

UML

DIAGRAMA DE CLASSES

Projeto de Programas – PPR0001

Introdução

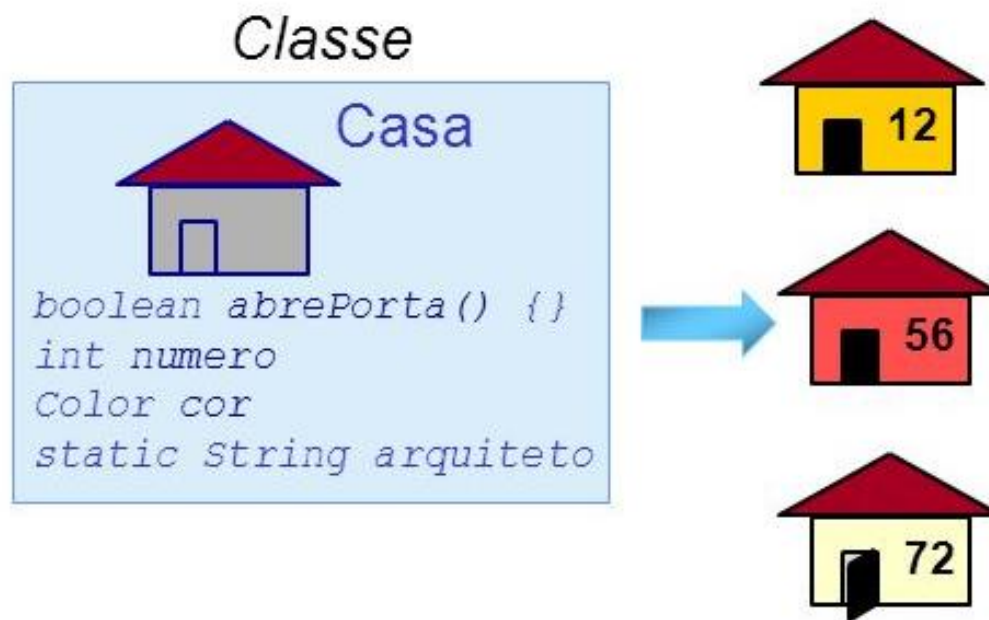
- Diagrama de Casos de Uso → perspectiva externa ao sistema
- Sistema Orientado a Objetos
 - Colaboração entre Objetos pode ser vista:
 - Aspecto Dinâmico → Diagrama de Sequência
 - Aspecto Estático (estrutural) → Diagrama de classes
 - Modelos complementares => Um modelo adiciona detalhes ao outro

Objetos x Classes

- Qual a diferença entre objetos e classes?
 - Classes de objetos são a “forma” dos objetos, a partir dela que os objetos serão criados
 - Objetos são **instâncias** que representam entidades do mundo real
 - Objetos são independentes e encapsulam estado e informação de representação
 - Objetos se comunicam através de troca de mensagens
 - Chamadas de métodos

Objetos x Classes

- Qual a diferença entre objetos e classes?

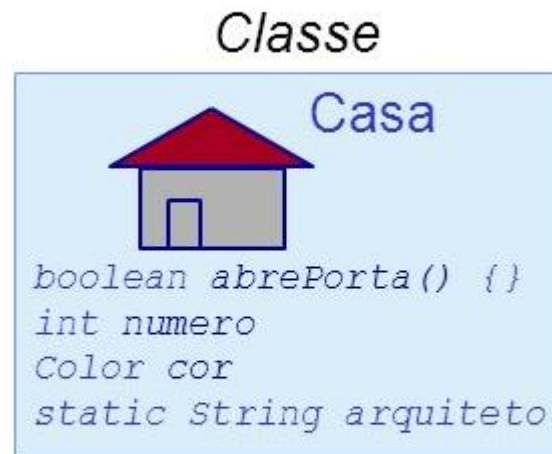


Classes

- Quais são os dados principais de uma classe?

Classes

- Quais são os dados principais de uma classe?
 - Nome
 - Atributos
 - Métodos



Classes

- Quais são os dados principais de uma classe?



