Durante a **Análise** OO, a ênfase está em achar e descrever objetos (ou conceitos) no domínio do problema.
- Durante o **projeto** orientado a objeto, a ênfase está em achar objetos lógicos de software que poderão ser eventualmente implementados usando uma linguagem de programação OO.
 - Tais objetos podem ter atributos e métodos.

O que é Análise e Projeto Orientados a Objeto (APOO)?



Detalhes sobre a análise

- A análise gera um modelo para **entender o domínio do problema**.
- Análise também trata em alto nível de como uma solução possível pode ser montada para atender aos requisitos.
 - Acaba gerando uma especificação, mas **sempre do ponto de vista do usuário** e tratando apenas do domínio do problema.
- Não trata de detalhes de implementação.
- Objetos tratados são sempre do domínio do problema (business objects).
- Muitos diagramas UML podem ser usados.
 - O modelo **é para o cliente e não para o programador**.

Detalhes sobre o projeto (design)

- O projeto é uma extensão do modelo de análise **visando sua implementação** num computador.
- Novos objetos aparecem, mas não são do domínio do problema.
- O resultado **é para o programador ver, não o cliente**.
- Objetos da análise são (geralmente) mantidos e são embutidos numa infra-estrutura técnica.
- As classes técnicas ajudam os business objects a:
 - Serem persistentes
 - Se comunicarem
 - Se apresentarem na interface do usuário
 - Terem desempenho aceitável (usando caches ou threads, por exemplo)

Conceitos

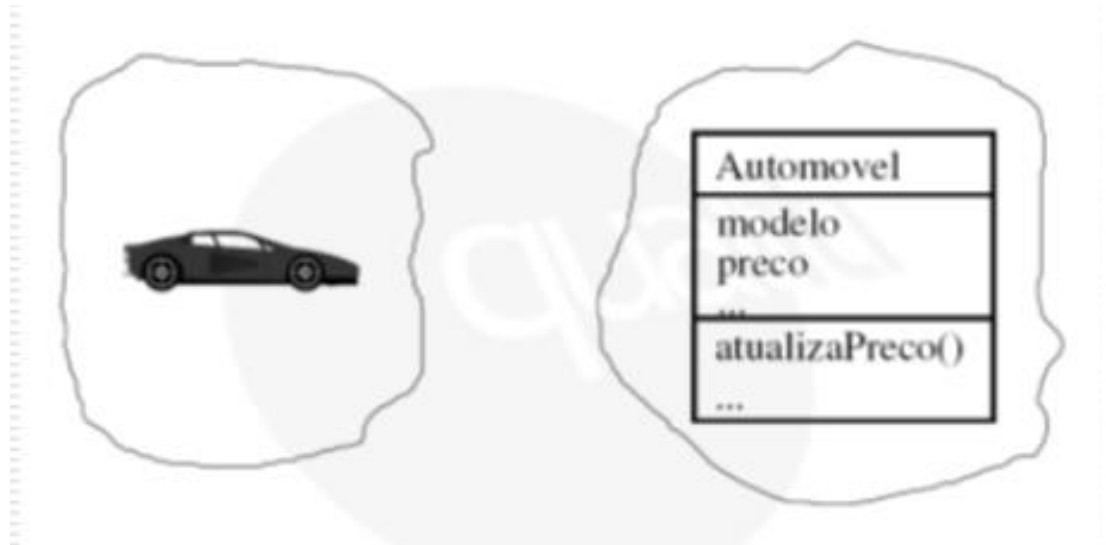
Objetos:

- Tudo que é perceptível por qualquer dos sentidos.
- São coisas do mundo real.
- Cada objeto possui características distintas como: Altura, peso, cor, etc...

Conceitos

Abstração:

- Construção de um modelo para **representação** da realidade.
- Concentrado nas características e o gerenciamento de suas complexidades.



Conceitos

Orientação O.O.:

- Modelar objetos do mundo real, definindo classes, características e funcionalidades dentro de um contexto.
- Contexto → Domínio da aplicação.

Conceitos

Classe:

- Principal mecanismo de abstração.
- Objetos → Classe



Conceitos

Classe:

- Descreve um conjunto de objetos que compartilham características (atributos e operações) em comum.
- Classe encapsula DADOS e OPERAÇÕES.

