



FACULDADE DE CIÊNCIAS E TECNOLOGIA

CURSO DE ENGENHARIA EM TECNOLOGIAS E SISTEMAS DE INFORMAÇÃO

PLANO ANALÍTICO

Disciplina : Programação Orientada a Objectos Avançada		Estatuto: Nuclearr
Ano Académico: 2022		Semestre: 2
Carga Horária Semestral : 96h		Carga Horária Semanal: 6h
Horas de interação com o docente: 96	Horas de estudo independente : 84	Nº de Créditos:
Docente: Joseph Katame	Tel: 823064067 / 848477722	E-mail: jkatame@gmail.com

I. OBJECTIVOS DA CADEIRA

1. Objectivo Geral

Compreender conceitos de programação orientada a objectos, arquitectura e camadas de software, padrões de projecto, trabalho em equipa e modelagem de programação orientada a objectos.

2. Objectivos Específicos

Ao final da disciplina é esperado que o estudante:

- Entenda com profundidade os conceitos de Programação Orientada a Objectos, sendo capaz de aplicá-los de maneira apropriada;
- Seja fluente na escrita, teste e depuração de programas orientados a objectos, com uso de APIs;
- Seja capaz de desenvolver sistemas orientados a objectos com acesso e abstraindo banco de dados;
- Seja capaz de desenvolver sistemas utilizando objectos remotos.



UNIVERSIDADE
JOAQUIM CHISSANO

FACULDADE DE CIÊNCIAS E TECNOLOGIA

II. UNIDADES TEMÁTICAS

TEMA	CARGA HORÁRIA
1. Classes e Objectos	10
2. Herança Restrita e Polimorfismo	6
3. Classes Abstractas	8
4. Interfaces	10
5. Controlo de Erros - Excepções	8
6. Organização de classes em Bibliotecas	8
7. Collection Framework	4
8. Genéricos	4
9. Colecções	8
10. Objectos persistentes	6
11. Arquitectura J2EE	8
12. Concorrência	
TOTAL DE HORAS	76 Hrs

III. PLANO ANALÍTICO

Tema	Conteúdo	Objectivos de Aprendizagem	Bibliografia	Obs.
Semana 1 (08 a 12 de Ago.)				
Classes e Objectos	Vantagens da orientação a objectos Conceitos Classes e objectos Instâncias vs Classes Acessando objectos por referência Packages Mecanismos de controlo de acesso Encapsulamento. Exercícios.	<ul style="list-style-type: none"> • Detectar o que são variáveis locais de um método e como elas diferem de variáveis de instância. • Distinguir o que são tipos primitivos e tipos de referência. • Utilizar a palavra-chave this em um construtor para chamar outro construtor na mesma classe. • Usar variáveis e métodos static. • Importar membros static de uma classe. 	<ul style="list-style-type: none"> • P. J. DEITEL, H. M. DEITEL. Java Como Programar. Décima edição. Pearson, 2016. • http://www.caelum.com.br/caelum/formação 	
Semana 2 (15 a 19 de Ago.)				
Classes e Objectos	Exercícios práticos e laboratoriais	Praticar os conceitos aprendidos em aulas teóricas	<ul style="list-style-type: none"> • P. J. DEITEL, H. M. DEITEL. Java Como Programar. Décima edição. Pearson, 2016. • http://www.caelum.com.br/caelum/formação 	
Semana 3 (22 a 26 de Ago.)				
Herança Restrita e Polimorfismo	Conceito de Herança. Hierarquia de classes e	<ul style="list-style-type: none"> • Entender o que é herança e como usá-la para desenvolver novas 	<ul style="list-style-type: none"> • P. J. DEITEL, H. M. DEITEL. Java Como Programar. Décima edição. Pearson, 2016. 	

	<p>mecanismo de ligação. Herança de classes em Java. Princípio da substitutividade. Métodos equals e clone. Exercícios práticos e laboratoriais</p>	<p>classes com base nas existentes.</p> <ul style="list-style-type: none"> • Aprender noções de superclasses e subclasses, além do relacionamento entre elas. • Utilizar a palavra-chave extends para criar uma classe que herda atributos e comportamentos de outra classe. • Empregar o modificador de acesso protected em uma superclasse para dar a métodos de subclasse acesso aos membros desta superclasse. 	<ul style="list-style-type: none"> • http://www.caelum.com.br/caelum/formação 	
Semana 4 (29 de Ago. a 03 de Set.)				
Polimorfismo	<p>Sobrecarga (overloading) de métodos. Sobreposição (overriding) de métodos. Ocultação (hiding) de variáveis. Utilização de this e super em construtores. Construtor de cópia. Princípio da substitutividade. Exercícios.</p>	<ul style="list-style-type: none"> • Aprender o conceito de polimorfismo. • Usar métodos sobrescritos para impactar o polimorfismo • Acessar membros da superclasse Com super a partir de uma subclasse. • Entender como os construtores são usados em hierarquias de herança. • Aprender os métodos da classe Object, a superclasse directa ou indirecta de todas as classes. 	<ul style="list-style-type: none"> • P. J. DEITEL, H. M. DEITEL. Java Como Programar. Décima edição. Pearson, 2016. • http://www.caelum.com.br/caelum/formação 	



