




Laboratório de Engenharia de Software

INF1636 – PROGRAMAÇÃO ORIENTADA A OBJETOS

Departamento de Informática – PUC-Rio

Ivan Mathias Filho
ivan@inf.puc-rio.br



Laboratório de Engenharia de Software


Programa – Capítulo 1

- Conceitos de Orientação a Objetos
 - Classe
 - Objeto
 - Mensagem
- Orientação a Objetos em Java
- Exercícios

© LES/PUC-Rio 2

Laboratório de Engenharia de Software

Programa – Capítulo 1




- **Conceitos de Orientação a Objetos**
 - **Classe**
 - **Objeto**
 - **Mensagem**
- Orientação a Objetos em Java
- Exercícios

© LES/PUC-Rio 3

Laboratório de Engenharia de Software

Extensibilidade em Dados



- Linguagens de programação atuais disponibilizam, além dos tipos primitivos, tais como **int**, **double**, **char**, **boolean** e etc, construtores de novos tipos.


```
typedef struct vetor
{
    double x,y;
} TVet;
```

- No exemplo acima, o tipo **TVet** foi criado para representar, a partir do tipo primitivo **double**, vetores no **R²**.

© LES/PUC-Rio 4

Laboratório de Engenharia de Software

Extensibilidade em Operações



- Além das operações básicas sobre os tipos primitivos, as linguagens de programação em geral permitem a construção de novas operações por meio de funções e procedimentos.


```
double prodInterno(TVet v1,TVet v2)
{
    return v1.x*v2.x+v1.y*v2.y;
}
```

- No exemplo acima, a função **prodInterno** permite calcular o produto interno de dois vetores do **R²**.

© LES/PUC-Rio
5

Laboratório de Engenharia de Software

Problema



- Linguagens como **C** e **Pascal** permitem operar livremente sobre os elementos básicos de tipos não-primitivos;
- Isso pode conduzir a situações em que o valor de uma variável seja incompatível com o seu tipo.

```
typedef struct data
{
    int dia,mes,ano;
} TData;

int main(void)
{
    TData dtNasc;

    dtNasc.mes=100; // Inconsistente!!!
}
```

© LES/PUC-Rio
6


Laboratório de Engenharia de Software

Solução

- A solução para esse problema seria um mecanismo sintático que não permitisse o livre acesso aos elementos básicos de tipos não-primitivos;
- Desse modo, a manipulação de dados baseados em tipos não-primitivos só poderia feita por meio de operações definidas especialmente para os novos tipos;
- Uma resposta para este problema é encontrada no conceito de Tipo Abstrato de Dados (ADT);
- Tal mecanismo é conhecido como **Classe**.

Uma classe é um tipo abstrato de dados possivelmente parcialmente implementado.

Bertrand Meyer
Object Oriented Software Construction



© LES/PUC-Rio
7

Laboratório de Engenharia de Software


Classe – Definição (1)

- Outra definição, mais operacional, de classe é:


Uma classe é a descrição de um conjunto de objetos que possuem a mesma semântica e compartilham as mesmas propriedades (atributos, operações e relacionamentos).


Grady Booch
UML User Guide

- Nesta definição uma classe é vista como um conjunto de objetos;
- Um objeto é uma instância de alguma classe.



© LES/PUC-Rio
8

Laboratório de Engenharia de Software	Classe – Definição (2)	
	<ul style="list-style-type: none"> • Uma classe, ou tipo classe, define uma estrutura que consiste em atributos, métodos e propriedades; • As instâncias de uma classe são chamadas de objetos; • Os atributos, métodos e propriedades de uma classe são chamados de seus membros; • Um atributo é essencialmente uma variável que faz parte de um objeto; • Tais como os campos de uma struct, os atributos de uma classe representam itens de dados que existem em cada uma de suas instâncias; • Um método é um procedimento ou função associada a uma classe; 	
	© LES/PUC-Rio	9

Laboratório de Engenharia de Software	Classe – Definição (3)	
	<ul style="list-style-type: none"> • A maioria dos métodos opera sobre as instâncias das classes; • Alguns métodos, entretanto, operam sobre as próprias classes; • São chamados métodos estáticos ou métodos de classe; • Uma propriedade é uma interface para os dados relativos a um objeto; • As propriedades têm especificadores de acesso, que determinam como os seus dados são lidos e modificados; • Métodos de outras classes veem as propriedades de certa classe como se fossem atributos. 	
	© LES/PUC-Rio	10

Laboratório de Engenharia de Software

Classe - Ilustração

Classe

Objetos

© LES/ PUC-Rio

11

Laboratório de Engenharia de Software

Objeto

- Um objeto é uma entidade com fronteiras bem definidas e que possui identidade, estado e comportamento próprios;
- O estado de um objeto é definido pelos valores dos seus atributos, e o seu comportamento pela suas operações.

