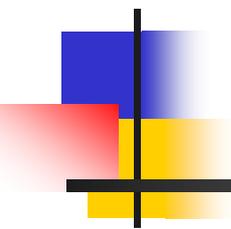


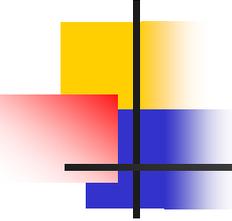
Métodos de Construção de Software: Orientação a Objetos



Mestrado em Ciência da Computação

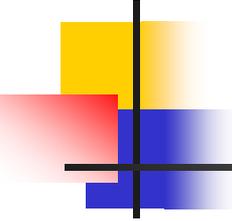
2008

Profa. Itana Gimenes



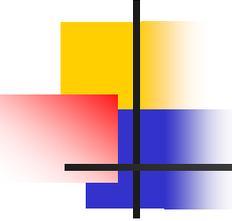
Problemas com Modelagem Funcional

- Falta de aproximação de conceitos com o mundo real – vivemos num mundo de objetos
- Dados e processos separados
- Tipos abstratos de dados



Introdução à Orientação a Objetos (O-O)

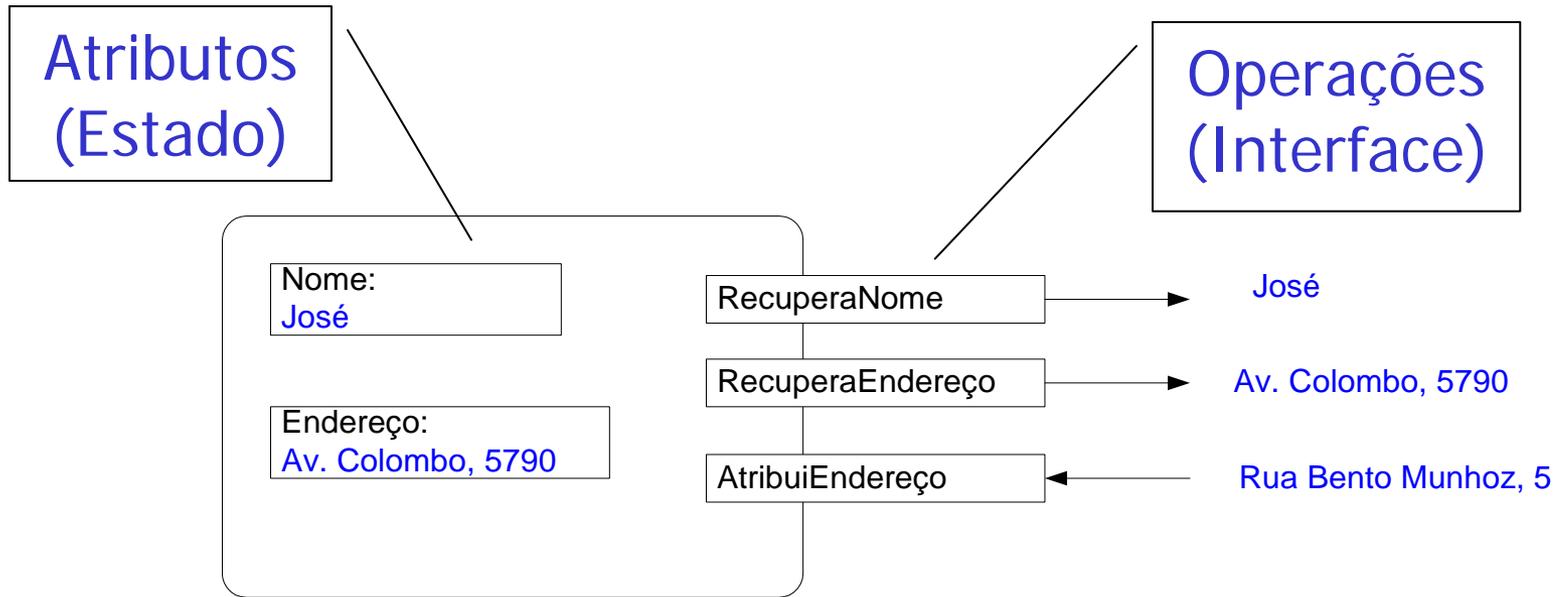
- Iniciou com as linguagens de programação, 1980s.
Ex. Simula, Smalltalk, Eiffel, C++, Java.
- Aplicada à análise e projeto, 1990s
- Grandes apelos
 - Possibilidade de construir software a partir de componentes existentes ao invés de sempre começar do zero.
 - Possibilidade de evoluir entre os estágios de desenvolvimento mantendo consistência entre os modelos e conceitos.

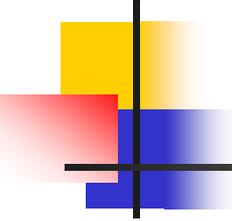


Conceitos de O-O

- **Objeto** é algo similar à uma instância de uma entidade como visto na análise estruturada, pois é uma unidade da qual queremos representar informações no sistema.
- Objetos possuem **atributos**
- Um conjunto de atributos forma o **estado** do objeto
- Objetos possuem **operações (serviços ou métodos)** que manipulam o estado do objeto.
- As operações associadas com um objeto são chamadas de **interface** pois constituem o único meio de manipular o estado do objeto.

