

PROGRAMAÇÃO ORIENTADA A OBJETOS

Prof. Júlio Machado

MODELAGEM ORIENTADA A OBJETOS

Projetando Objetos

- De uma forma simples, o projeto orientado a objetos de um sistema pode ser dividido em três etapas:
 - Identificar as abstrações/entidades envolvidas no problema
 - Identificar o comportamento que cada uma destas entidades deve ser capaz de fornecer
 - Identificar os relacionamentos entre essas entidades
 - Identificar as estruturas de dados internas necessárias para implementar o comportamento e relacionamentos desejado

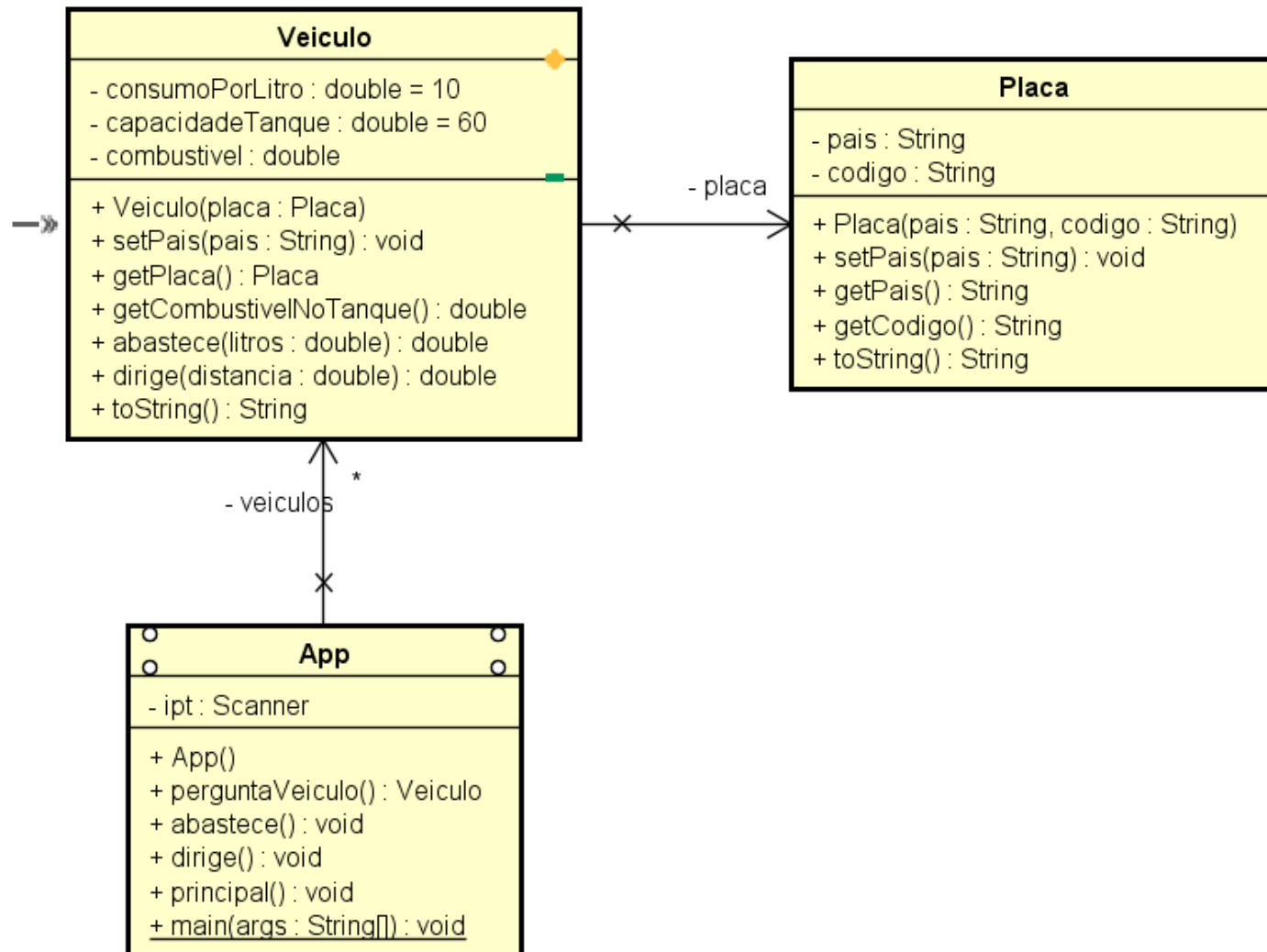
Como posso representar a estrutura de um programa?

```
class Veiculo{  
    ...  
}
```

```
class Placa{  
    ...  
}
```

```
class App{  
    ...  
}
```