Conceitos

Atributos:

- São recursos de uma classe.
- Propriedades, elementos de dados.
- “Variáveis” de instancia.

Conceitos

Operações:

- Comportamento.
- Serviços.
- Funções.

Conceitos

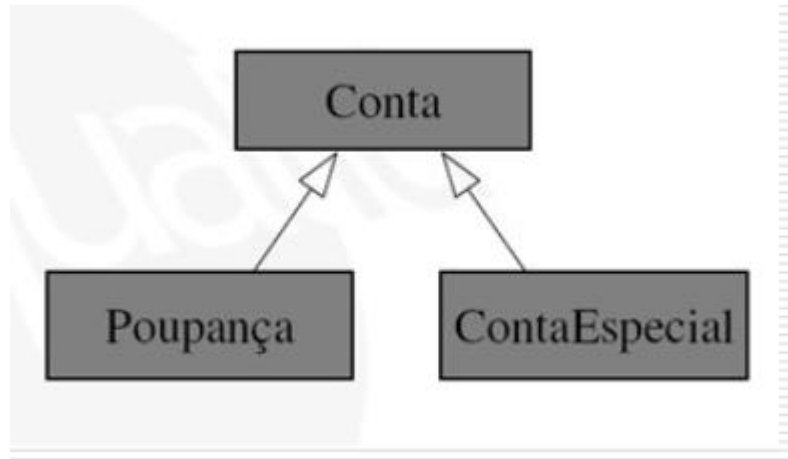
Encapsulamento:

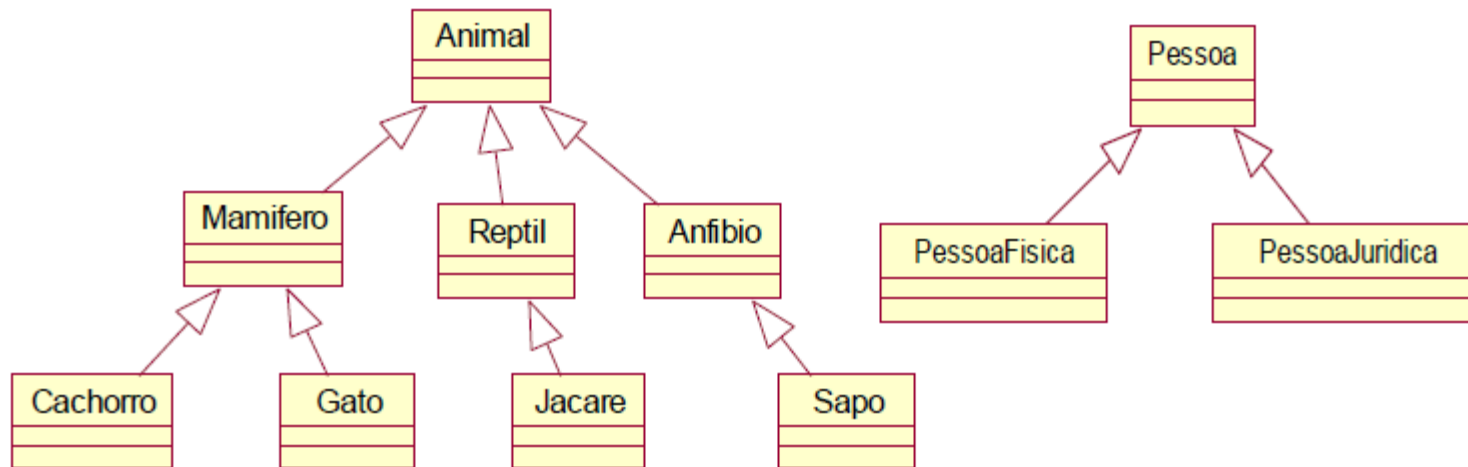
- Encapsulamento de dados e operações.
- Ocultamento da informação
- Independência de aplicação.