Ilustração de Objeto





Comunicação entre objetos

- **Passagem de mensagens**
 - Chamada de uma operação sobre um objeto.
- Um sistema orientado a objetos consiste de vários objetos que se comunicam entre si.

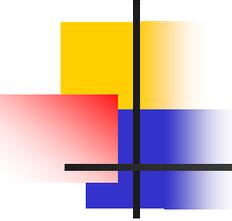
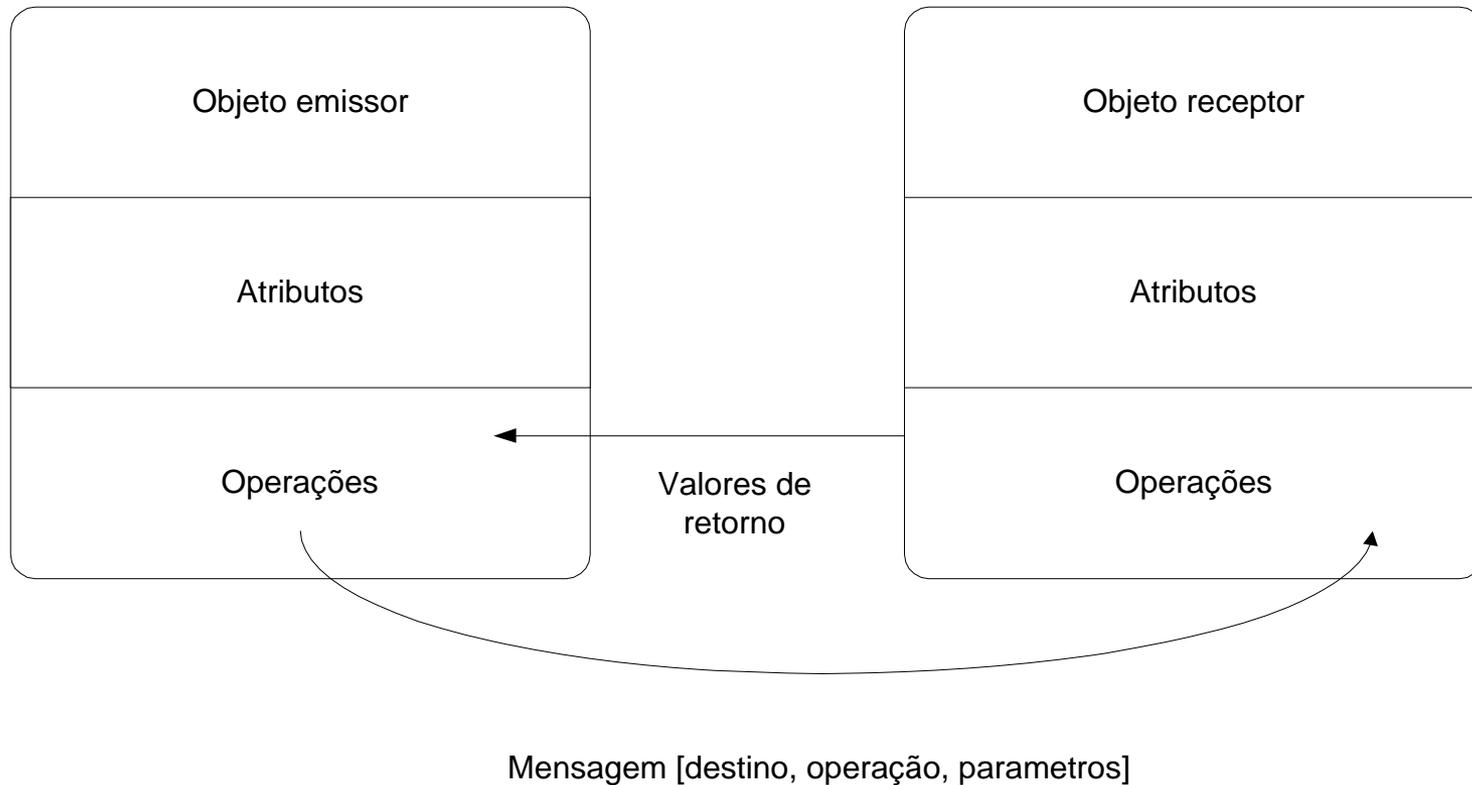
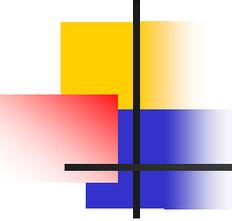


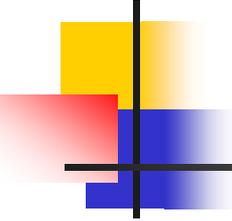
Ilustração de Comunicação





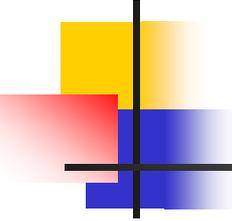
Vantagens O-O

- **Encapsulamento (ocultação de informação)**
 - Os dados associados a um objeto não estão disponíveis diretamente para os usuários do objeto.
 - A única maneira de utilizar os dados é através das operações visíveis na interface.
 - A implementação das operações não são visíveis ao usuário.
- **Independência de dados** – implementação das operações podem ser alteradas sem afetar os usuários dos objetos. A interface continua a mesma, apenas as ações internas são modificadas.



