

CÓDIGO DO CURSO	CURSO
GSISNF	BACHARELADO EM SISTEMAS DE INFORMAÇÃO

CÓDIGO DA DISCIPLINA	NOME DA DISCIPLINA	ANO/SEMESTRE
GS19301NF	PROGRAMAÇÃO ORIENTADA A OBJETOS	2022/1

CARGA HORÁRIA	CRÉDITOS	AULAS/SEMANA				PERÍODO
75 horas (90 h-a)	4	TEÓRICA	PRÁTICA	ESTÁGIO	NÃO PRESENCIAL	3º
		3	2	0	0	

PRÉ - REQUISITO (S)	CORREQUISITO (S)
Programação Estruturada	-----

EMENTA
História e evolução da linguagem Java. Ambiente de execução. Convenções da linguagem. Estruturas de decisão e repetição. Paradigma estruturado x paradigma orientado a objetos. Orientação a objetos – conceitos básicos e avançados. Tratamento de exceções. Tipos enumerados. Aprofundamento de entrada e saída de dados. Estruturas de dados. Manipulação de data e hora. Camada de persistência: acesso a banco de dados e as operações CRUD. Padrões aplicáveis à camada de persistência. Serialização, programação genérica, reflexão e threads.

BIBLIOGRAFIA
BIBLIOGRAFIA BÁSICA
- CORNELL, Gary. Core Java, volume 1: fundamentos. 8.ed. São Paulo: Pearson, 2010. xiii, 383, il. ISBN 9788576053576:(broch.). - DEITEL, Paul J.; DEITEL, Harvey M., 1945-. Java: como programar. 8. ed. São Paulo: Pearson Education do Brasil, 2010. xxix, 1144 p., il. color. ISBN 9788576055631 (Broch.). - SIERRA, Kathy. Use a cabeça: Java. 2.ed. Rio de Janeiro: Alta Books, 2007. xvii, 484 p., il. ISBN 9788576081739 (Broch.).
BIBLIOGRAFIA COMPLEMENTAR

1. KERIEVSKY, Joshua. **Java Efetivo**. 2.ed. Rio de Janeiro: Alta Books, 2010. xi, 297. ISBN 9788576082804 (broch.).
2. FREEMAN, Eric et al. **Use a cabeça: padrões e projetos**. 2.ed. rev. Rio de Janeiro: Alta Books, 2009. xxiv, 478p., il. ISBN 9788576081746.
3. KURNIAWAN, Budi. **Java para a Web com Servlets, JSP e EJB**. Rio de Janeiro: Ciência Moderna, c2002. xxiv, 807p., il. Inclui índice. ISBN 8573932104 (Broch.).
4. SCHILDT, Herbert, 1951-. **Java: a referência completa**. 1.ed. Rio de Janeiro: Alta Books, 2014. 1096 p. ISBN 9788576087557.
5. **Introdução à arquitetura e design de software: uma visão sobre a plataforma Java**. Rio de Janeiro: Campus, 2012. xvi, 257p., il., 25 cm. Bibliografia: p. [237]-252. ISBN 9788535250299 (broch.).

OBJETIVOS GERAIS

1. Identificar as diferenças do paradigma da orientação a objetos com relação ao paradigma estruturado;
2. Reconhecer as convenções e particularidades da linguagem ao aplicar os conceitos de orientação a objetos;
3. Estabelecer boas práticas no desenvolvimento de sistemas orientados a objetos;
4. Desenvolver programas com tratamento correto de exceções;
5. Mostrar conceitos e padrões na prática, para que servem e em que situações devem ser utilizados, para extrair o melhor do paradigma orientado a objetos;
6. Trabalhar com entrada e saída de dados de forma aprofundada;
7. Trabalhar com estruturas de dados da linguagem;
8. Trabalhar com uma camada de persistência;
9. Introduzir o conceito de *threads* e aplicá-los em alguns exercícios.

METODOLOGIA

Aulas divididas entre teoria e prática, com aplicação e correção de exercícios relativos a todos os tópicos. Todas as aulas serão realizadas em laboratório devidamente equipado com ferramentas de apoio.

CRITÉRIO DE AVALIAÇÃO

Provas escritas individuais serão o principal método de avaliação. Trabalhos de pesquisa e/ou seminários poderão ser aplicados para compor a nota.

