

Lista de Exercícios 2

Disciplina: Projeto de Sistemas Orientados a Objetos

Aula: Orientação a Objetos

Conteúdo Formativo: Conceitos do paradigma da programação orientada a objetos, Diagrama de Classes (UML), classes e objetos, atributos e métodos; Agregação, composição, herança e polimorfismo

Exercício 1: Crie uma classe Contador, que encapsule um valor usado para contagem de itens ou eventos. A classe deve oferecer métodos que devem:

- a) Zerar;
- b) Incrementar;
- c) Retornar o valor do contador.

Exercício 2: Faça um programa (F.U.P) que instancie a classe contador do exercício anterior, criando um objeto novoContador do tipo Contador. Depois o programa deve realizar as seguintes ações:

- 1) Use um loop para incrementar o contador até 10.
- 2) Zere o valor do contador.
- 3) Use um loop para incrementar o contador até 100.
- 4) Zere o valor do contador.
- 5) Use um loop para incrementar o contador até 1000.
- 6) Zere o valor do contador.

Exercício 3: Crie uma classe Retangulo que obedeça à descrição abaixo:

Retangulo
- lado1: double - lado2: double - area: double - perimetro: double
+ Retangulo() + Retangulo(lado1: double, lado2: double) + calcularArea(): void + calcularPerimetro(): void

- A classe possui os atributos lado1, lado2, area e perimetro, todos do tipo float.
- O método calcularArea deve realizar o cálculo da área do retângulo ($\text{area} = \text{lado1} * \text{lado2}$). Em seguida, deve escrever o valor da área na tela.
- O método calcularPerimetro faz o cálculo do perimetro ($\text{perimetro} = 2 * \text{lado1} + 2 * \text{lado2}$). Em seguida, deve escrever o valor do perímetro na tela.

Exercício 4: F.U.P que instancie a classe Retangulo do exercício anterior, criando um objeto novoRetangulo do tipo Retangulo. Depois o programa deve realizar as seguintes ações:

- 7) Atribua o valor 10 ao atributo lado1.
- 8) Atribua o valor 5 ao atributo lado2.
- 9) Chame o método calcularArea.

- 10) Chame o método calcularPerimetro.
- 11) Atribua o valor 7 ao atributo lado2.
- 12) Chame o método calcularArea.
- 13) Chame o método calcularPerimetro.

Exercício 5: Crie uma classe Circulo que obedeça à descrição abaixo:

Circulo
- raio: double - area: double - perimetro: double
+ Circulo() + Circulo(raio: double) + calcularArea(): void + calcularPerimetro(): void

- A classe possui os atributos raio, area e perímetro, todos do tipo float.
- O método calcularArea deve realizar o cálculo da área do retângulo ($\text{area} = \text{raio} * \text{raio} * 3.14$). Em seguida, deve escreve o valor da area na tela.
- O método calcularPerimetro faz o cálculo do perimetro ($\text{perimetro} = 2 * 3.14 * \text{raio}$). Em seguida, deve escreve o valor do perímetro na tela.

Exercício 6: F.U.P que instancie a classe Circulo do exercício anterior, criando um objeto novoCirculo do tipo Circulo. Depois o programa deve realizar as seguintes ações:

- 1) Atribua o valor 10 ao atributo raio.
- 2) Chame o método calcularArea.
- 3) Escreva na tela o valor da área.
- 4) Chame o método calcularPerimetro.
- 5) Escreva na tela o valor do perímetro.
- 6) Atribua o valor 4 ao atributo raio.
- 7) Chame o método calcularArea.
- 8) Escreva na tela o valor da área.
- 9) Chame o método calcularPerimetro.
- 10) Escreva na tela o valor do perímetro.

Exercício 7: Crie uma classe Funcionario que obedeça à descrição abaixo:

Funcionario
- nome: String - sobrenome: String # horasTrabalhadas: Integer # valorPorHora: double
+ Funcionario() + Funcionario(nome: String, sobrenome: String) + nomeCompleto(): void + calcularSalario(): void + incrementarHoras(horas: Integer): void

- A classe possui os atributos nome, sobrenome, horasTrabalhadas e valorPorHora.
- O método nomeCompleto deve escrever na tela o atributo nome concatenado ao atributo sobrenome.

- O método `calcularSalario` faz o cálculo de quanto o funcionário irá receber no mês, multiplicando o atributo `horasTrabalhadas` pelo atributo `valorPorHora`. Em seguida, escreve o valor na tela.
- O método `incrementarHoras` adiciona um valor passado por parâmetro ao valor já existente no atributo `valorPorHora`.