Diagrama de Classes



UML

- Unified Modeling Language (<http://www.uml.org/>)
 - Linguagem visual para especificação, visualização, documentação e construção de sistemas de software
 - Padrão para modelagem orientada a objetos
 - Composta por vários diagramas

Diagramas UML

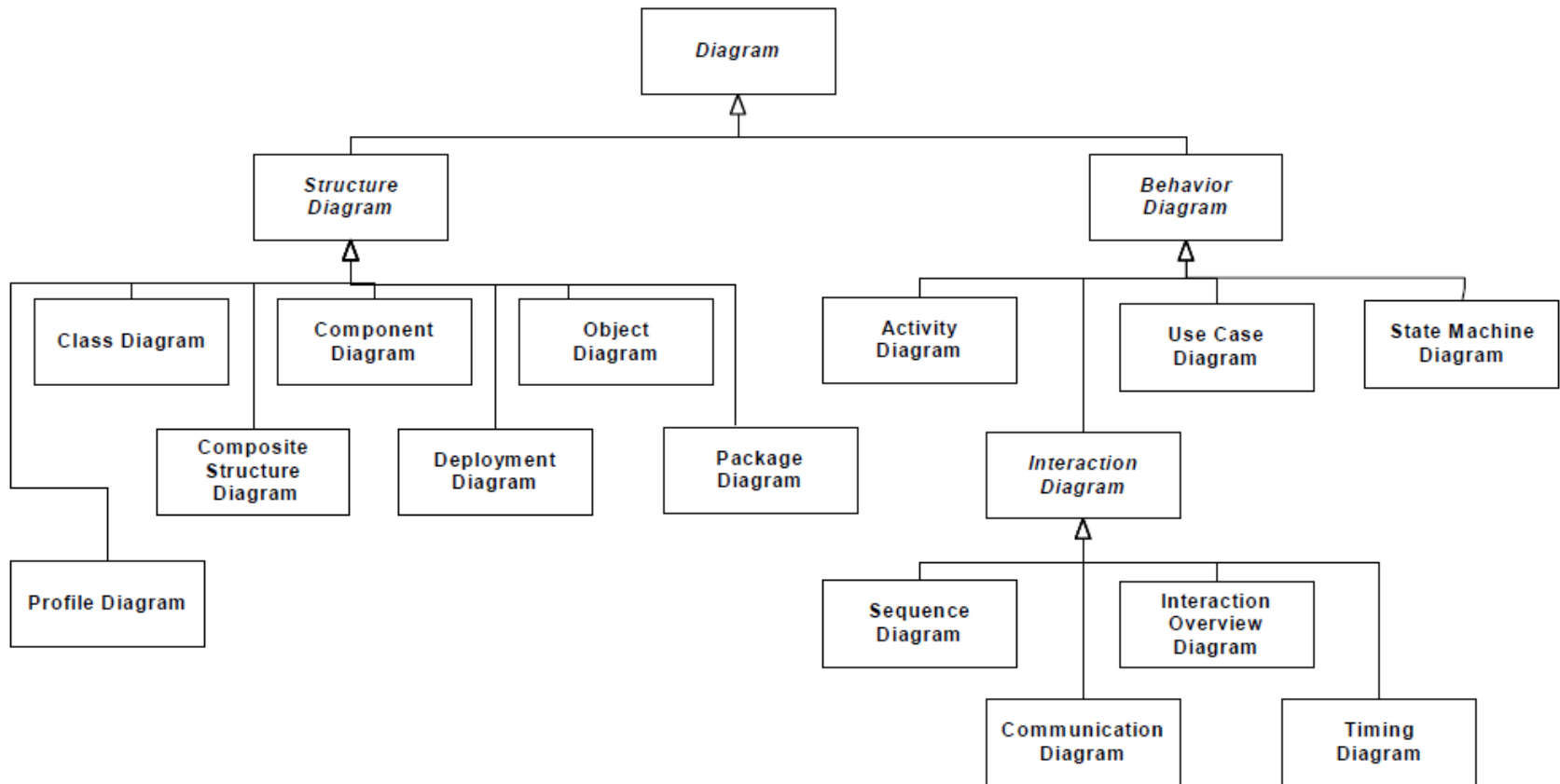


Diagrama de Classes UML

- Denota a estrutura estática do sistema
- Apresenta as classes e seu relacionamentos com outras classes

