

MAC0321

Laboratório de Programação Orientada a Objetos

Denis Mauá
IME-USP

Baseado fortemente no material didático do Prof. Fabio Kon





Aula 2

Introdução à Programação Orientada a Objetos

Nos primórdios era o caos...

- Linguagem de máquina
- Linguagem de montagem (ASSEMBLY)
- Linguagem de alto nível (FORTRAN)
- Programação estruturada (ALGOL)
 - Muita repetição de código
 - Código mal organizado
 - Código de difícil entendimento
 - Muitos erros (*bugs*)

Como resolver esses problemas?

- Programação Orientada a Objetos
 - 1967 a 1980
- Simula
 - 1962 a 1970
- Smalltalk
 - 1972 a 1980
- C++ (1990)
- Java (1995)
- Ruby
- Python

Objeto

Algo que **encapsula**

– dados

- variáveis
- atributos
- propriedades

– código

- funções
- métodos

Classe

- Define os elementos de um conjunto de objetos
 - quais os métodos
 - quais os atributos
- Funciona como uma fábrica de objetos

Classe e Objeto

- Classe
 - Fruta
 - Cor, forma, peso, preço
- Objetos
 - Maçã
 - Cor, forma, peso, preço
 - Laranja
 - Cor, forma, peso, preço

Exemplo (Java)

```
// Bola.java
class Bola {
    double diametro;
    String cor;
    public Bola(double d, String c) {
        diametro = d; cor = c;
    }
    public void descrever() {
        System.out.println("Diametro: " + diametro);
        System.out.println("Cor: " + cor);
    }
}
```


Exemplo (Python)

```
# Bola.py  
class Bola:  
    def __init__(self, diametro, cor):  
        self.diametro = diametro  
        self.cor = cor  
    def descrever(self):  
        print(f"Diametro: {self.diametro}")  
        print(f"Cor: {self.cor}")
```

Instanciação de objetos

- Criamos objetos de um classe por meio da instanciação
- Exemplos de sintaxes:
 - Java/C#: `Carro c = new Carro();`
 - C++: `Carro *c = new Carro();`
 - Python: `c = Carro()`
 - Ruby: `c = Carro.new`
 - Smalltalk: `c ← Carro new`

Exemplo (Java)

```
// Main.java  
public class Main {  
    public static void main(String[] args) {  
        Bola bola1 = new Bola(12, "amarelo");  
        Bola bola2 = new Bola(10, "roxo");  
        bola1.descrever();  
        bola2.descrever();  
    }  
}
```

Exemplo (Python)

```
# Main.py
from bola import Bola
if __name__ == "__main__":
    bola1 = new Bola(12, "amarelo")
    bola2 = new Bola(9, "roxo")
    bola1.descrever()
    bola2.descrever()
```

Comunicação entre objetos

- Envio de mensagem
 - Chamada de método
 - Chamada de função
- como se comporta um sistema de objetos?

Abstração

- Objetos representam
 - coisas do mundo real
 - do mundo virtual ou computacional
- Aumentam o nível de abstração da programação

Variáveis de classe

- Informação/dado que é compartilhado entre todos objetos instanciados de uma classe
 - Exemplo: número de instâncias, tempo desde que última instância foi criada, rótulo da classe (nome amigável para exibição)
- Variáveis de classe são armazenadas em memória compartilhada
 - Melhor uso de memória

Exemplo (Java)

```
Class Bola {  
    static int no_bolas = 0;  
    public Bola() {  
        no_bolas += 1;  
    }  
    public static void main(String[] args) {  
        Bola b1 = new Bola();  
        Bola b2 = new Bola();  
        System.out.println("Qtd. Bolas: " + Bola.no_bolas);  
    }  
}
```


Exemplo (Java)

```
Class Bola {  
    static int no_bolas = 0;  
    public Bola() {  
        no_bolas += 1;  
    }  
    public static void main(String[] args) {  
        Bola b1 = new Bola();  
        Bola b2 = new Bola();  
        System.out.println("Qtd. Bolas: " + b1.no_bolas);  
    }  
}
```

Exemplo (Python)

```
class Bola:
    no_bolas = 0;
    def __init__(self):
        no_bolas += 1;

def main():
    b1 = new Bola()
    b2 = new Bola()
    print(f"Qtd. Bolas: {Bola.no_bolas}")
```

Exemplo (Python)

```
class Bola:
    no_bolas = 0;
    def __init__(self):
        no_bolas += 1;

def main():
    b1 = new Bola()
    b2 = new Bola()
    print(f"Qtd. Bolas: {b1.no_bolas}")
    # Nota: usar b1.no_bolas = valor torna
    variável não estático
```

Métodos de classe

- Podemos também ter métodos de classe, que acessam apenas variáveis de classe
 - Exemplo: devolver número de instâncias da classe, exibir informações sobre a classe

Recapitulando...