Instâncias da classe Casa (objetos)

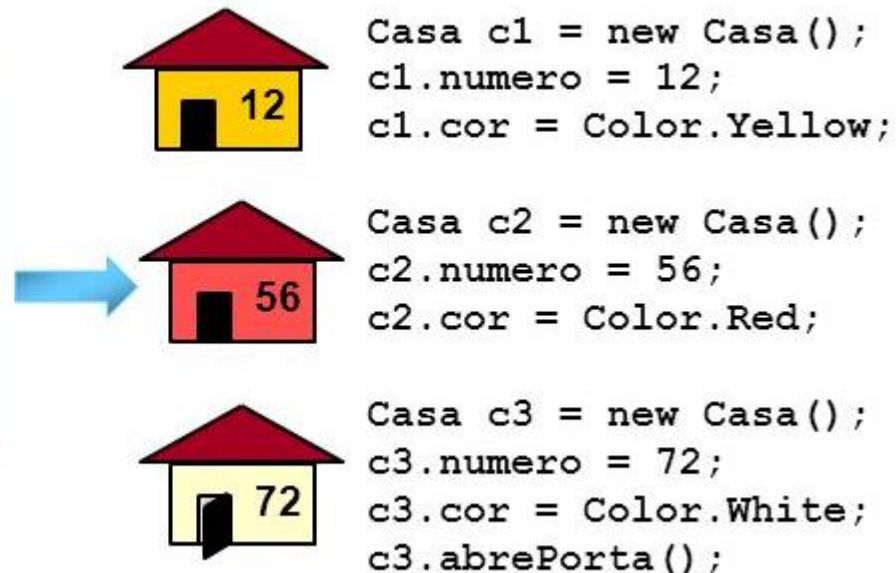