Conceitos

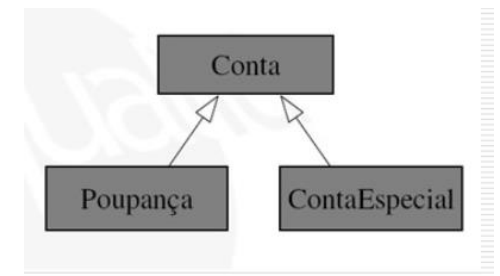
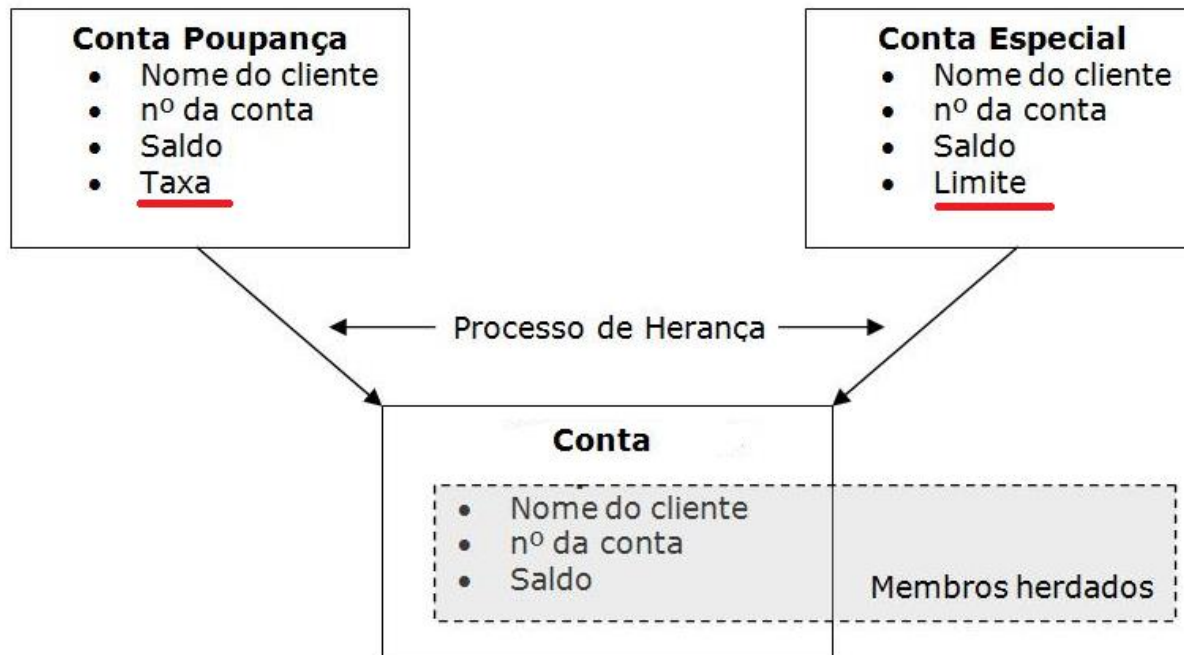
Herança:

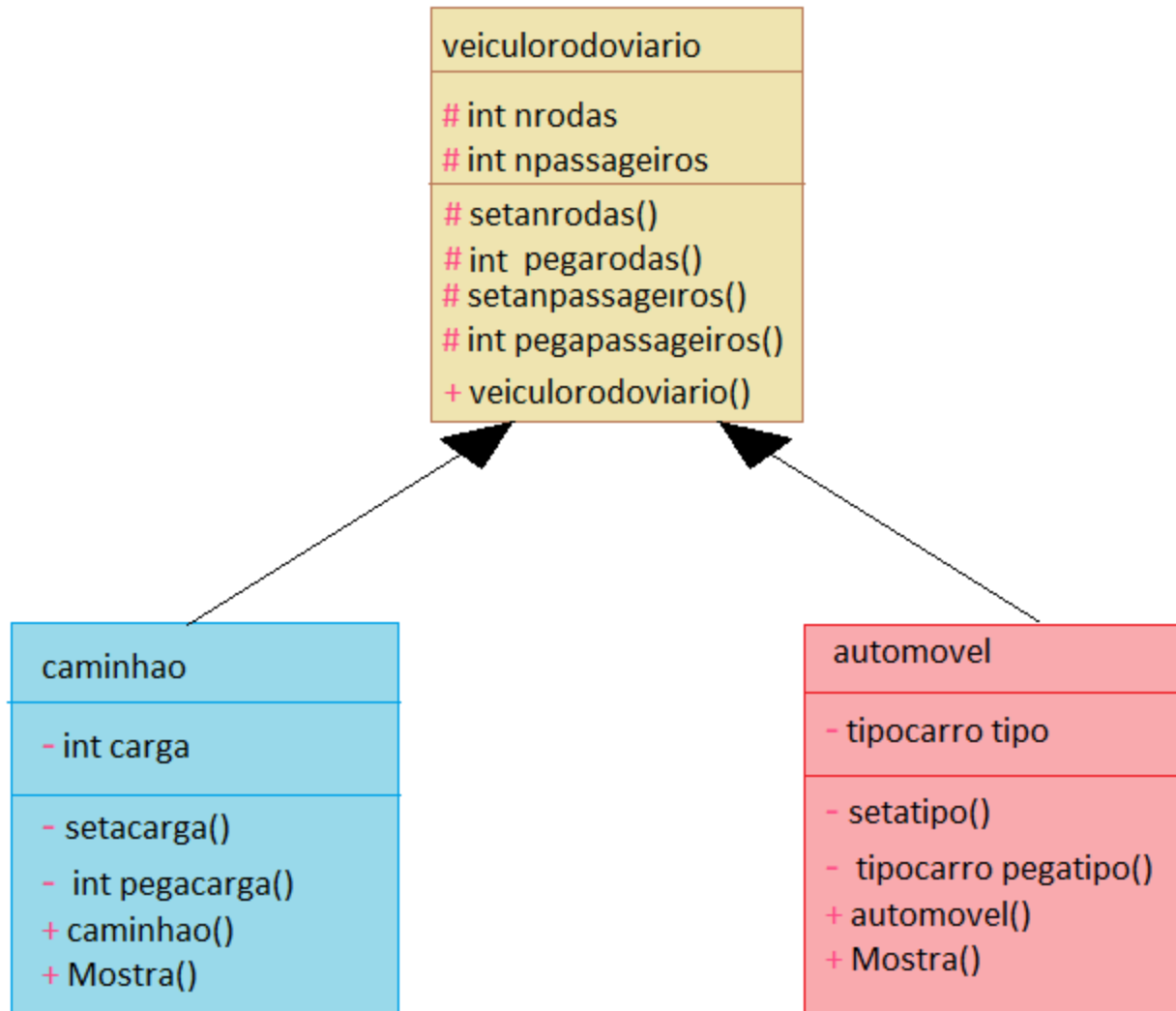
- Classes semelhantes que possuem mesmas características mas com algumas próprias.





Herança acontece quando duas classes são próximas, têm características mútuas mas não são iguais e existe uma especificação de uma delas.





Conceitos

Objeto:

- Modelo de um objeto real, conceitual ou de software.
- Entidade física.
- Possui comportamento, Estado e **identidade**.