**UNIVERSIDADE
JOAQUIM CHISSANO**

FACULDADE DE CIÊNCIAS E TECNOLOGIA

Semana 5 (05 a 09 de Set.)				
Classes Abstractas e Interfaces	<p>Classe Abstracta. Interfaces (em Java). Classes Abstractas versus Interfaces. Exercícios. Supervisão de TP.</p>	<ul style="list-style-type: none"> Distinguir entre classes abstratas e concretas. Declarar métodos abstratos para criar classes abstratas. Declarar e implementar interfaces. 	<ul style="list-style-type: none"> P. J. DEITEL, H. M. DEITEL. Java Como Programar. Décima edição. Pearson, 2016. http://www.caelum.com.br/caelum/formação 	
Semana 6 (12 a 16 de Set.)				
Exceções	<p>Exceções verificáveis pelo compilador. Exceções não verificáveis pelo compilador. Herança e cláusula throws. Exercícios.</p>	<ul style="list-style-type: none"> Entender o que são exceções e como elas são tratadas. Aprender quando usar o tratamento de exceção. Usar blocos try para delimitar o código em que podem ocorrer exceções. Usar throw para indicar um problema. Usar blocos catch para especificar rotinas de tratamento de exceção. Usar o bloco finally para liberar recursos. Compreender a hierarquia de classes de exceção. Criar exceções definidas pelo usuário. 	<ul style="list-style-type: none"> P. J. DEITEL, H. M. DEITEL. Java Como Programar. Décima edição. Pearson, 2016. 	

Semana 7 (19 a 23 de Set.)				
Organizando classes em bibliotecas	Por que organizar as classes? Pacotes da API Java Modularidade Declaração do pacote Convenção de nomes Importação Localização de pacotes O pacote padrão Especificadores de acesso Ferramentas JAR e JAVADOC Exercícios práticos e laboratoriais	<ul style="list-style-type: none"> Compreender a necessidade de organização de classes em bibliotecas Familiarizar-se com algumas bibliotecas de Java Declarar e usar pacotes Entender e aplicar os especificadores de acesso 	<ul style="list-style-type: none"> P. J. DEITEL, H. M. DEITEL. Java Como Programar. Décima edição. Pearson, 2016. http://www.caelum.com.br/caelum/formação 	
Semana 8 (26 a 30 de Set.)				
Preparação para T1.	Teste1 Supervisão do TP	Avaliar a compreensão dos conteúdos lecionados.		
Semana 9 (03 a 07 de Out.)				
Collection Framework	<ul style="list-style-type: none"> Genéricos <ul style="list-style-type: none"> Métodos Genéricos Classes Genéricas Tipos "Crus" Coringas em Métodos Genéricos Genéricos e Herança Exercícios práticos e laboratoriais	<ul style="list-style-type: none"> Entender e aplicar métodos e classes genéricas Aplicar a herança em genéricos 	<ul style="list-style-type: none"> P. J. DEITEL, H. M. DEITEL. Java Como Programar. Décima edição. Pearson, 2016. [Gosling] "The Java Language Specification" <i>James Gosling, Bill Joy and Guy Steele</i>, Addison-Wesley. 	

