



SISTEMAS DE INFORMAÇÃO  
UEG - CCET



# ATIVIDADE PRÁTICA EM PROGRAMAÇÃO I – ATIVIDADE PARCIAL -1ºVA

Esta atividade é um componente de primeira VA. Estimula e avalia o desenvolvimento dos conhecimentos envolvidos na disciplina Programação I. O valor dessa atividade é 6,0 (seis) pontos.

Prof. Márcio Giovane Cunha Fernandes, Me.

Goiânia, junho de 2021.



## SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>1</b>
<b>2. REGRAS DA ATIVIDADE PRÁTICA.....</b>	<b>1</b>
2.1. DESENVOLVIMENTO DO TRABALHO .....	1
2.2. REQUISITOS FUNCIONAIS .....	1
2.3. REQUISITOS NÃO-FUNCIONAIS .....	1
2.4. REQUISITOS TÉCNICOS.....	2
<b>3. FORMA DE AVALIAÇÃO .....</b>	<b>2</b>
<b>4. AJUSTES DE REALIZAÇÃO .....</b>	<b>2</b>
<b>5. DATAS .....</b>	<b>2</b>
<b>ANEXO A – EXERCÍCIOS PRÁTICA JAVA .....</b>	<b>3</b>
<b>ANEXO B – EXERCÍCIO PRÁTICA POO E JAVA .....</b>	<b>4</b>

# 1. INTRODUÇÃO

O trabalho introduzido nesse documento busca a realização de prática em programação I e catalisar a aplicação dos conceitos de programação de computadores, da linguagem de programação Java e do paradigma de programação orientado a objeto (POO).

O processo de aprendizado exige, além da literatura e da exposição de conteúdo, uma prática empírica cuja dimensão não pode ser abordada apenas durante a realização da disciplina.

Enfrentar-se-á essa realidade na referida disciplina através da pesquisa sobre a estrutura programática do Java e da realização de exercícios. Para tal, é proposta a realização dos exercícios dirigidos ao desenvolvimento do conhecimento sobre a ferramenta e dos conceitos do paradigma de programação orientado a objeto.

Na realização dessa atividade encerrar-se-á os conceitos mínimos necessários à obtenção da nota parcial para a 1ª VA na disciplina de Programação I, assunto que esse documento propõe-se a esclarecer e definir.

## 2. REGRAS DA ATIVIDADE PRÁTICA

### 2.1. DESENVOLVIMENTO DO TRABALHO

1. Realizar os exercícios para a prática da linguagem Java e de traços da programação orientada a objeto (POO).
2. Demonstrar a capacidade de implementar programas pequenos utilizando conceitos introdutórios de POO e Java.

### 2.2. REQUISITOS FUNCIONAIS

1. Implementar todos os exercícios indicados.
2. Saber implementar **qualquer** exercício da lista ao ser solicitado pelo professor.

### 2.3. REQUISITOS NÃO-FUNCIONAIS

1. Todos os alunos devem ser capazes de implementar qualquer exercício dos ANEXOS A e B em 10 minutos utilizando os conceitos e termos introdutórios da POO e todos os conceitos da linguagem de programação Java utilizados para solucionar os exercícios.

## 2.4. REQUISITOS TÉCNICOS

1. Uso da linguagem Java.
2. Uso, **quando direcionado**, dos elementos da POO (classe, atributo/campo, métodos, modificadores de acesso, métodos sets e gets, construtores etc.).
3. Paradigma de programação orientada a objeto.
4. **Compilar e executar os programas produzidos nos exercícios utilizando o prompt (linhas de comandos para Windows) ou terminal de comandos (linhas de comandos para Linux).**

## 3. FORMA DE AVALIAÇÃO

1. Formação de grupos de **exatos** três alunos.
2. **A entrega da implementação dos exercícios possui valor 3,0 pontos.**
3. **Um dos alunos do grupo será escolhido para representar o grupo e a nota auferida será atribuída a todos os integrantes.**
4. O aluno implementará um exercício escolhido dentre os ANEXOS A e B **em 10 minutos**. O valor dessa implementação é 3,0 pontos.
5. Ao apoio do aluno, poder-se-á consultar a documentação online do java (api documents – sugiro ao aluno realizar o *download* da documentação e tê-la disponível).
6. A nota completa (3,0 pontos) será obtida apenas com a **percepção da execução correta** do programa. De outra forma, a avaliação do código sem o programa executar ou com execução incorreta/incompleta será avaliado com a nota total reduzida para 1,5 pontos.
7. As apresentações acontecerão em ordem de sorteio dos grupos. Iniciarão no horário da aula, seguido o intervalo de 10 em 10 minutos para a troca de grupo;
8. Por dia serão avaliados 5 grupos (30 alunos em 2 dias);
9. O grupo que estiver aguardando o momento de apresentar deverá ser célere para iniciar a avaliação ao ser convidado. Haverá uma sala virtual específica para cada grupo.
10. Questões não arroladas nesse documento serão resolvidas em debate entre os reclamantes e o professor.