Objeto Genérico

Objeto Bicicleta

© LES/ PUC-Rio

12

6

Objetos



- No modelo anterior, os atributos são **mantidos** no centro, ou núcleo, do objeto;
- As operações (métodos) rodeiam e escondem o núcleo do objeto dos demais objetos de um programa;
- O empacotamento da estrutura interna de um objeto é denominado **encapsulamento**;
- Esta concepção de um núcleo de atributos protegido pelas operações de um objeto é considerada por muitos como a representação **ideal** de um objeto de software.

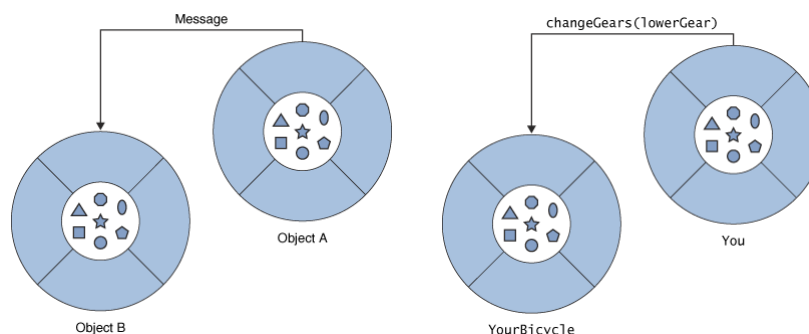
© LES/PUC-Rio

13

Mensagens




- Objetos interagem por meio de **mensagens**;
- Quando um objeto **A** deseja que um objeto **B** execute uma de suas operações, **A** envia uma mensagem para **B**;
- As informações passadas em uma mensagem são os **parâmetros** da operação a ser executada.



Laboratório de Engenharia de Software

Programa – Capítulo 1




- Conceitos de Orientação a Objetos
 - Classe
 - Objeto
 - Mensagem
- **Orientação a Objetos em Java**
- Exercícios

© LES/PUC-Rio

15

Laboratório de Engenharia de Software

Classe em Java – Exemplo



Class declaration

Variables

Constructor

Methods

```
import java.util.*;

public class Stack {
    private List<Object> items;

    public Stack() {
        items = new ArrayList<Object>();
    }

    public void push(Object item) {
        items.add(item);
    }

    public Object pop() {
        if (items.size() == 0)
            throw new EmptyStackException();
        return items.remove(items.size() - 1);
    }

    public boolean isEmpty() {
        return items.isEmpty();
    }
}
```

© LES/PUC-Rio

16

Classes em Java – Estrutura Básica



- Uma classe Java possui a seguinte estrutura básica:

```
<modificadores> class <nome> {
    // variáveis
    <lista_modificadores> <tipo> <nome> [=<valor_inicial>;]

    // métodos
    <lista_modificadores> <tipo> <nome>([<lista_parametros>])
    {
        <comandos>
    }
}
```

Classes em Java – Exemplo



```
public class Vetor {
    // variaveis
    private double x=3.0;
    private double y=4.0;

    // métodos
    public double modulo() {
        double a;

        a=Math.sqrt(Math.pow(x,2.0)+Math.pow(y,2.0));
        return a;
    }
}
```

Modificador de Acesso **public**



- Na definição de uma classe o modificador **public** declara que essa classe pode ser referenciada por qualquer outra classe. Sem **public**, uma classe pode ser referenciada apenas por classes que estejam no mesmo pacote em que ela está;
- Na definição de um atributo, o modificador **public** declara que esse atributo pode ser acessado por métodos externos à classe na qual ele foi definido.

```
public class Teste {
    private Vetor v;

    public void umMetodo()
    {
        v.x=5.0; // quebra do encapsulamento
                // indesejável!!!!
    }
}
```

© LES/PUC-Rio

19

Modificador de Acesso **private**



- Para não permitir a quebra do encapsulamento, deve-se usar o modificador de acesso **private**;
- Um atributo privado só pode ser acessado pelos métodos definidos na própria classe.

```
public class Teste {
    private Vetor v;

    public void umMetodo()
    {
        v.x=5.0; // quebra do encapsulamento
                // indesejável!!!!
    }
}
```

The field Vetor.x is not visible
Press F2 for focus.


- Os métodos de uma classe são normalmente definidos com o modificador de acesso **public**.

© LES/PUC-Rio

20

Laboratório de Engenharia de Software

Programa – Capítulo 1



- Conceitos de Orientação a Objetos
 - Classe
 - Objeto
 - Mensagem
- Orientação a Objetos em Java
- **Exercícios**

© LES/PUC-Rio21