Diagrama de Classes UML

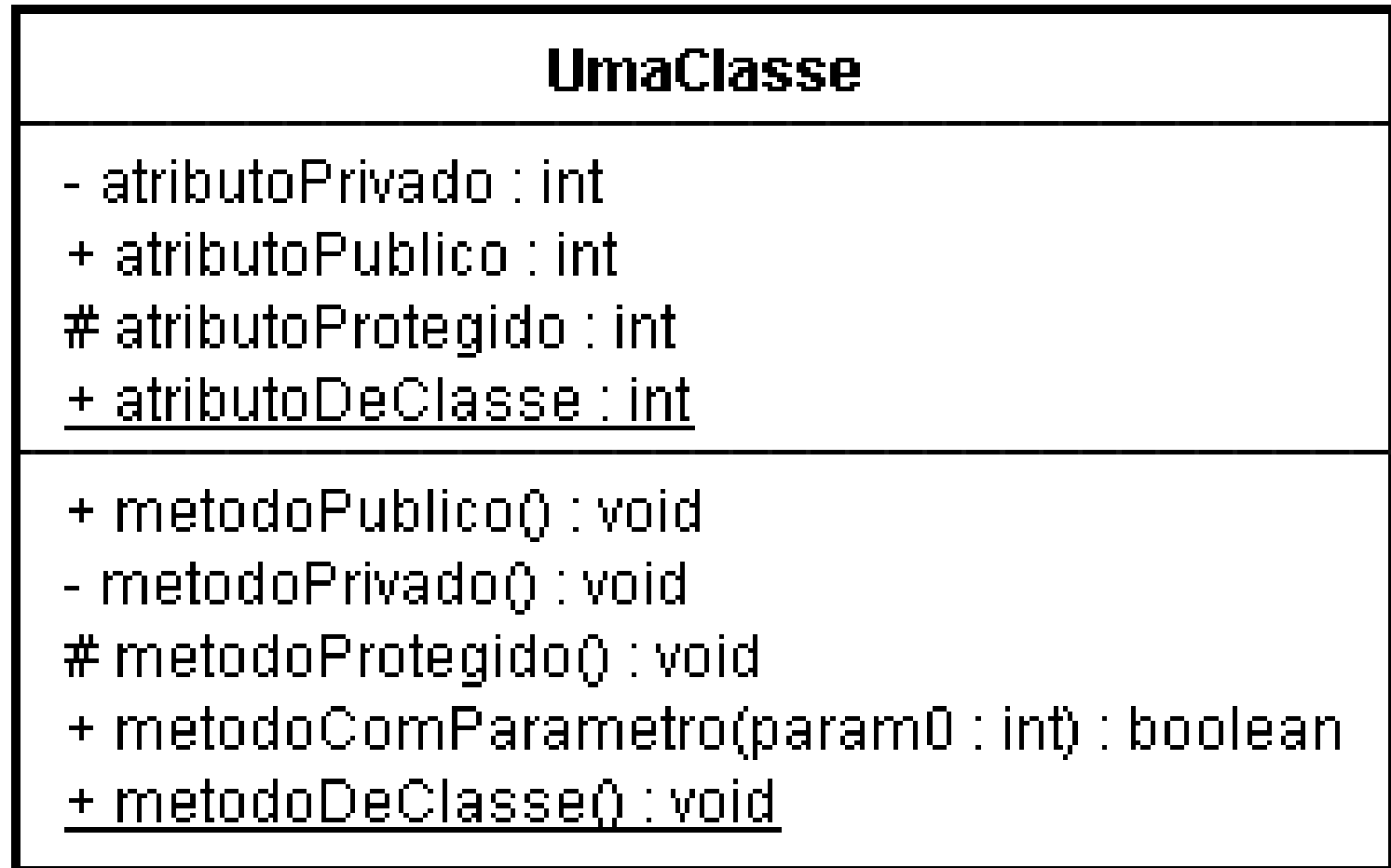


Diagrama de Classes UML

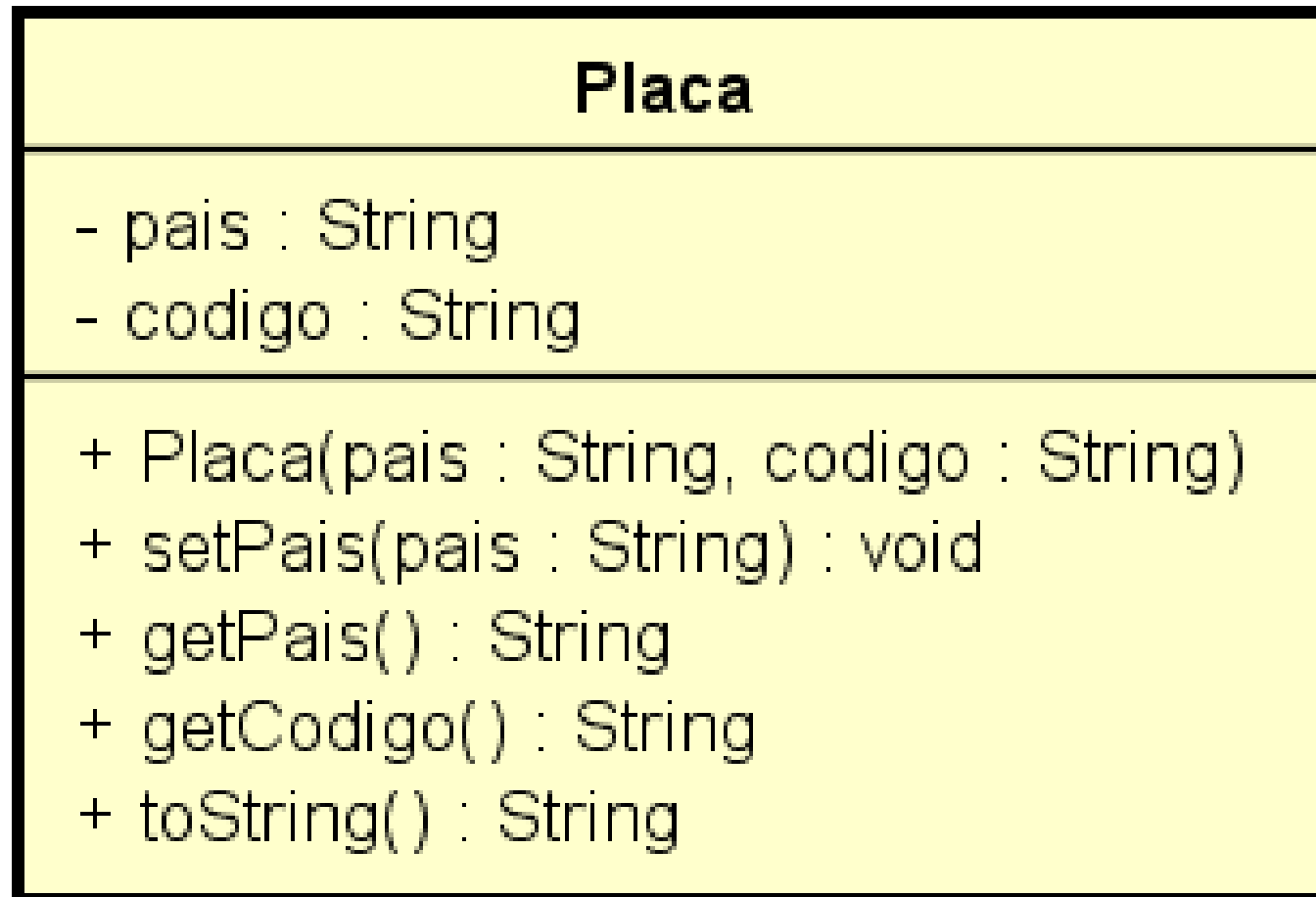


Diagrama de Classes UML

Relacionamentos
entre classes:

- Explicitam a forma pela qual os objetos são compostos, mas uma vez definidos não existe interação entre eles.

Relacionamentos
entre objetos:

- Representam interações entre diferentes objetos.