Diagrama de Classes

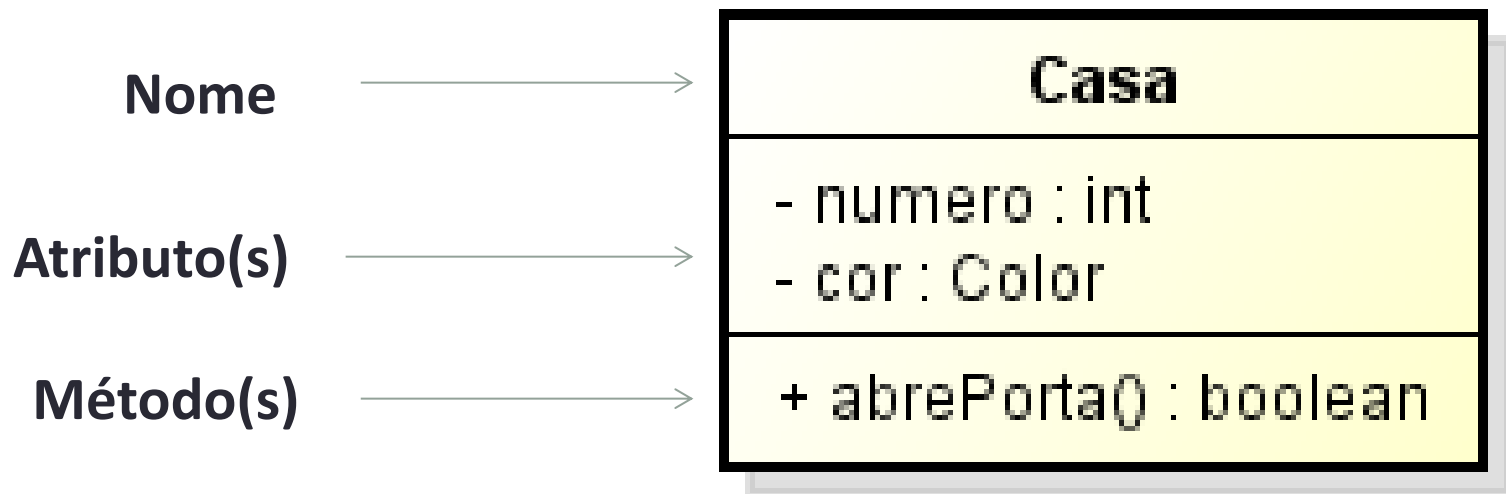


Diagrama de Clases

