



UNIRIO

Orientação a objetos

Aula 3

BSI — 2018.2

Jefferson Elbert Simões

CCET/DIA

28 de agosto de 2018

O que fizemos até agora?

O que fizemos até agora?

- Fizemos programas

O que fizemos até agora?

- Fizemos programas
 - ▶ Descrevemos algoritmos passo-a-passo
 - ▶ Armazenamos dados em variáveis e arrays
 - ▶ Destacamos funções
 - ▶ Trechos autocontidos que realizam tarefas específicas
 - ▶ Encontramos problemas dentro de problemas maiores
 - ▶ e aplicamos recursão

O que fizemos até agora?

- Fizemos programas
 - ▶ Descrevemos algoritmos passo-a-passo
 - ▶ Armazenamos dados em variáveis e arrays
 - ▶ Destacamos funções
 - ▶ Trechos autocontidos que realizam tarefas específicas
 - ▶ Encontramos problemas dentro de problemas maiores
 - ▶ e aplicamos recursão
- Tudo que fizemos até agora se encaixa no paradigma de **programação procedural**
 - ▶ O "centro do universo" são os procedimentos (funções)

Por que mudar?

- Em time que está ganhando...
- Para programar proceduralmente, é preciso:
 - ▶ Acompanhar o fluxo do programa
 - ▶ Identificar a semântica de todas as variáveis em todos os possíveis estados do programa
 - ▶ Garantir que "está tudo sob controle" (mesmo quando vários procedimentos operam sobre os mesmos dados)
- Tudo isto é extremamente difícil de realizar quando projetos crescem

Como podemos ajudar?

- Certas "metodologias de programação" podem ajudar
 - ▶ Modularização de código
 - ▶ Separação do código em diferentes módulos lidando com diferentes aspectos da funcionalidade do software
 - ▶ Clarifica as responsabilidades de diferentes trechos de código
 - ▶ Ocultamento de informação
 - ▶ Imposição de restrições de acesso entre trechos de código
 - ▶ Evita que tais responsabilidades sejam ignoradas acidentalmente
 - ▶ **Objetivo:** permitir que o código tenha semântica, aos olhos de nós, desenvolvedores

Orientação a objetos

- Novo? paradigma de programação
 - ▶ Proposto nos anos 60, dominante nos anos 90
- **Ideia fundamental**: não pensar primeiro nos procedimentos, mas nos dados, no que eles representam e como eles se relacionam
- Estrutura do código projetada para naturalmente conferir semântica aos seus elementos
 - ▶ Força boas metodologias de programação, inclusive na fase de planejamento do desenvolvimento do software
 - ▶ Ajuda a organizar o raciocínio durante o processo de desenvolvimento

Orientação a objetos

Como o ser humano raciocina?

- Identificando conceitos (concretos ou abstratos) através de ocorrências
- Compreendendo suas propriedades
- Compreendendo seus comportamentos
- Reunindo ocorrências semelhantes em grupos

Orientação a objetos

Como o ser humano raciocina?

- Identificando conceitos (concretos ou abstratos) através de **objetos**
- Compreendendo suas propriedades
- Compreendendo seus comportamentos
- Reunindo **objetos** semelhantes em grupos

Orientação a objetos

Como o ser humano raciocina?

- Identificando conceitos (concretos ou abstratos) através de **objetos**
- Compreendendo suas propriedades
- Compreendendo seus comportamentos
- Reunindo **objetos** semelhantes em **classes**

Orientação a objetos

Como o ser humano raciocina?

- Identificando conceitos (concretos ou abstratos) através de **objetos**
- Compreendendo seus **atributos**
- Compreendendo seus comportamentos
- Reunindo **objetos** semelhantes em **classes**

Orientação a objetos

Como o ser humano raciocina?

