



PUCRS
Pontifícia Universidade Católica
do Rio Grande do Sul

ESCOLA
POLITÉCNICA

Reutilização APIs

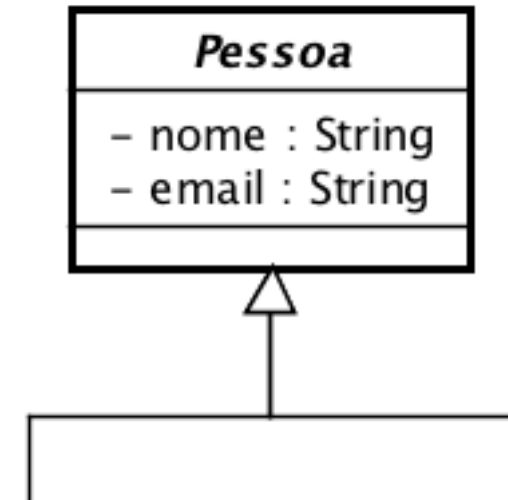
46504-04 - Construção de Software

Prof. Msc. Eduardo Arruda
eduardo.arruda@pucrs.br

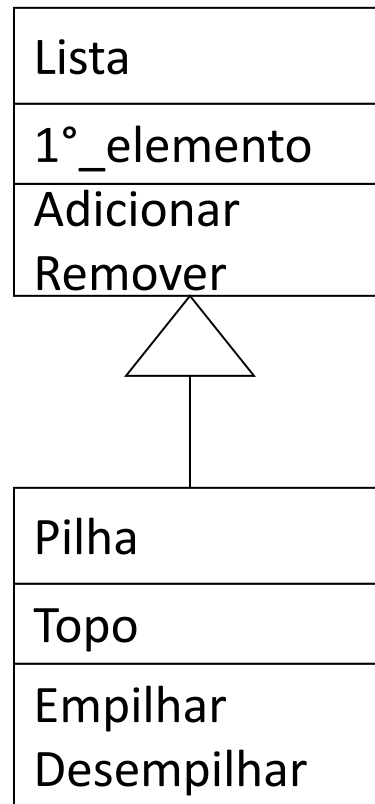
Material original elaborado pelo Prof. Dr.
Marcelo Yamaguti

- Estude para aprofundamento no conteúdo:
 - IEEE. **Guide to the Software Engineering Body of Knowledge**. SWEBOK. Version 3. IEEE Computer Society. 2014. – Chapter 3.
 - SOMMERVILLE, I. **Engenharia de software**. 9ª ed. São Paulo: Pearson Brasil, 2011. – Capítulo 16.
 - EZRAN, Michel. **Practical Software Reuse**. London: Springer, 2013. – Chapter 2.

- Reúso de classes:
 - Classes podem ser aproveitadas em outros contextos (sistemas).
 - Relacionamentos:
 - Composição/Delegação
 - Herança

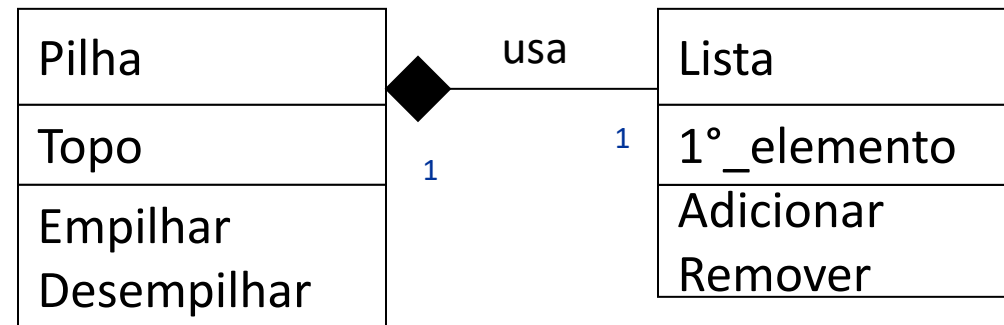


- Herança X delegação:



Com o uso de herança

Com o uso de delegação



- Biblioteca ou API (*Application Programming Interface*)
 - Implementa serviços usados por vários programas/aplicações
 - É uma forma comum de reuso
 - Pode disponibilizar várias funções comuns a sistemas diferentes :
 - Conversão de dados em outros formatos
 - Acesso a repositório de dados
 - Disponibilização de TADs
 - Cuidado: também é chamada de API as interfaces de comunicação com outras aplicações

- Bibliotecas
 - Implementam serviços que podem ser usados por programas
 - Disponibiliza funcionalidades comuns a diferentes tipos de sistemas

```
import java.util.Vector;

public class Customer {

    String name;
    Vector phoneNumbers = new Vector();

    void removePhoneNumber(String c) {
        phoneNumbers.removeElement(c);
    }

    void addPhoneNumber(String c) {
        phoneNumbers.addElement(c);
    }

    ...
}
```

- Boas características
 - Fácil de entender
 - Fácil de usar
 - Fácil de ler e manter
 - Fácil de estender
 - Evita o uso incorreto

- Cuidados:
 - *Single Responsibility Principle* – focar em uma única responsabilidade
 - Simplicidade para uso
 - Ser possível de estender, mas não ser modificado
 - Cuidar o encapsulamento de dados e implementação
 - Manter nomenclatura clara e consistente
 - Possuir documentação associada para o seu entendimento
 - Cuidar o desempenho / concorrência

- Cuidados:
 - Manutenção:
 - Uma biblioteca (API) é *software*, portanto evolui
 - Cuidado com versionamento
 - Cuidado com retrocompatibilidade
 - Permitir extensibilidade