- Objetos encapsulam dados e código
- Classes descrevem tipos de objetos e funcionam como fábricas
- Sistemas orientados a objetos são compostos por vários objetos que trocam mensagens entre si
- Um programa OO usa objetos para representar os elementos do seu domínio

Boas práticas de programação

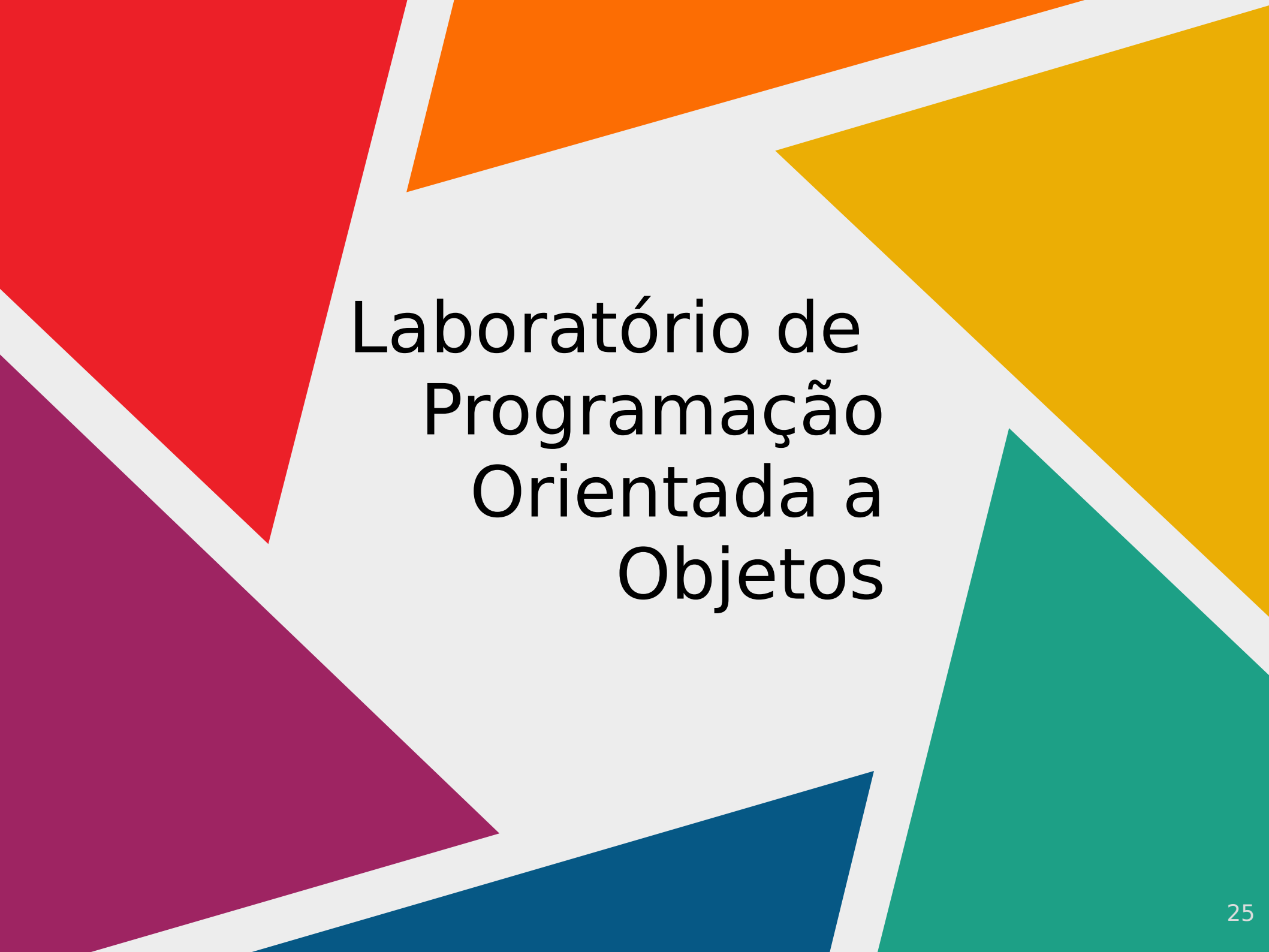
- Escolha de bons nomes para
 - variáveis
 - atributos/propriedades
 - métodos e seus argumentos
 - classes
- *Intention-revealing names*
- Classes geralmente são substantivos
- Métodos geralmente são verbos

Princípio da Responsabilidade Única

- Em um sistema orientado a objetos, as classes devem possuir uma única responsabilidade.
- Ou seja, uma classe não deve tratar de assuntos diferentes.
- Se uma classe trata de 2 ou 3 assuntos, veja se não dá para quebrá-la em 2 ou 3 classes menores.
 - conta corrente vs. pessoa vs. endereço

Conceitos Fundamentais de OO

- Objetos
- Classes
- Abstração
- variáveis de instância e de classe
- Troca de mensagens/chamada de método
- Nomes que revelam intenção
- Princípio da Responsabilidade Única
- *Herança*
- *Polimorfismo*



Laboratório de Programação Orientada a Objetos



IME-USP