Exercício 8: F.U.P que instancie a classe `Funcionário` do exercício anterior, criando um objeto `novoFuncionario` do tipo `Funcionario`. Depois o programa deve realizar as seguintes ações:

- 1) Atribua o valor “Luis” ao atributo `nome`.
- 2) Atribua o valor “Silva” ao atributo `sobrenome`.
- 3) Atribua o valor 10 ao atributo `horasTrabalhadas`
- 4) Atribua o valor 25.50 ao atributo `valorPorHora`.
- 5) Chame o método `nomeCompleto`.
- 6) Chame o método `calcularSalario`.
- 7) Adicione 8 ao atributo `horasTrabalhadas` utilizando o método
- 8) `incrementarHoras`.
- 9) Chame novamente o método `calcularSalario`.

Exercício 9: Construa a classe `Livro` em Java, que obedeça à descrição abaixo:

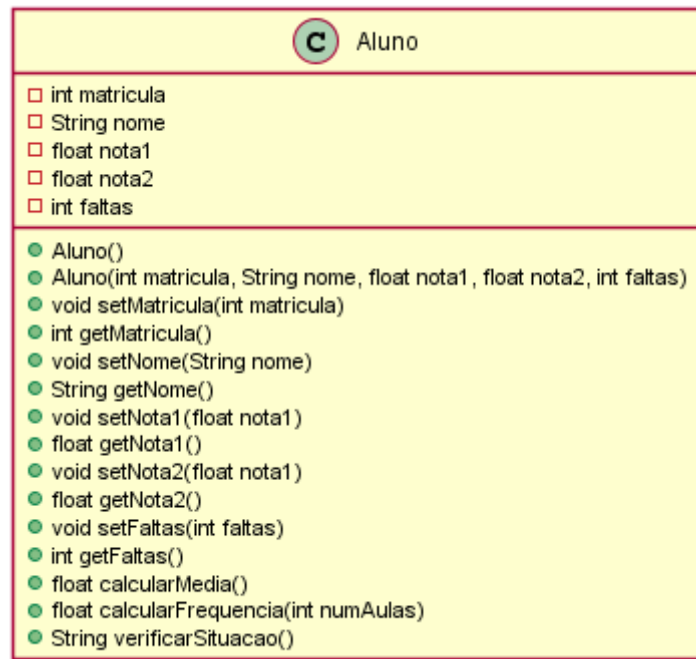
Livro
- titulo: String
- qtdPaginas: Integer
- paginasLidas: Integer
+ Livro()
+ Livro(nome: String, qtdPaginas: Integer)
+ verificarProgresso(): void

- A classe possui os atributos `titulo`, `qtdPaginas` e `paginasLidas`. Esses atributos
- devem ser marcados com o modificador de acesso `private`.
- Crie os métodos `get` e `set` para cada um dos atributos.
- Crie ainda o método `verificarProgresso` que deverá calcular a porcentagem de leitura do livro até o momento. Para isso, deverá usar os valores dos atributos `qtdPaginas` e `paginasLidas`, através da fórmula: $\text{porcentagem} = \text{paginasLidas} * 100 / \text{qtdPaginas}$. O valor da porcentagem deverá ser mostrado na tela conforme a mensagem “Você já leu X por cento do livro”, onde o valor de X é o valor calculado pela fórmula apresentada anteriormente.

Exercício 10: F.U.P que instancia a classe `Livro` do exercício anterior, criando um objeto `livrofavorito` do tipo `Livro`. Depois o programa deve realizar as seguintes ações:

- 1) Altere o atributo `titulo` para “O Pequeno Príncipe”. Utilize, para isso, o método `setTitulo`.
- 2) Altere o atributo `qtdPaginas` para 98. Utilize, para isso, o método `setQtdPaginas`.
- 3) Escreva na tela a mensagem: “O livro X possui Y páginas”, onde no lugar de X deverá aparecer o valor do atributo `titulo` e, no lugar de Y deverá aparecer o valor do atributo `qtdPaginas`. Utilize, para tanto, os métodos `getTitulo` e `getQtdPaginas`.
- 4) Altere a quantidade de `paginasLidas` para 20.
- 5) Chame o método `verificarProgresso`.
- 6) Altere a quantidade de `paginasLidas` para 50.
- 7) Chame o método `verificarProgresso`.