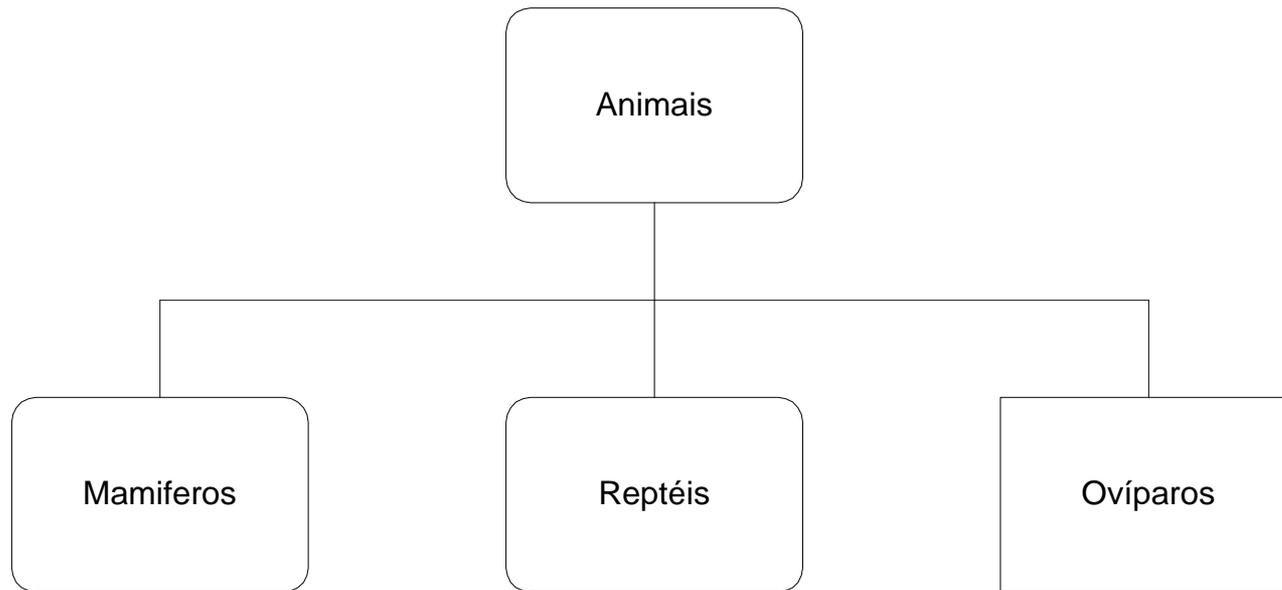
Vantagens O-O

- **Reutilização**
 - A abordagem O-O facilita reutilização pois objetos encapsulam dados e operações.
 - Exemplo: o objeto livro pode ser reutilizado em um sistema de venda de livros.