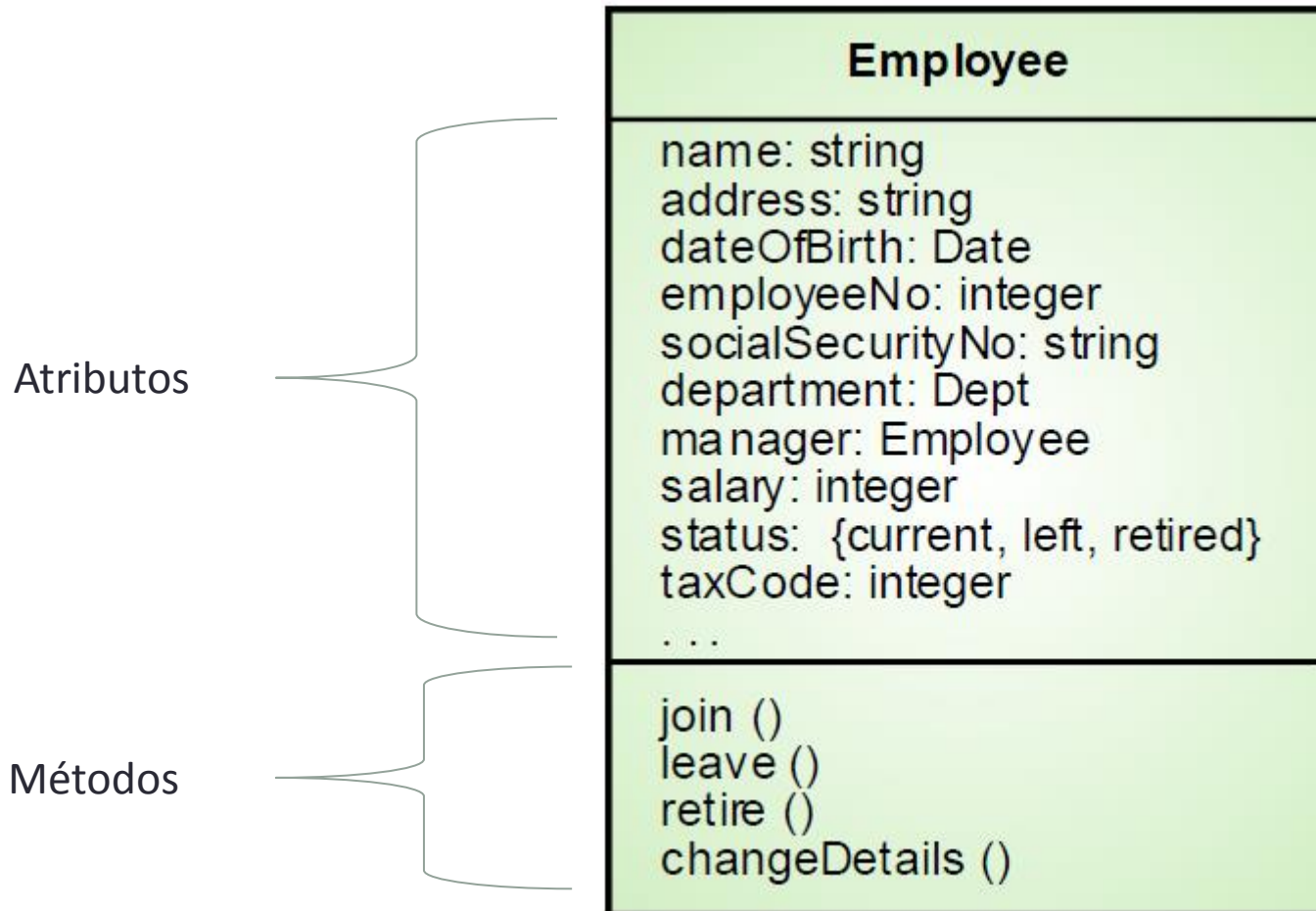
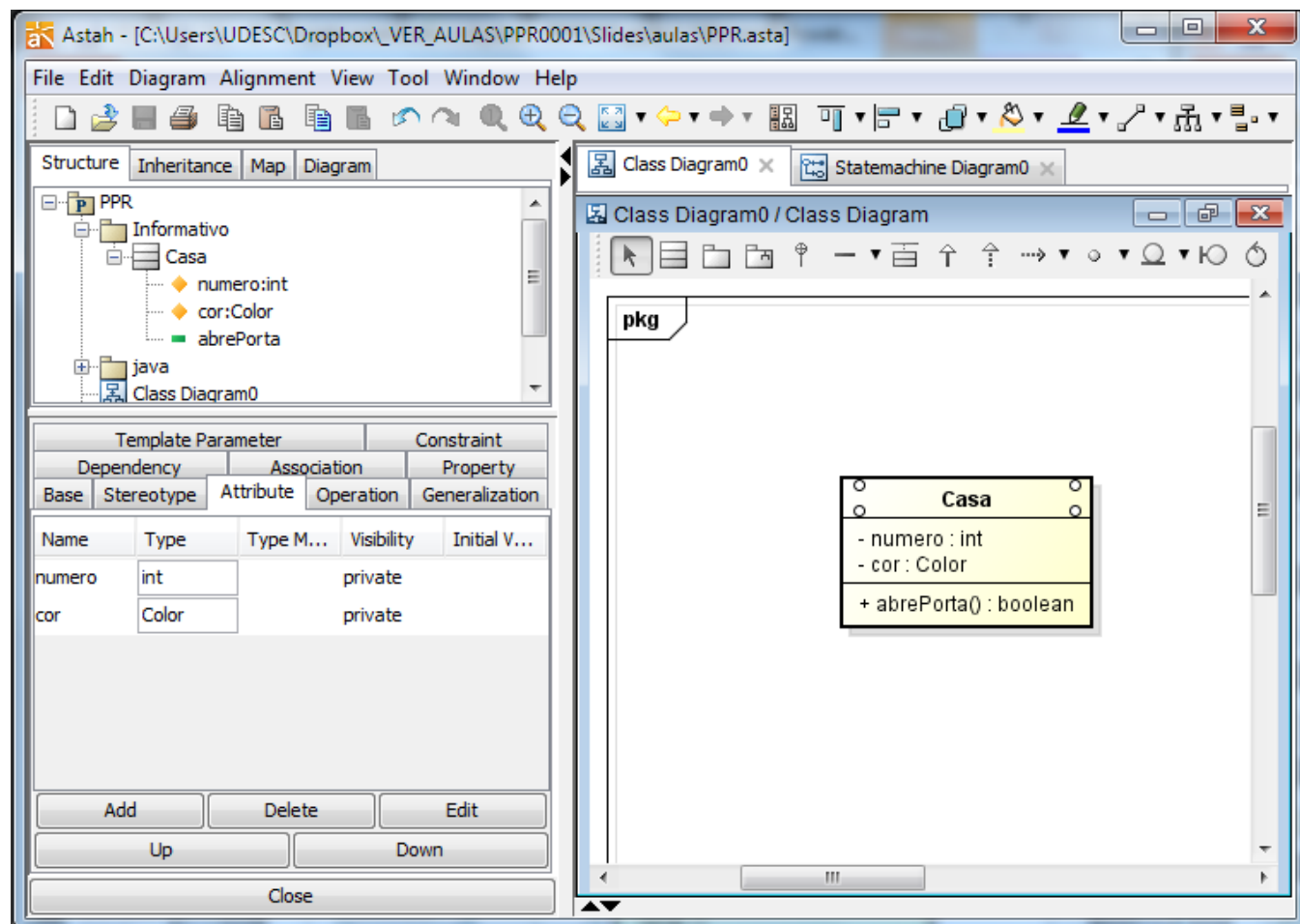