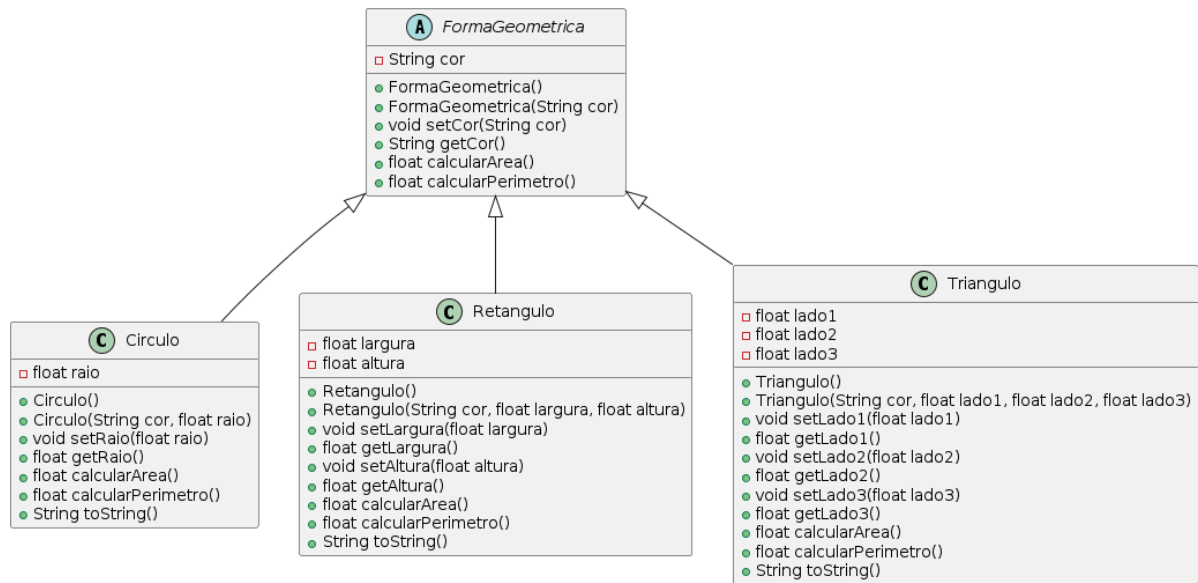
Exercício 11: Crie uma classe Aluno em Java, que obedeça às seguintes especificações:



- Um aluno possui Matricula, Nome, Faltas, Nota1 e Nota2
- A classe deve possuir o construtor padrão e a sobrecarga com os atributos Matricula, Nome, Nota1 e Nota2
- Deve haver os setters e getters para os atributos Matricula, Nome, Nota1 e Nota2
- A classe deve possuir uma função para calcular a média através da fórmula: $media = (nota1 + nota2) / 2$.
- A classe deve possuir uma função para calcular a frequência através da fórmula: $frequencia = faltas / numeroAulas$. O parâmetro `numeroAulas` é recebido como parâmetro da função.
- A classe deve possuir uma função para retornar a situação do aluno conforme as regras a seguir:
 - Média $\geq 7,0$ E Frequência $\geq 75\%$ = APROVADO
 - $4,0 < Média < 7,0$ E Frequência $\geq 75\%$ = RECUPERAÇÃO
 - Média $< 4,0$ OU Frequência $< 75\%$ = REPROVADO

Exercício 12: F.U.P que crie 10 instâncias da classe Aluno. Para cada instância, o programa solicita via componente Scanner as informações de cada aluno. No final da leitura, o programa deve informar a média da turma, a quantidade de aprovados, alunos em recuperação e reprovados. O programa também informa o nome do aluno que tem a maior média.

Exercício 13: Crie as classes FormaGeometrica (abstrata), Circulo, Retângulo e Triangulo conforme o diagrama de classes (UML) abaixo:



- A classe **FormaGeometrica** é abstrata e contém a propriedade `cor`
- As classes **Circulo**, **Retangulo** e **Triangulo** são concretas e herdeiras da classe **FormaGeometrica**
- As classes filhas devem sobrescrever os métodos `calcularArea()` e `calcularPerimetro()` conforme abaixo:
 - Perímetro e área do círculo: $P = 2 \cdot \pi \cdot \text{raio}$; $A = \pi \cdot \text{raio} \cdot \text{raio}$
 - Perímetro e área do retângulo: $P = 2 \cdot \text{largura} + 2 \cdot \text{altura}$; $A = \text{largura} \cdot \text{altura}$
 - Perímetro e área do triângulo: $P = \text{lado1} + \text{lado2} + \text{lado3}$; $A =$ use a fórmula Heron (<https://mundoeducacao.uol.com.br/matematica/formula-heron.htm>)
- As classes filhas devem sobrescrever o método `toString()` da seguinte forma:
 - Círculo: "Circulo de cor <cor> e raio <raio>"
 - Retângulo: "Retângulo de cor <cor>, largura <largura> e altura <altura>"
 - Triângulo: "Triângulo de cor <cor>, lado1 <lado1>, lado2 <lado2> e lado3 <lado3>"

Exercício 14: F.U.P que instancie as classes do exercício anterior da seguinte maneira:

- Crie uma instancia da classe **Circulo** com a cor "vermelho", raio = 2 e guarde em uma variável chamada **instanciaCirculo**
- Crie uma instancia da classe **Retangulo** com a cor "verde", largura = 5, altura = 3 e guarde em uma variável chamada **instanciaRetangulo**
- Crie uma instancia da classe **Triangulo** com a cor "azul", lado1 = 2, lado2 = 5, lado3 = 7 e guarde em uma variável chamada **instanciaTriangulo**
- Guarde as formas em um vetor do tipo **FormaGeometrica[]**
- Use um loop para:
 - Mostrar as formas usando o método `toString()`
 - Calcular o perímetro total (soma de todos os perímetros das formas)
 - Calcular a área total (soma de todas as áreas das formas)