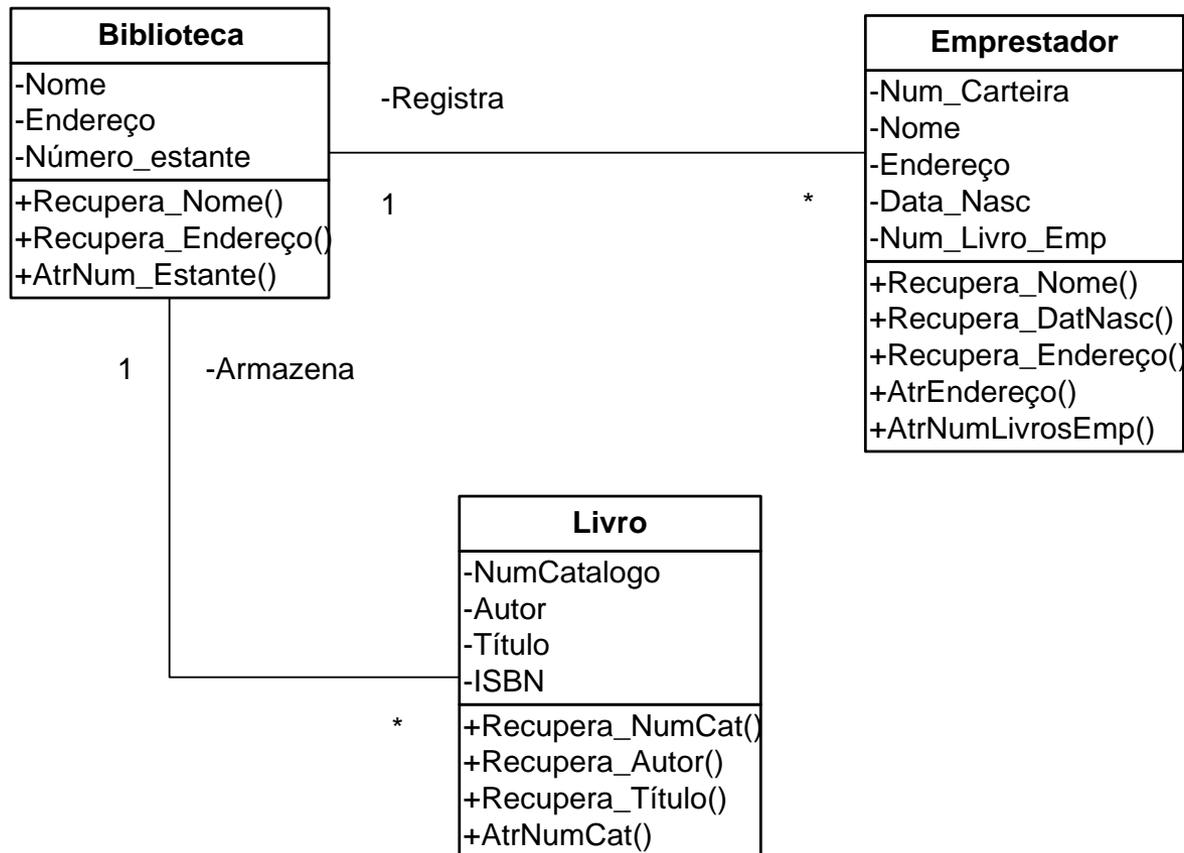


Classes

- Objetos que tem características comuns (atributos e operações) são agrupados em categorias chamadas de **Classes**.

