
A yellow L-shaped line is positioned in the top-left corner of the slide.

# CC3642

# Orientação a Objetos

A short blue horizontal line is centered below the title.

Prof. Danilo H. Perico

A yellow L-shaped line is positioned in the bottom-right corner of the slide.

# Construtores e UML

---

# Exercício 1 - Classe Data aprimorada

(Deitel 8.16) Crie uma classe *Data* com as seguintes capacidades:

- A. Gerar saída em múltiplos formatos:
  - i. MM/DD/YYYY
  - ii. Março 02, 2019
  - iii. DDD YYYY
- B. Utilizar construtores sobrecarregados para criar objetos *Data* inicializados com datas nos formatos da parte (A). No primeiro caso, o construtor deve receber 3 valores inteiros. No segundo caso deve receber uma String e dois valores inteiros. No terceiro caso deve receber dois valores inteiros, o primeiro sendo o número de dias no ano. [Dica: para converter a representação de string do mês em valor numérico, compare as strings utilizando o método *equals*. Por exemplo, se *s1* e *s2* forem strings, a chamada de método *s1.equals(s2)* retornará true se as strings forem idênticas.

# Exercício 2 - Classe Racional

(Deitel 8.17) Crie uma classe chamada *Racional* para realizar aritmética com frações. Utilize variáveis do tipo inteiro para representar as variáveis de instância *private* da classe: o *numerador* e o *denominador*. Forneça um construtor que permita que um objeto dessa classe seja inicializado quando ele for declarado. O construtor deve armazenar a fração em uma forma reduzida, por exemplo, a fração  $2/4$  é equivalente a  $1/2$  e seria armazenada no objeto como 1 no numerador e 2 no denominador. Forneça um construtor sem argumento com valores padrão caso nenhum inicializador seja fornecido. Forneça métodos *public* que realizam cada uma das operações a seguir:

- a) Somar dois números *Racional*: o resultado da adição deve ser armazenado na forma reduzida.
- b) Subtrair dois números *Racional*: o resultado da subtração deve ser armazenado na forma reduzida.

## Exercício 2 - Classe Racional

- c) Multiplicar dois números *Racional*: o resultado da multiplicação deve ser armazenado na forma reduzida.
- d) Dividir dois números *Racional*: o resultado da divisão deve ser armazenado na forma reduzida.
- e) Retornar uma representação *String* de um número *Racional* na forma a/b, onde a é o numerador e b é o denominador.
- f) Retornar uma representação *String* de um número *Racional* no formato de ponto flutuante. (Considere a possibilidade de fornecer capacidades de formatação que permitam que o usuário da classe)

Escreva um programa para testar sua classe.

## Exercício 3 - Classe HeartRates

3.16 (Calculadora da frequência cardíaca) Durante o exercício, você pode usar um monitor de frequência cardíaca para verificar se sua frequência cardíaca permanece dentro de um intervalo seguro sugerido por seus treinadores e médicos. A fórmula para calcular sua frequência cardíaca máxima em batimentos por minuto é 220 menos a sua idade em anos. A sua frequência cardíaca alvo é um intervalo que representa 50 a 85% da sua frequência cardíaca máxima. Crie uma classe chamada *HeartRates*. Os atributos da classe devem incluir o nome, sobrenome e data de nascimento da pessoa (que consistem em atributos separados para o mês, dia e ano de nascimento). Sua classe deve ter um construtor que receba esses dados como parâmetros. Para cada atributo, forneça os métodos set e get. A classe também deve incluir um método que calcula e retorna a idade da pessoa (em anos), um método que calcula e retorna a frequência cardíaca máxima da pessoa e um método que calcula e retorna a frequência cardíaca alvo da pessoa.

# Exercício 3 - Classe HeartRates

Escreva um aplicativo Java que solicite as informações da pessoa, instancia um objeto da classe HeartRates e imprima as informações desse objeto - incluindo o nome, sobrenome e data de nascimento da pessoa - e calcule e imprima a idade da pessoa em (anos), frequência cardíaca máxima e faixa de frequência cardíaca alvo.

# Exercício 4 - UML

Construa a relação entre as classes / objetos a seguir. Escolha fazer com Composição ou Agregação. Implemente e faça o teste no main. Deixe claro nos comentários do código qual foi a sua escolha (composição ou agregação) e o motivo da escolha.