UML:
relacionamentos
entre objetos

Dependência

Associação

Agregação

Composição

Diagrama de Classes UML

- Relacionamento de dependência:
 - É um relacionamento que significa que um elemento necessita de outro elemento para sua especificação ou implementação
 - É um relacionamento “fornecedor-cliente”
 - Um objeto fornece algo que outro objeto utiliza



Exemplo: relação de dependência

- Os métodos da classe “Persistencia” recebem e/ou retornam instancias de “Turma” por parâmetro
- No método “carregaTurma” dados das turmas são lidos de um arquivo e instancias de turma são criadas

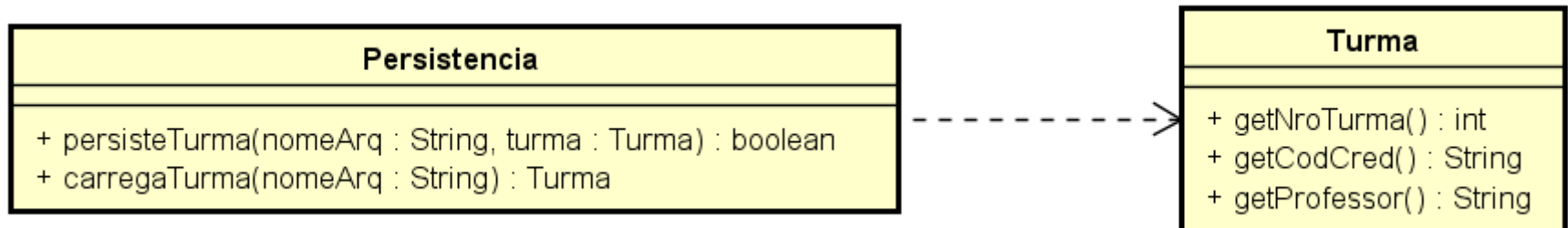


Diagrama de Classes UML

- Relacionamento de associação:
 - É um relacionamento estrutural que descreve um conjunto de ligações, onde uma ligação é uma conexão entre objetos
 - Usualmente implementado através de atributos

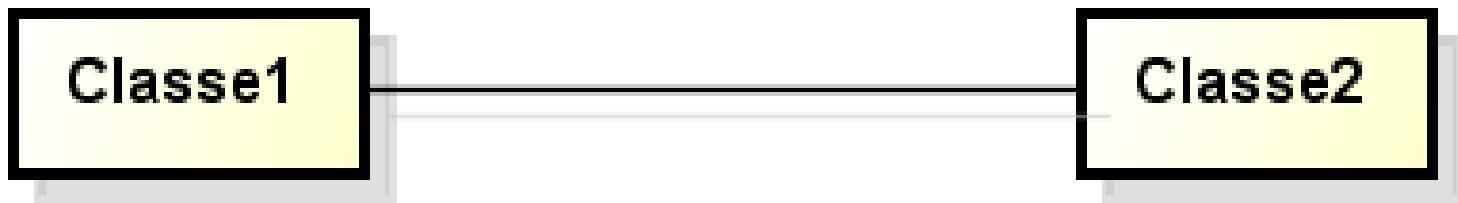
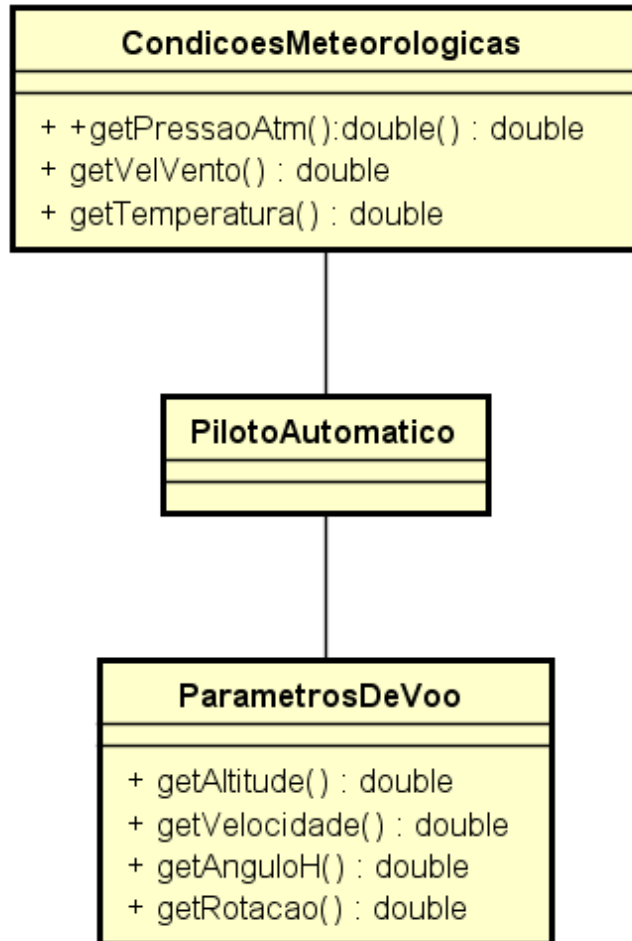