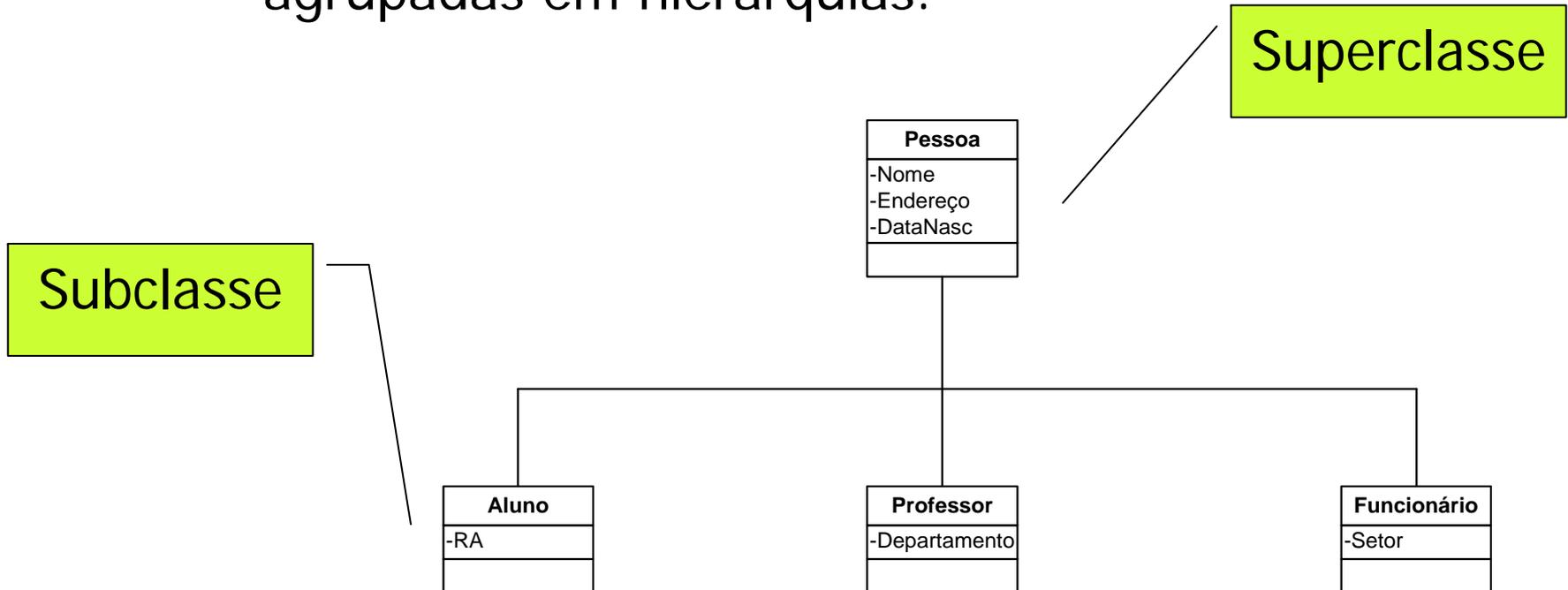


Relacionamento entre classes

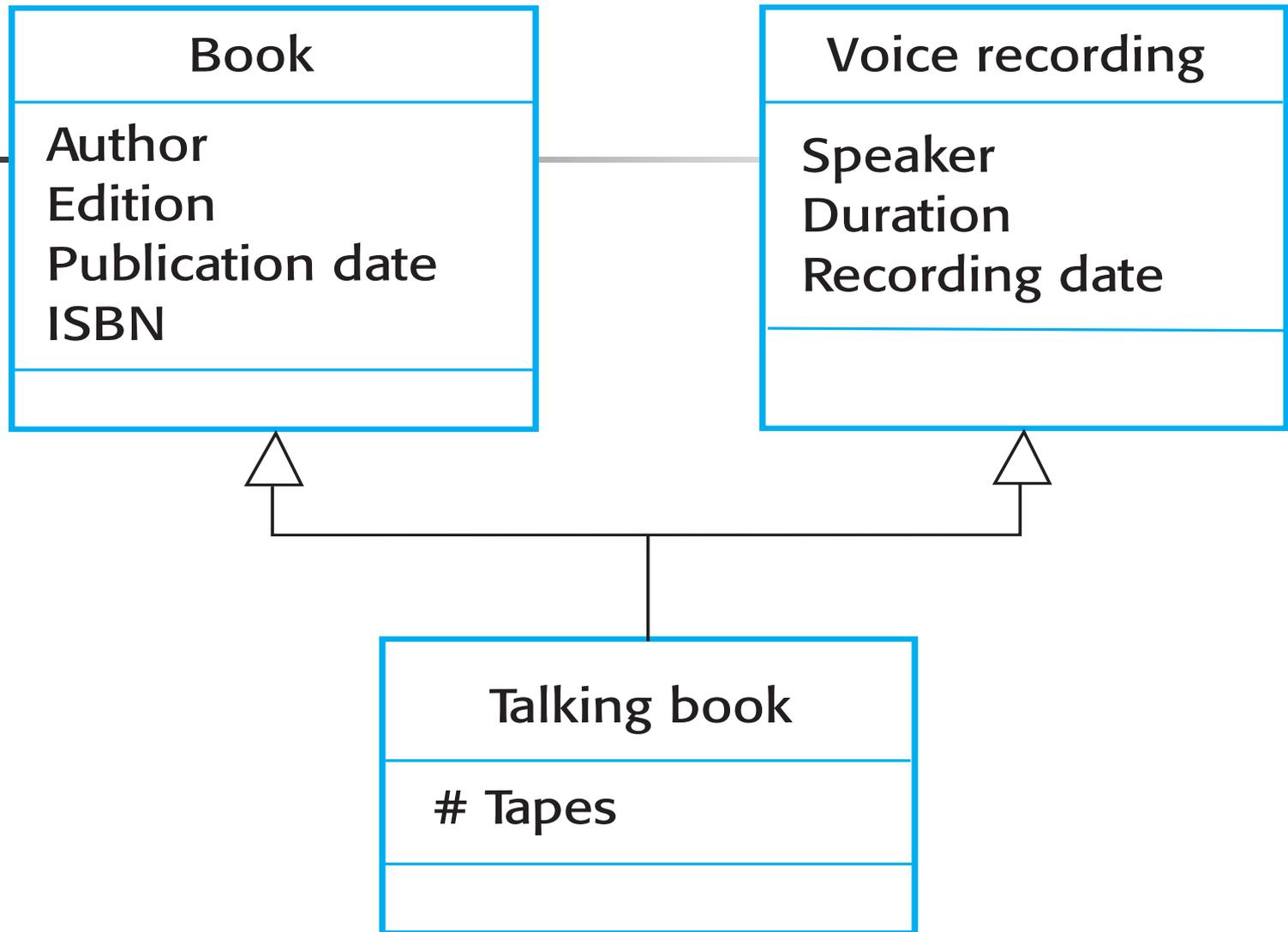


Herança

- Classes com atributos e operações iguais podem ser agrupadas em hierarquias.