## 4. AJUSTES DE REALIZAÇÃO

Houve ajustes à realização dessa atividade avaliativa provocada pelos discentes, a saber:

1. Entrega dos exercícios no Moodle no dia 23/06/2021 até às 17 horas.
2. No anexo B, entrega apenas dos exercícios 1 e 2.

## 5. DATAS

- Apresentações dias 23 e 25 de junho de 2021.

## ANEXO A – EXERCÍCIOS PRÁTICA JAVA

- 1) Escreva um programa em Java que peça ao usuário que digite uma String, e só aceite Strings cujo comprimento seja maior ou igual a 10 caracteres.
- 2) Escreva um programa em Java que leia uma String do teclado e imprima seus caracteres, um por um.
- 3) Escreva um programa em Java que receba uma String e um valor numérico N do teclado, e imprima somente os primeiros N caracteres da String. O programa deve verificar se N é um valor válido (menor que o comprimento da String).
- 4) Escreva um programa em Java que leia uma String do teclado e troque todas as vogais não acentuadas da String pelo caracter '?'. [1] [SEP]
- 5) Escreva um programa em Java que filtre uma String, de forma que a partir de uma String lida do teclado, o programa crie uma nova String contendo somente os valores numéricos da String entrada. Por exemplo, se a String entrada for "1a2b09cd", a String resultante deve ser "1209". Dica: você pode usar o método estático isDigit() da classe Character. [1] [SEP]
- 6) Escreva um programa em Java que tenha um método chamado é URL que recebe uma String como argumento e verdadeiro se esta String começa com "http:" ou "ftp:" ou falso caso contrário. [1] [SEP]
- 7) Escreva um programa que calcule a área de um círculo, usando o valor constante de pi e o valor do raio lido do teclado. [1] [SEP]
- 8) Escreva uma classe Dado que tenha um método estático arremesso() que ao ser chamado imprima um valor entre 1 e 6. [1] [SEP]
- 9) Escreva uma classe Triangulo que encapsule três valores de ponto flutuante correspondentes aos lados do triângulo. Os valores devem ser passados para o construtor da classe. Nesta classe, escreva os métodos calculaPerímetro() e calculaÁrea() que calculam o perímetro e área do triângulo. Escreva também o método tipo() que retorna 1 se o triângulo for escaleno, 2 se o triângulo for isósceles e 3 se for equilátero. [1] [SEP]
- 10) Escreva um programa em Java que sorteie 50 números entre 0 e 99 para jogar na Lotomania. O programa não deverá sortear duas vezes o mesmo número. Dica: use um array para armazenar os números sorteados.
- 11) Escreva um programa em Java que declare e inicialize um vetor de booleanos (lendo-os do teclado), e calcule quantos elementos são iguais a true.
- 12) Escreva um programa em Java que crie um array de Strings e o popule (através da leitura dos seus elementos pelo teclado). Após isto, faça com que o programa calcule e imprima quantas Strings tem menos do que dez caracteres.
- 13) Escreva um programa em Java que peça ao usuário uma frase e verifique se nela há as palavras "-help" ou "-h".