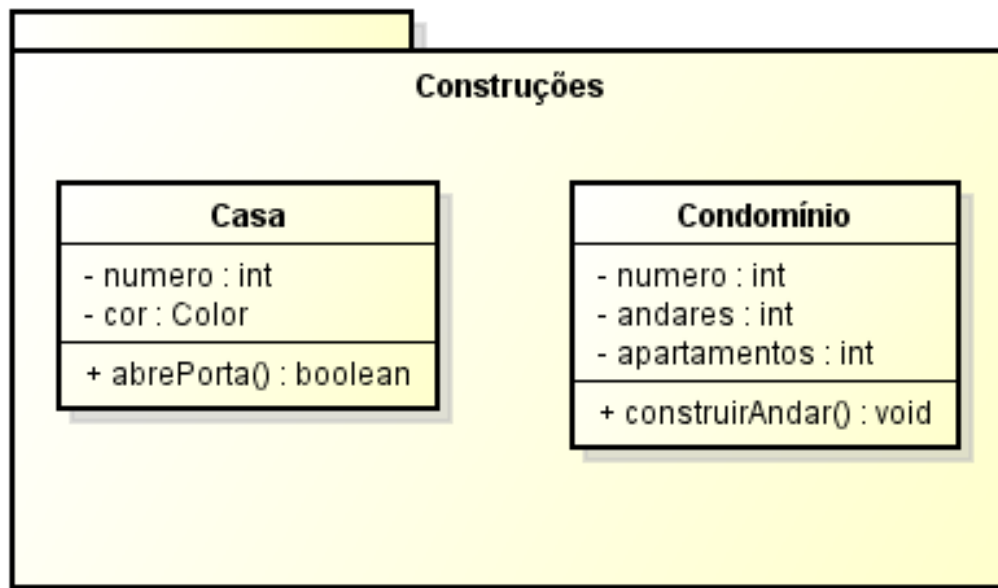
Diagrama de Classes

- Como criar um diagrama de classes no Astah?



Pacotes

- Conceito de Orientação a Objetos utilizado para a organização das classes e demais elementos
 - Agrupar classes de funcionalidades similares ou relacionadas
 - Pacote → Pasta (Diretório)



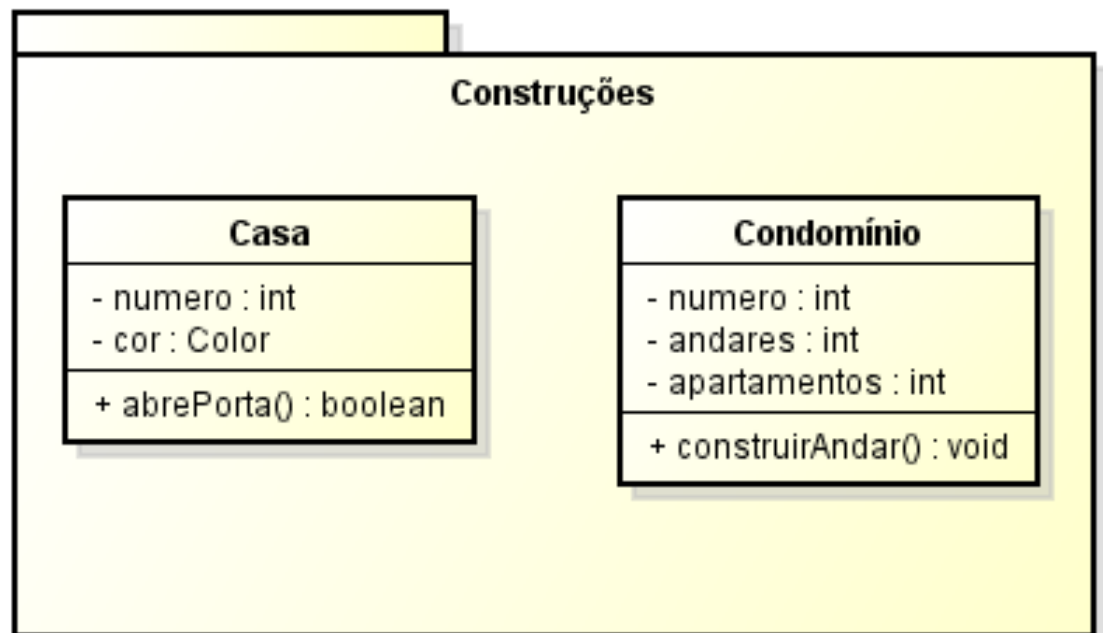
Encapsulamento

- Preocupação na Orientação a Objetos é o encapsulamento:
 - O que só eu posso usar?
 - O que todo mundo pode usar?
- Em C não há como restringir, qualquer um pode alterar o valor de um atributo de uma estrutura
- Em orientação a objetos nós temos os níveis de encapsulamento:
 - **Public:** Todos podem acessar
 - **Private:** Pode ser alterado apenas na classe
 - **Protected:** Apenas classes do mesmo pacote podem alterar (e que herdam)
 - **Package:** (sem modificador) classes do mesmo pacote podem alterar