Diagrama de Classes UML

- Relacionamento de associação:
 - Possui diferentes adornos como multiplicidade, direção, navegabilidade, nome, etc



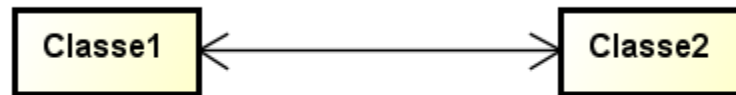
Exemplo: relação de associação



- A classe “PilotoAutomatico” troca mensagens com as classes “CondicoesMeteorologicas” e “ParametrosDeVoo”
- Na prática armazena referências para estas classes

Diagrama de Classes UML

- Relacionamento de associação:
 - Navegabilidade da associação
 - Bidirecional



powered by astah*

- Unidirecional



powered by astah*



powered by astah*

Exemplo: associação com navegabilidade

- A classe “Turma” mantém uma relação (lista) de “Aluno”

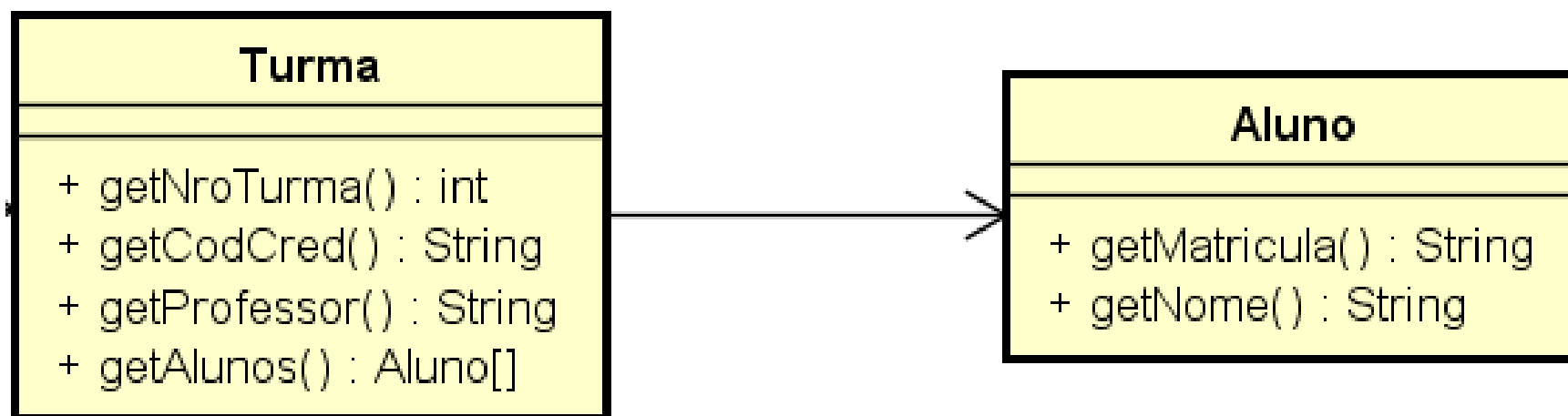


Diagrama de Classes UML

- Relacionamento de associação:
 - Multiplicidade da associação
 - Especifica-se o menor e o maior valor
 - Formato Menor..Maior
 - Valores mais utilizados
 - Menor: 0 (opcional), 1 (obrigatório)
 - Maior: 1 (somente um), * (vários)

Diagrama de Classes UML

- Relacionamento de associação:

- Multiplicidade da associação

- Cliente tem uma única conta (1..1 ou 1)



- Cliente pode ter ou não uma conta



- Cliente tem várias contas, mas no mínimo uma



- Cliente tem varias contas, mas nao e obrigatório (0..* ou *)



Diagrama de Classes UML

Relacionamento de agregação:

Revela uma
associação “todo-parte”
onde as partes tem
ciclo de vida
independente do
agregador

Objeto (o “todo”)
contém referências
para outros objetos (as
“partes”)

Exemplo: relação de agregação

- A classe “Servidor” mantém uma “lista” de “CPUs”. A diferença em relação a associação direcional é uma questão semântica. No caso da agregação o servidor é composto por um conjunto de CPUs interligadas. Além disso, se o servidor for “desmontado” as CPUs continuam existindo.