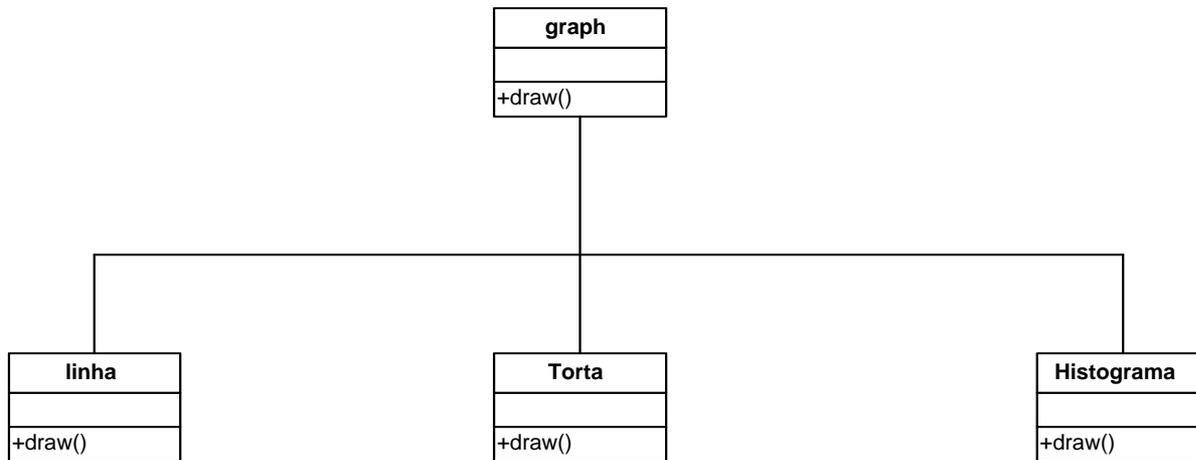


Herança múltipla (Sommerville)

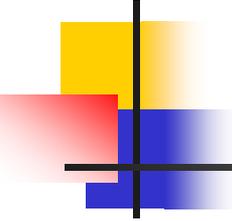


Polimorfismo

- Aplicação da mesma operação a diferentes tipos de objetos.
- Ex. writeln do Pascal que imprime qualquer tipo parâmetro (string, inteiro, real).



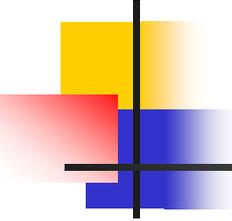
A operação draw() é overloaded .



Exemplos de Código

Programa para mostrar formatos na tela

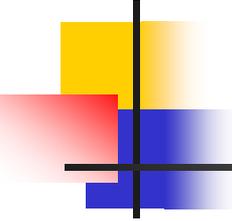
```
class shape {
    point center;
    color co1;
    // ...
public:
    void move(point to) { center=to; draw();}
    point where() {return center;}
    virtual void draw();
    virtual void rotate(int);
    // ...
};
```



Exemplos de Código

Programa para mostrar formatos na tela

```
class circle: public shape {  
    int radius;  
Public:  
    void draw();  
    void rotate(int i);  
    // ...  
};
```