Encapsulamento

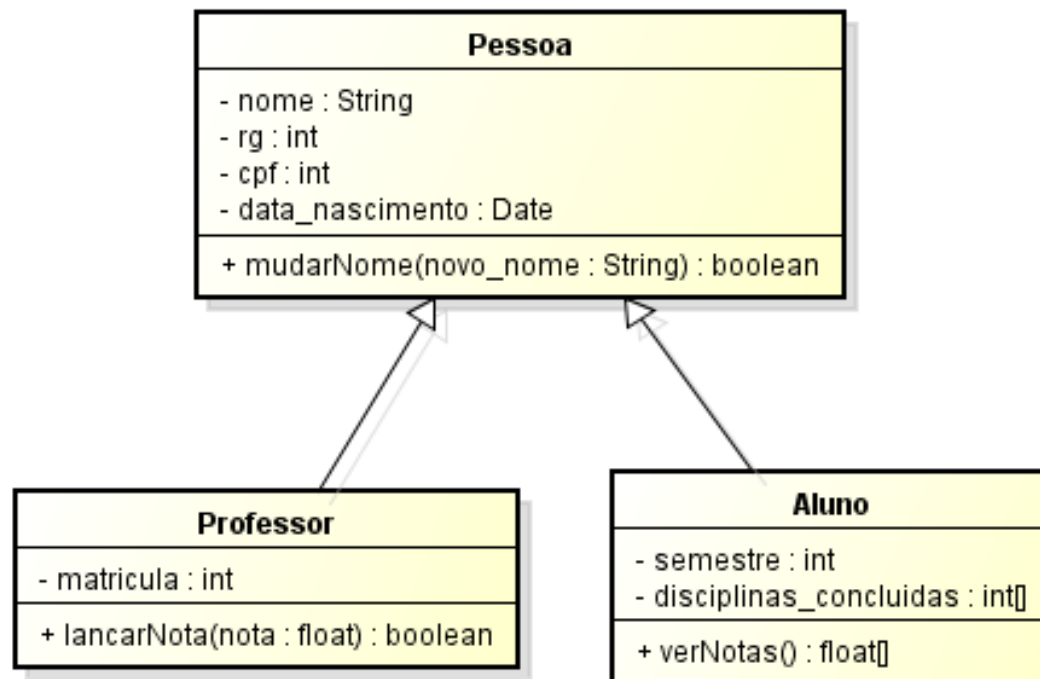
- Preocupação na Orientação a Objetos é o encapsulamento:

- **Public:** +
- **Private:** -
- **Protected:** #
- **Pacote:** ~



Relacionamentos entre Classes

- **Generalização** ou **Herança** também é representada nos diagramas de classe:
- **OBS:** uma generalização de ator ou caso de uso não implica necessariamente numa generalização de classes



Relacionamentos entre Classes

- **Associações**: Indica o relacionamento entre duas classes
Professor - Disciplina
- **Classes Associativas**:
Pessoa – Empresa → Emprego
- **Agregação**: relacionamento optativo entre duas classes
Jogador – Equipe / Equipe – Associação Esportiva
- **Composição**: relacionamento obrigatório entre duas classes
Automóvel – Motor / Automóvel – Roda

Separar atributos (agregação/composição) ou deixar tudo junto?

Bibliografia

- **Básica:**

BEZERRA, E. Princípios de Análise e Projetos de Sistemas com UML. Rio de Janeiro: Campus, 2003.

PRESSMAN, R.S. Engenharia de Software. São Paulo: Makron Books, 2002.

SOMMERVILLE, I. Engenharia de Software. São Paulo: Addison Wesley, 2003.

- **Complementar:**

WARNIER, J. Lógica de Construção de Programas. Rio de Janeiro: Campus, 1984.

JACKSON, M. Princípios de Projeto de Programas. Rio de Janeiro: Campus, 1988.

PAGE-JONES, M. Projeto Estruturado de Sistemas. São Paulo: McGraw-Hill, 1988.