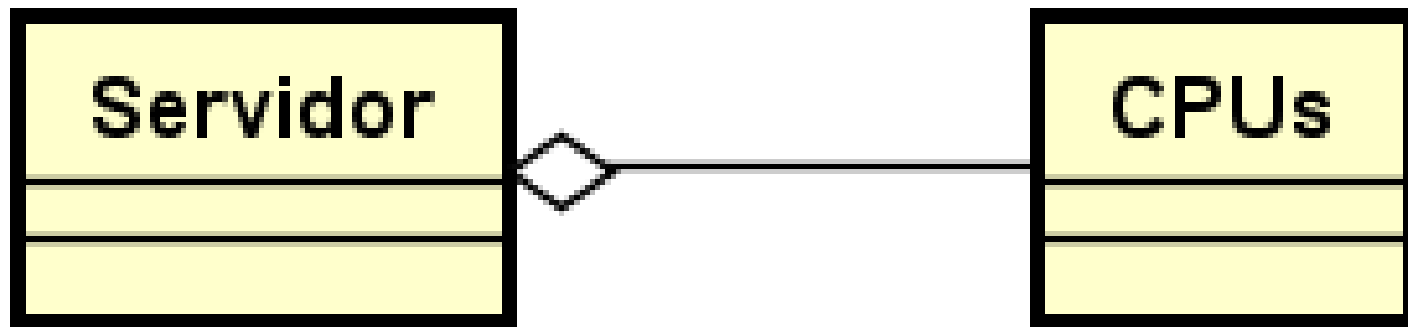
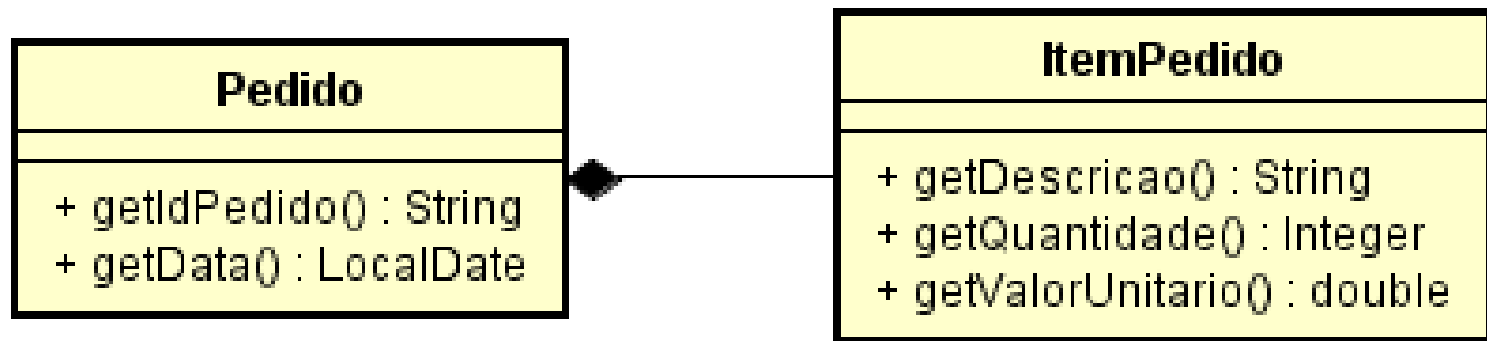


Diagrama de Classes UML

- Relacionamento de composição:
 - Agregação na qual o ciclo de vida das partes depende do ciclo de vida do agregador



- No exemplo, se o pedido é extinto os itens do pedido são extintos também!

Considerações finais

- Diagramas de classe devem ser usados como ferramenta de planejamento
- De nada adianta simplesmente gerar o diagrama automaticamente depois que o sistema está pronto
- Também não é produtivo detalhar excessivamente o diagrama se isso não estiver ajudando no planejamento das classes
- Pode-se colocar no diagrama apenas aquela informação que irá nos ajudar durante o desenvolvimento do programa/sistema