Semana 10 (10 a 14 de Out.)				
Collection Framework	<ul style="list-style-type: none"> ▪ Coleções <ul style="list-style-type: none"> ▪ Classe <i>Arrays</i> ▪ Interface <i>Collection</i> e Classe <i>Collections</i> <p>Exercícios práticos e laboratoriais Supervisão do TP</p>	<ul style="list-style-type: none"> • Entender e aplicar a classe <i>Arrays</i> • Aplicar a interface <i>Collection</i> e classe <i>Collections</i> 	<ul style="list-style-type: none"> • P. J. DEITEL, H. M. DEITEL. Java Como Programar. Décima edição. Pearson, 2016. • [Gosling] “The Java Language Specification” <i>James Gosling, Bill Joy and Guy Steele</i>, Addison-Wesley. 	
Semana 11 (17 a 21 de Out.)				
Collection Framework	<ul style="list-style-type: none"> ▪ Listas <ul style="list-style-type: none"> ▪ <i>ArrayList</i> e <i>Iterator</i> ▪ <i>LinkedList</i> ▪ <i>Vector</i> ▪ Algoritmos <p>Exercícios práticos e laboratoriais</p>	<ul style="list-style-type: none"> • Entender e aplicar as listas da biblioteca Java (<i>ArrayList</i> e <i>Iterator</i>, <i>LinkedList</i>) • Entender e aplicar a classe <i>Vector</i> do Java • Entender e aplicar algoritmos do Java Collections Framework 	<ul style="list-style-type: none"> • P. J. DEITEL, H. M. DEITEL. Java Como Programar. Décima edição. Pearson, 2016. • [Gosling] “The Java Language Specification” <i>James Gosling, Bill Joy and Guy Steele</i>, Addison-Wesley. 	
Semana 12 (24 a 28 de Out.)				
Collection Framework	<ul style="list-style-type: none"> ▪ Pilhas ▪ Filas de Prioridade ▪ Conjuntos ▪ Mapas <p>Exercícios práticos e laboratoriais</p>	Entender e aplicar as coleções de Pilhas, Filas de Prioridade, Conjuntos e Mapas.	<ul style="list-style-type: none"> • P. J. DEITEL, H. M. DEITEL. Java Como Programar. Décima edição. Pearson, 2016. • [Gosling] “The Java Language Specification” <i>James Gosling, Bill Joy and Guy Steele</i>, Addison-Wesley. 	



UNIVERSIDADE
JOAQUIM CHISSANO

FACULDADE DE CIÊNCIAS E TECNOLOGIA

Semana 13 (31 de Out. a 04 de Nov.)

Concorrência	<p>Ciclo de vida e estados de thread</p> <ul style="list-style-type: none"> • Estados novo e executável • Estado de espera • Estado de espera sincronizada • Estado bloqueado • Estado terminado <p>Servidores para múltiplas requisições simultâneas.</p> <p>Sincronização de threads</p> <p>O problema do Produtor / Consumidor</p> <p>Exercícios práticos e laboratoriais</p>	<ul style="list-style-type: none"> • Entender concorrência, paralelismo e multithreading. • Aprender o ciclo de vida de uma thread. • Utilizar métodos synchronized para coordenar o acesso a dados mutáveis compartilhados. • Compreender os relacionamentos produtor/consumidor. • Usar fluxos paralelos para melhor desempenho em sistemas multiprocessados. 	<ul style="list-style-type: none"> • Jeff Magee, Jeff Kramer. Concurrency – State Models and Java Programs, , John Wiley 1999. • [Gosling] “The Java Language Specification” <i>James Gosling, Bill Joy and Guy Steele</i>, Addison-Wesley. 	
---------------------	---	--	--	--

Semana 14 (07 a 11 de Nov.)

Realização de segunda avaliação escrita	Preparação para Teste2	Avaliar a compreensão dos conteúdos lecionados.		
--	------------------------	---	--	--

Semana 15 (14 a 18 de Nov.)

Defesas	Apresentação e defesas de TP			
----------------	------------------------------	--	--	--

Semana 16 (21 a 25 de Nov.)				
Defesas e reposição de avaliações	Apresentação e defesas de TP Avaliações de reposição			

IV. ESTRATÉGIAS E MÉTODOS DE ENSINO E APRENDIZAGEM

Para o alcance dos objectivos plasmados neste plano analítico o método de ensino-aprendizagem é centrado no estudante com recurso a:

- Mapas e aulas expositivas-explicativas;
- Debates
- Consolidação da experiência com os temas tratados em aula
 - Listas de exercícios a cada conteúdo apresentado.
 - Trabalhos para casa
 - Trabalhos práticos

VI. BIBLIGRAFIA

- P. J. DEITEL, H. M. DEITEL. Java Como Programar. Décima edição. Pearson, 2016.
- B. MEYER. *Object-Oriented Software Construction*. Segunda Edição. Prentice-Hall, 2000.
- Jeff Magee, Jeff Kramer. Concurrency – State Models and Java Programs, , John Wiley 1999.



FACULDADE DE CIÊNCIAS E TECNOLOGIA

- **[Gosling]** “The Java Language Specification” *James Gosling, Bill Joy and Guy Steele*, Addison-Wesley.