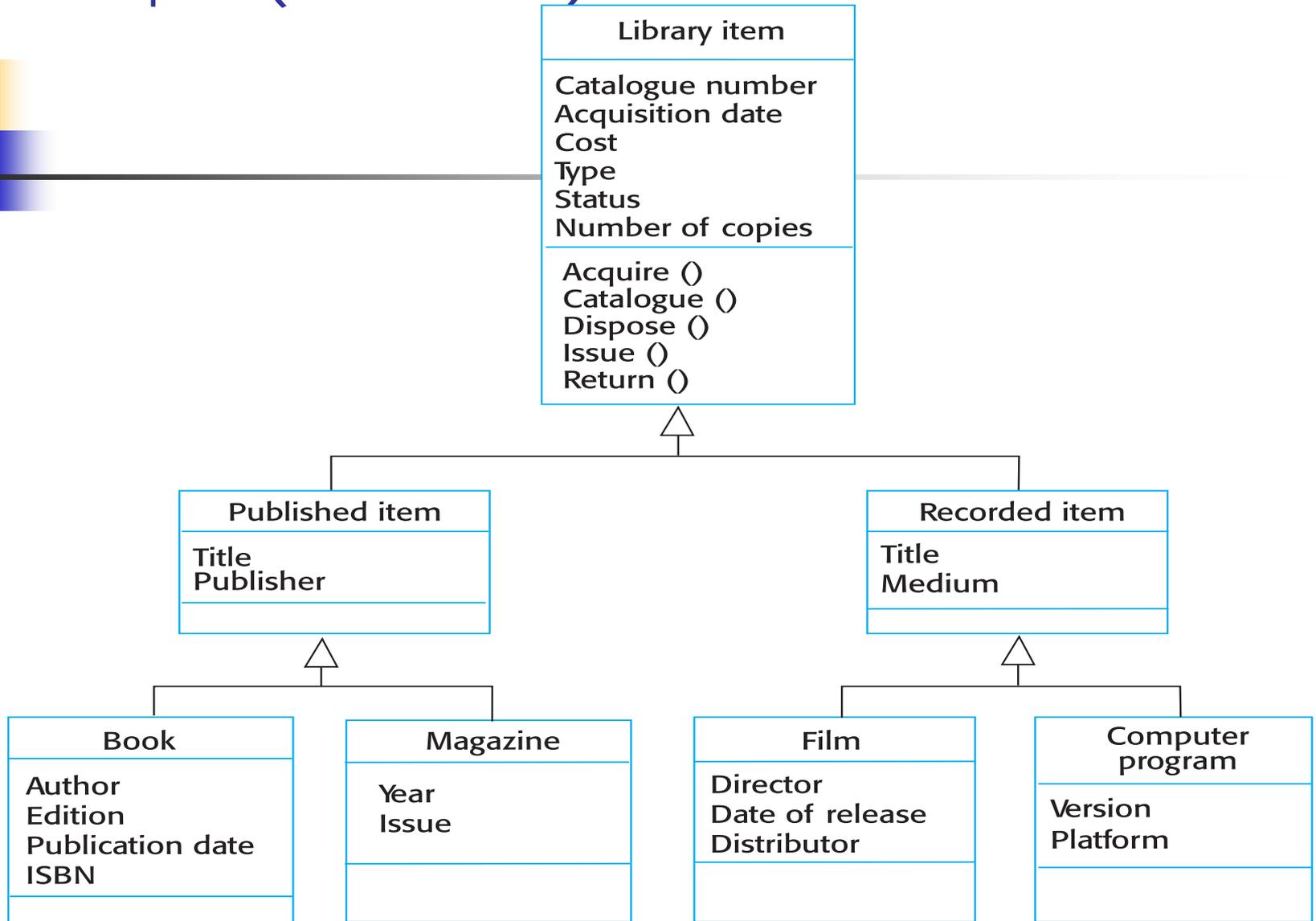


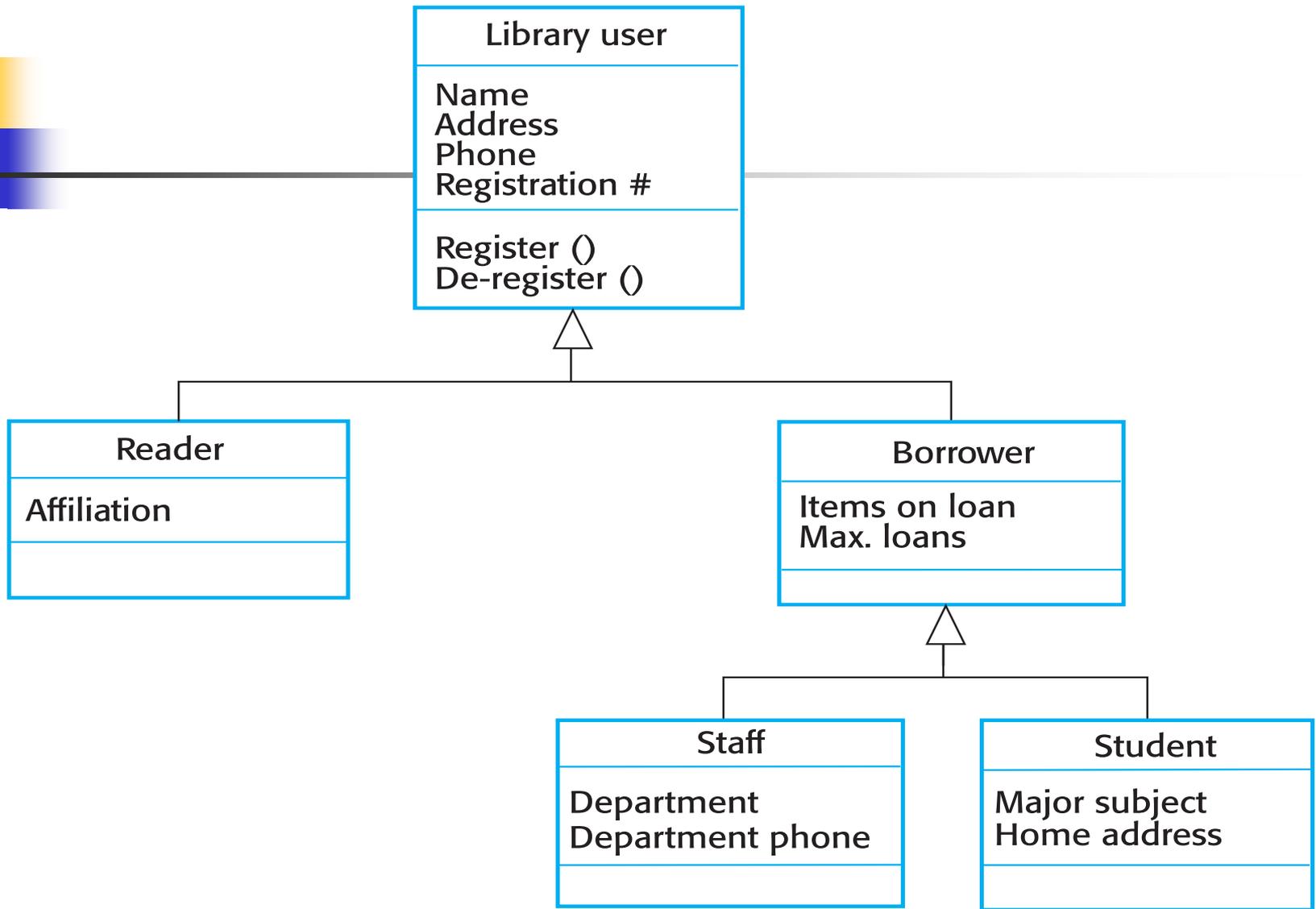
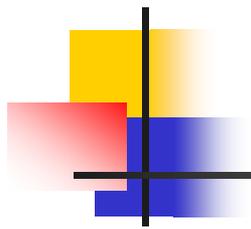
Exemplos de Código