## ANEXO B – EXERCÍCIO PRÁTICA POO E JAVA

1. Criar uma classe **Pessoa** com os seguintes itens:
  - nome;
  - idade;
  - peso;
  - altura;
  - calcular o IMC ( $\text{peso}/\text{altura}^2$ );
2. Desenvolva, usando a UML (diagrama de classe), uma classe que modele um objeto **pirâmide** em conformidade com o paradigma orientado a objeto. Posteriormente implemente esta classe. A classe deverá ter as seguintes características: base, altura e calcular volume.  $\text{Volume} = (1/3 * \text{base} * \text{altura})$
3. Desenvolva, usando a UML (diagrama de classe), uma classe que modele um objeto **esfera** em conformidade com o paradigma orientado a objeto. Posteriormente implemente esta classe. A classe deverá conter as seguintes características: raio e cálculo da área e do volume da esfera. Obs.: o valor do raio não pode ser negativo. A área total =  $(4 * 3.1415 * \text{raio}^2)$  e volume =  $((4/3) * 3.1415 * \text{raio}^3)$ .
4. Desenvolva, usando a UML (diagrama de classe), uma classe que modele um objeto **retângulo** em conformidade com o paradigma orientado a objeto. Posteriormente implemente esta classe. A classe deverá ter as seguintes características: comprimento, largura e cálculo do perímetro e da área do retângulo. Obs.: os valores da largura e do comprimento não podem ser negativos. O cálculo da área ( $\text{largura} * \text{comprimento}$ ) e do perímetro (soma dos quatro lados).
5. Desenvolva, usando a UML (diagrama de classe), uma classe que modele um objeto **paralelepípedo** em conformidade com o paradigma orientado a objeto. Posteriormente implemente esta classe. A classe deverá ter as seguintes características: altura, largura, comprimento, calcular volume e calcular área. O cálculo do volume ( $\text{altura} * \text{comprimento} * \text{largura}$ ) e o cálculo da área ( $2 * (\text{altura} * \text{largura} + \text{altura} * \text{comprimento} + \text{largura} * \text{comprimento})$ ).
6. Desenvolva, usando a UML (diagrama de classe), uma classe que modele um objeto **cilindro** em conformidade com o paradigma orientado a objeto. Posteriormente implemente esta classe. A classe deverá ter as seguintes características: raio, altura e cálculo da área lateral, área total e volume. Obs.: os valores do raio e da altura não podem ser negativos. O cálculo da área lateral é  $(2 * 3.1415 * \text{raio} * \text{altura})$ , área total  $(2 * 3.1415 * \text{raio} * (\text{altura} + \text{raio}))$  e volume  $(3.1415 * \text{raio}^2 * \text{altura})$ .
7. Desenvolva, usando a UML (diagrama de classe), uma classe que modele um objeto **cone** em conformidade com o paradigma orientado a objeto. Posteriormente implemente esta classe. A classe deverá ter as seguintes características: raio, altura e cálculo da geratriz, área lateral, área total e o volume. Obs.: os valores do raio e da altura não podem ser negativos. O cálculo da geratriz é  $(\text{Raiz Quadrada}((\text{altura}^2) + (\text{raio}^2)))$ , da área lateral é  $(3.1415 * \text{raio} * \text{geratriz})$ , da área total é  $(3.1415 * \text{raio} * (\text{geratriz} + \text{raio}))$  e do volume é  $(1.0/3.0 * 3.1415 * \text{raio}^2 * \text{altura})$ .
8. Desenvolva, usando a UML (diagrama de classe), uma classe que modele um objeto **conta poupança** em conformidade com o paradigma orientado a objeto. Posteriormente implemente esta classe. A classe deverá ter as seguintes características: nome do cliente, número da conta, agência, saldo, sacar dinheiro e depositar dinheiro.

9. Desenvolva, usando a UML (diagrama de classe), uma classe que modele um objeto **funcionário** em conformidade com o paradigma orientado a objeto. Posteriormente implemente esta classe. A classe deverá ter as seguintes características: nome do funcionário, CPF, número de horas trabalhadas, valor da hora de trabalho, calcular salário (número de horas trabalhadas \* valor da hora de trabalho).
10. Desenvolva, usando a UML (diagrama de classe), uma classe que modele um objeto **funcionário** em conformidade com o paradigma orientado a objeto. Posteriormente implemente esta classe. A classe deverá ter as seguintes características: nome, salário, número de filhos, calcular aumento de salário (a porcentagem é fornecida por parâmetro), calcular desconto no imposto de renda (R\$1250,25 por filho) e calcular INSS (6% para salário ≤ R\$2545,00 e 10% para o restante).
11. Desenvolva, usando a UML (diagrama de classe), uma classe que modele um objeto **paciente** em conformidade com o paradigma orientado a objeto. Posteriormente implemente esta classe. A classe deverá ter as seguintes características: nome, peso, altura, calcular IMC e calcular a faixa de peso. O cálculo do IMC =  $\text{peso}/(\text{altura}^2)$ .

IMC	Faixa de Risco
abaixo de 20	abaixo do peso ideal
a partir de 20 até 25	peso normal
acima de 25 até 30	excesso de peso
acima de 30 até 35	obesidade
acima de 35	obesidade mórbida