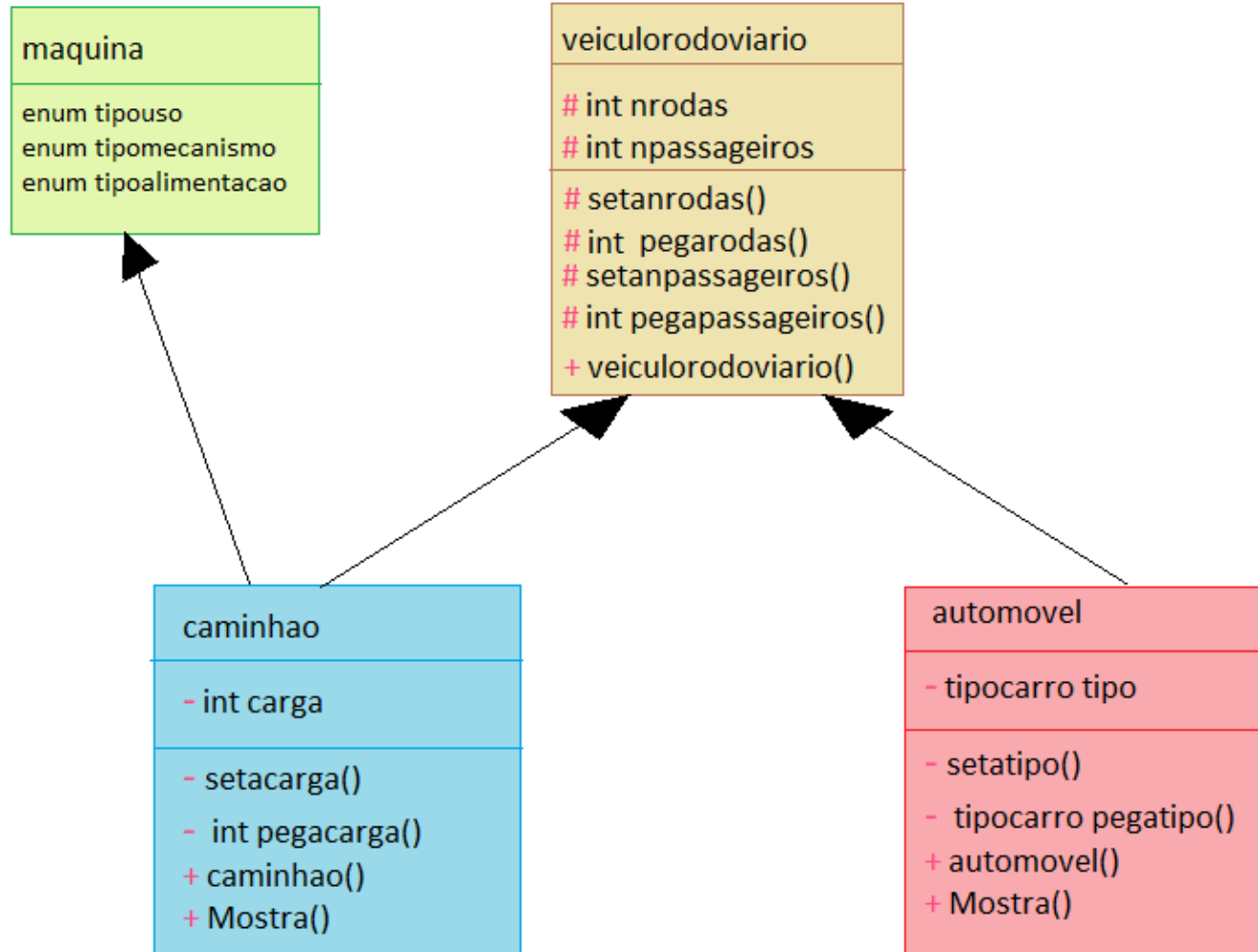
Conceitos

Operação:

- Modela o comportamento da classe.
- Serviços oferecidos.



Outro exemplo



Analise→Projeto→Implementação/Programação

Registrar Novo:

☒ Caminhão

NOVO

☐ Automóvel

REGISTRO DE CAMINHÕES

Número de Rodas: 06

Número de Passageiros: 04

Carga que transporta [ton]: 02

Uso

☒ Veículo de carga

☐ Transporte de passageiros

☐ Carga Viva (animais)

Controle

☐ Automático

☐ Eletrônico

☒ Mecânico

Combustível

☐ Elétrico

☐ Óleo Diesel

☒ Gasolina

REGISTRAR CAMINHÃO

REGISTRO DE CARROS

Número de Rodas: 06

Número de Passageiros: 04

Uso

☐ Van/transporte de passageiros

☒ Automóvel

☐ SUV/Utilitário

Controle

☐ Automático

☐ Eletrônico

☒ Mecânico

Combustível

☐ Elétrico

☐ Óleo Diesel

☒ Gasolina

Prática

- Proponha 3 CLASSES.
- Descreva cada classe.
- Descreva os atributos de cada classe.
- Descreva as operações de cada classe.
- Objetos que poderiam pertencer a esta classes (pelo menos 2 objetos para cada classe).
- Explique encapsulamento para um dos objetos.
- Explique herança neste exemplo.