Programa para mostrar formatos na tela

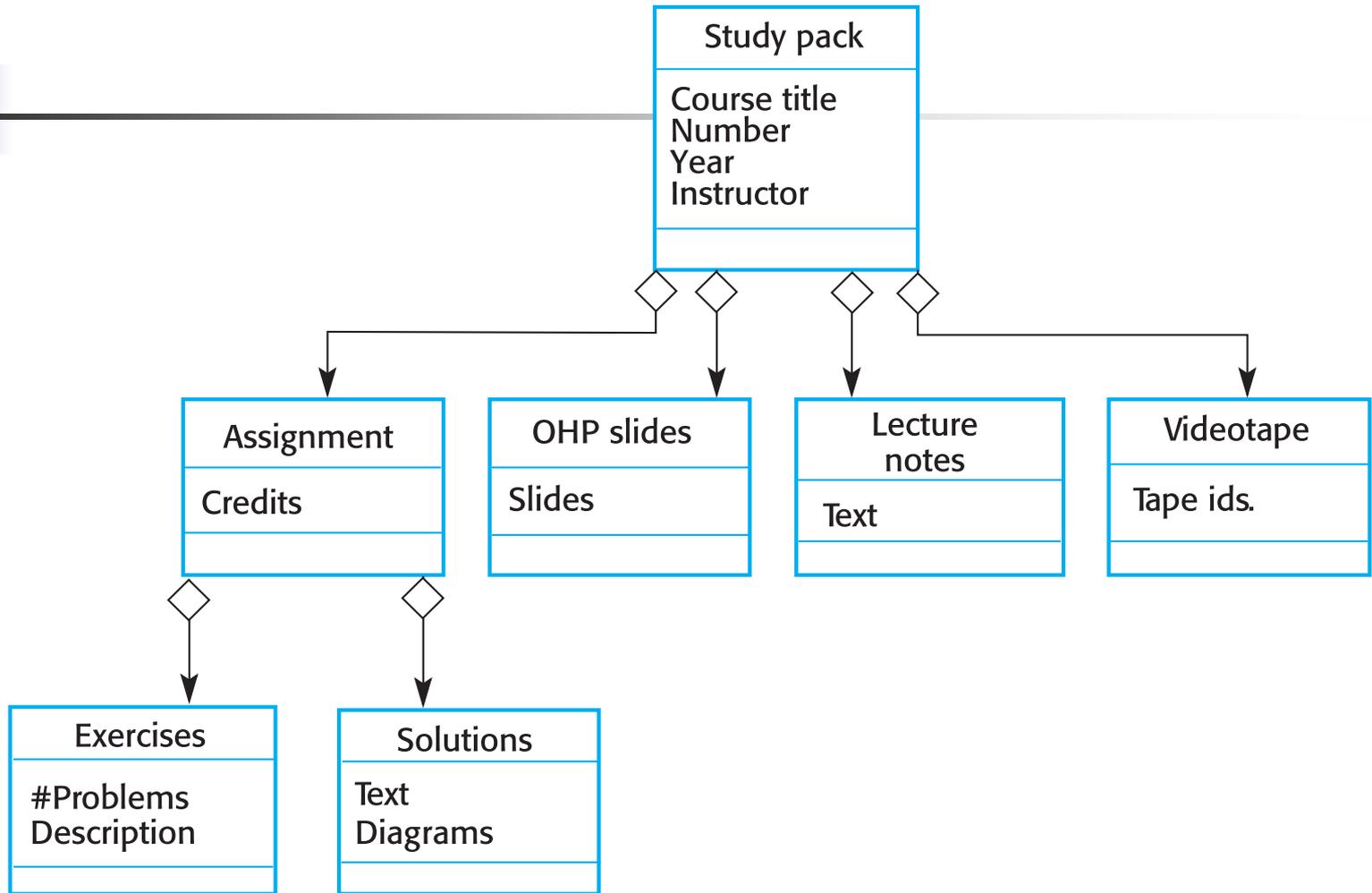
- Criando ou declarando um círculo
`circle* umcírculo = new circle(); circle umcírculo;`
- Referência a umcírculo
`x = umcírculo->where()
umcírculo.where();`
- Vetor para girar todos os formatos de 45°
`for (int i = 0; i<no_of_shapes; i++)
shape_vec[i].rotate(45);`

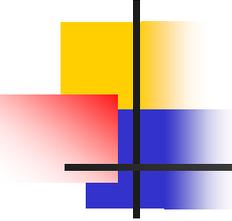
Exemplos (Sommerville)





Agregação de objetos (Sommerville)





Exercício

- Imagine um sistema para controlar cursos de treinamento em uma empresa
 - Quais seriam as classes?
 - Quais os potenciais atributos e operações entre as classes?
 - Quais os relacionamentos entre as classes?
 - Quais as potenciais mensagens entre as classes?

Possível Solução

- Agendar cursos
- Preparar material
- Contratar instrutores
- Marcar exames
- Gerenciar cursos em andamento