- Identificando conceitos (concretos ou abstratos) através de **objetos**
- Compreendendo seus **atributos**
- Compreendendo seus **métodos**
- Reunindo **objetos** semelhantes em **classes**

Objetos e classes

Objeto: representação de um elemento (concreto ou abstrato) do mundo real. Ex:

- Os computadores desta sala
- Os professores da UNIRIO
- As músicas que vocês ouviram hoje

Objetos e classes

Objeto: representação de um elemento (concreto ou abstrato) do mundo real. Ex:

- Os computadores desta sala
- Os professores da UNIRIO
- As músicas que vocês ouviram hoje

Classe: tipo ao qual este objeto pertence. Ex:

- Computador
- Professor
- Música

* **Classes** são fôrmas, moldes para **Objetos**

Objetos e classes

Exemplo: modelando esta turma

```
public class Pessoa {  
}  
... // em outro ponto do código (ex: função Main)  
Pessoa professor = new Pessoa();  
Pessoa monitor = new Pessoa();  
Pessoa aluno = new Pessoa();
```

O que esse código faz?

Objetos e classes

Exemplo: modelando esta turma

```
public class Pessoa {  
}  
... // em outro ponto do código (ex: função Main)  
Pessoa professor = new Pessoa();  
Pessoa monitor = new Pessoa();  
Pessoa aluno = new Pessoa();
```

O que esse código faz? Tecnicamente não faz nada!

- mas começa a representar nosso mundo de interesse

Objetos e classes

Exemplo: modelando esta turma

```
public class Pessoa {  
}  
... // em outro ponto do código (ex: função Main)  
Pessoa professor = new Pessoa();  
Pessoa monitor = new Pessoa();  
Pessoa aluno = new Pessoa();
```

O que esse código faz? Tecnicamente não faz nada!

- mas começa a representar nosso mundo de interesse (declaração de uma classe, instanciação de 3 objetos)

Atributos

- Objetos podem possuir atributos
 - ▶ Representam propriedades, características
 - ▶ Variáveis "locais" — cada objeto possui as suas
- Classes definem atributos dos objetos

```
public class Pessoa {  
    public int altura; // em cm  
}  
  
... // em algum outro ponto do código  
Pessoa professor = new Pessoa();  
Pessoa aluno = new Pessoa();  
professor.altura = 172;  
aluno.altura = 210;
```

Métodos

- Objetos podem possuir métodos
 - ▶ Representam comportamentos, atividades
 - ▶ Funções "locais" — cada objeto possui as suas
- Classes definem atributos dos objetos

```
public class Pessoa {
    public int altura; // em cm
    public void fazerPilates () {
        altura += 3;
    }
}

... // em algum outro ponto do código
Pessoa professor = new Pessoa();
Pessoa aluno = new Pessoa();
professor.altura = 172;
aluno.altura = 210;
professor.fazerPilates();
System.out.println( professor.altura ); // 175
System.out.println( aluno.altura ); // 210
```

Objetos e classes

- Objetos são instâncias de classes
 - ▶ Restritos à descrição que a classe determina para eles — através de atributos e métodos
 - ▶ Classes (tipos) definem o que objetos (variáveis) podem ser
- Classes são os blocos de construção mais fundamental de orientação a objetos
- A escolha de atributos e métodos faz parte do processo de modelagem

"All models are wrong. Some models are useful."
(George E. P. Box)

Exercícios

Projete classes para representar os seguintes conceitos e procedimentos relacionados:

1. **Números racionais**: impressão em tela (fração e decimal), incremento e complementar
2. **Vetores em \mathbb{R}^3** : cálculo de normas de Manhattan, Euclídeana e de Chebyshev
 - Bônus: generalize para \mathbb{R}^n (para qualquer n)
3. **Hora (sem data)**: incrementos e decrementos (em segundos), impressão em tela em formato curto e longo
4. **Lista de dados, ponto flutuante**: adição de valores, busca por valor mínimo e máximo, cálculo de média e desvio padrão

Exercícios

Mais detalhes:

2. Se $(x, y, z), (x', y', z')$ são vetores em \mathbb{R}^3 :

Norma de Manhattan $\|(x, y, z)\|_1 = |x| + |y| + |z|$

Norma Euclideana $\|(x, y, z)\|_2 = \sqrt{x^2 + y^2 + z^2}$

Norma de Chebyshev $\|(x, y, z)\|_\infty = \max\{|x|, |y|, |z|\}$

Exercícios

Projete classes para representar os seguintes conceitos. Desta vez vocês irão decidir quais atributos e métodos vocês vão modelar =>

5. Carro
6. Elevador
7. Jogo da velha
8. Conta Bancária

Créditos

- Baseado em slides de Gleison Santos e Sérgio Villas-Boas