PROGRAMA

1. Conceitos básicos
 - 1.1. História e evolução da linguagem
 - 1.2. Conceito de máquina virtual e seu funcionamento
 - 1.3. Compilação dinâmica e pontos quentes da aplicação
2. Convenções da linguagem, variáveis, constantes e tipos de dados
 - 2.1. Formação de identificadores
 - 2.2. Declaração de variáveis e constantes
 - 2.3. Comando de atribuição
3. Operadores, arrays, desvios, laços e escopo
 - 3.1. Operadores aritméticos básicos: +, -, /, *, %
 - 3.2. Arrays em Java
 - 3.3. Desvio condicional e laços básicos: if/else, while, do/while e for

- 3.4. Operadores relacionais: >, <, =, <>, >=, <=
- 3.5. Operadores lógicos; e, ou, não
- 3.6. Interferindo no loop
- 3.7. Escopo das variáveis
- 4. Paradigma orientado a objetos x Paradigma procedural
 - 4.1. Diferenças entre os paradigmas e motivação para adoção do paradigma OO
 - 4.2. Definições de classe e objeto
 - 4.2.1. Representações gráficas de Classes
 - 4.3. Métodos de uma classe
 - 4.4. Objetos e referências
- 5. Conceitos básicos e boas práticas da orientação a objetos
 - 5.1. Alta coesão e baixo acoplamento
 - 5.2. Princípio da responsabilidade única
 - 5.3. Agregação e composição
 - 5.4. Modificadores de acesso
 - 5.5. Encapsulamento e métodos acessórios
 - 5.6. Construtores
 - 5.7. Atributos e métodos estáticos
 - 5.8. Herança e polimorfismo
 - 5.9. Destruição de um objeto
- 6. Outros conceitos da orientação a objetos
 - 6.1. Reescrita de métodos
 - 6.2. Classes e métodos abstratos
 - 6.3. Atributos, parâmetros, classes e métodos final
 - 6.4. Sobrecarga
 - 6.5. Interfaces
 - 6.6. Enumerações
 - 6.7. Serialização
- 7. Tratamento de Exceções
 - 7.1. Motivações, conceitos e funcionamento
 - 7.2. Sintaxe de captura de exceção
 - 7.2. Captura de exceções distintas e ordem de avaliação
 - 7.2.1. Captura geral
 - 7.2.2. Declaração de exceções e tipos de hierarquias de exceções
 - 7.3. Níveis de abstração
 - 7.4. Relançamento de exceções
 - 7.4.1. Motivação para relançamento de exceções
 - 7.5. Boas práticas para capturas de exceções
 - 7.6. Garantias pós exceção
 - 7.7. Motivações do uso de garantias pós exceção
 - 7.8. Boas práticas e estratégias de tratamento de exceções para aplicações
- 8. Entrada e saída de dados
 - 8.1. Leitura e escrita em arquivos
 - 8.2. Estratégias e padrões para flexibilidade dos fluxos de entrada e saída
 - 8.3. As classes File, Scanner e PrintStream
- 9. Programação genérica e coleções
 - 9.1. Programação genérica
 - 9.2. Coleções de dados (Listas, Conjuntos, Filas e Mapas)
 - 9.3. Estratégias para utilização de tipos de coleções

10. Bibliotecas para tratamento de datas e o pacote java.lang
 - 10.1. Apresentação das bibliotecas para tratamento de datas
 - 10.2. Cuidados com a conversão de datas
 - 10.3. Autoboxing e Unboxing
 - 10.4. Reescrevendo equals e toString
 - 10.5. Trabalhando com pacotes
11. Camada de persistência e Padrões de Projeto aplicáveis
 - 11.1. Utilizando JDBC
 - 11.2. Utilização do padrão Repository para simular coleções de objetos
 - 11.3. Operações de CRUD
 - 11.4. Controle de transação
 - 11.5. Utilização do padrão Factory Method
 - 11.6. Utilização do padrão Singleton
 - 11.7. Conexão única sem o uso de Singleton
12. Camada de persistência e Padrões de Projeto aplicáveis
 - 12.1. Threads
 - 12.2. A interface Runnable e a classe Thread
 - 13.3. Solucionando o problema do produtor e consumidor usando threads

PROFESSOR RESPONSÁVEL

NOME - SIAPE **Rafael Guimarães Rodrigues - 2968766**

ASSINATURA

COORDENADOR DO CURSO

NOME - SIAPE **Rodrigo Reis Gomes - 1644791**

ASSINATURA

LOCAL E DATA

Nova Friburgo, 19 de novembro de 2021.