



## Plano de Aula

### 1 Código e nome da disciplina

ARA0075 PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

### 2 Semana/Tema

Semana 1: Tema - 1. INTRODUÇÃO À PROGRAMAÇÃO OO EM JAVA

### 3 Objetivos

Distinguir o paradigma de orientação a objeto em face do paradigma estruturado, fazendo a reflexão sobre os conceitos chaves de classe e objetos e suas coleções, para construir programas que reflitam estruturas do mundo real.

### 4 Tópicos

1.1 CLASSES E OBJETOS

### 5 Procedimentos de ensino-aprendizagem

Apontar as motivações para estudar o paradigma de programação orientado a objetos. Como sugestão, segue o roteiro abaixo:

-Situação-problema:

Abstrair um problema é parte importante do processo de desenvolvimento de software, dessa forma, quanto mais próximo da realidade de um programador, que é um ser humano, o problema a ser resolvido possa ser representado, melhor será o seu entendimento e consequentemente, a implementação torna-se mais simples e inteligível para todos os envolvidos no processo de desenvolvimento, inclusive do cliente para o qual o programa está sendo desenvolvido. Vamos analisar a seguinte situação: Um programador é contratado para desenvolver um software para uma empresa de produtos de beleza, e o software em questão possui vários módulos; como o responsável pelo Recursos Humanos da Empresa, controle de estoque, marketing, controle da frota de caminhões. Todos esses módulos funcionam como peças de um todo, mesmo sendo independentes; sendo assim, diversos programadores poderiam trabalhar de forma independente em cada um dos módulos. Além disso, seria possível adquirir módulos prontos sem ao menos conhecer a implementação deles, utilizando apenas interfaces de comunicação. Quais as vantagens você enxerga nesse tipo de abordagem em relação ao paradigma estruturado? Levante questões em relação ao tempo de desenvolvimento, custos, reuso entre outras que achar pertinente.

- Metodologia:

O professor deve destacar os benefícios do paradigma de programação orientada a objetos em relação ao paradigma estruturado, apresentando um quadro comparativo entre eles. Após a apresentação inicial, o professor deve então apresentar o conceito de classes e objetos, fazendo sempre um link com

o mundo real de forma que o aluno compreenda a finalidade do paradigma orientado a objetos bem como realizar a conexão entre o conceito de classes e objetos e o mundo real. Nesse ponto, o professor deve apresentar diversos exemplos de classes que façam relação tanto com entidades do mundo real como entidades ligadas diretamente a um elemento de software como um botão de uma interface gráfica. O professor deve deixar bem claro a diferença entre uma classe e um objeto. Nessa aula o professor deve fazer uma revisão dos tipos de dados exemplificando a partir de atributos de classes e retorno de métodos. Orientar os alunos a realizar a leitura de [1] para fixação dos conceitos permeados na aula.

- Atividade verificadora de aprendizagem:

Nessa aula, o professor deve dividir a turma em grupos e propor para que se identifique as classes e atributos para cada uma das situações citadas abaixo:

A) Em uma turma de um curso de graduação temos disciplinas ministradas em salas diferentes

B) A transportadora "Super Cometa" monitora seus caminhões e motoristas através de um controlador embutido. Por meio dele é possível saber a localização de cada veículo.

Ao final da aula o professor deve trazer uma possível solução e realizar a discussão com os alunos.

## 6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual e ambiente de programação Netbeans instalado nos computadores.

## 7 Leitura específica

[1] FURGERI, Sérgio. Java 8 - Ensino Didático: Desenvolvimento e Implementação de Aplicações. Capítulo 7 - Orientação a Objetos. Disponível em:  
<https://integrada.minhabiblioteca.com.br/#/books/9788536519340/cfi/28!/4/2@100:0.00>

## 8 Aprenda +

Assista ao vídeo Orientação a Objetos: Disponível em: <https://www.youtube.com/watch?v=Gq1BS63pkRA>

Assista ao vídeo Orientação a Objetos: classes e métodos simples: Disponível em:  
[https://www.youtube.com/watch?v=-t\\_c6F\\_Uoeg](https://www.youtube.com/watch?v=-t_c6F_Uoeg)

### Atividade Autônoma Aura:

1) O paradigma de programação orientada a objetos nasceu da necessidade de trazer o entendimento de problemas computacionais para mais próximo do mundo real. Levando em consideração os conceitos de programação orientada a objetos, julgue os itens a seguir:

I - Classes e objetos possuem atributos e métodos, no entanto, uma classe é apenas um modelo que é usado para criar objetos diferentes do mesmo tipo.

II - Os atributos de um objeto são o que ele sabe fazer e o métodos são o que ele sabe.

III - uma classe é capaz de armazenar estados através de seus atributos e reagir a mensagens enviadas

a ela, assim como se relacionar e enviar mensagens a outras classes.

Está(ão) correta(s):

- A) I
- B) I e II
- C) I, II e III
- D) II e III
- E) III

2) Os paradigmas de orientação a objeto e estruturado tem aplicações diversas na computação pois apresentam características distintas que os tornam adequados a aplicações específicas. Por exemplo, para desenvolver um programa para uma empresa de Recursos Humanos que irá ajudar a selecionar candidatos para um perfil de vaga específico e será modificado e atualizado constantemente, o paradigma mais adequado seria o orientando a objetos, mas se você irá desenvolver um software para o controle do frio ABS de um automóvel, o qual irá executar em um microcontrolador que possui diversas restrições de memória e processamento, seria mais adequado a utilização do paradigma estruturado. Diante dessas duas situações motivadoras e de seus conhecimentos em relação aos paradigmas orientado a objetos e estruturado, relacione as características com os respectivos paradigmas

I - Orientado a objetos

II - Estruturado

- ( ) Reutilização de código
- ( ) Tratamento de dados misturados com o comportamento do programa
- ( ) utiliza um conjunto de procedimentos para resolver um problema
- ( ) representação de problemas de forma abstrata

- A) I, I, II, II
- B) II,I,II,I
- C) II,II,I,II
- D) I,I,II,I
- E) I, II, I, II



## Plano de Aula

### 1 Código e nome da disciplina

ARA0075 PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

### 2 Semana/Tema

Semana 2: Tema - 1. INTRODUÇÃO À PROGRAMAÇÃO OO EM JAVA

### 3 Objetivos

Explorar os conceitos de polimorfismo e herança, apresentando exemplos práticos e de implementação simples, de forma que os alunos tenham uma base sólida desses dois fundamentos da orientação a objetos.

### 4 Tópicos

1.2 IMPLEMENTAÇÃO DE HERANÇA E POLIMORFISMO: O BÁSICO

### 5 Procedimentos de ensino-aprendizagem

O paradigma orientado a objetos possui alguns pilares importantes, um deles é a herança que é um mecanismo utilizado para reaproveitamento de propriedades e métodos de classes existentes. Nessa aula o professor deve abordar tanto o conceito de herança quanto o conceito de polimorfismo. Como sugestão, o professor poderá seguir o roteiro abaixo:

#### Situação-problema:

Sistemas computacionais muitas vezes acabam reutilizando componentes em vários pontos do programa. Por exemplo, interfaces gráficas utilizam muitos botões, os quais possuem atributos e características muito parecidas, como forma retangular, método de execução quando clicado entre outros. Supondo que em um ambiente específico seja necessário utilizar um botão de formar circular para representar o botão de um elevador. Notamos que todos os outros comportamentos e atributos do botão ?retangular? serão utilizadas pelo botão ?circular?. Então, não seria interessante que o botão ?circular? herdasse as características do botão ?retangular? realizando apenas as alterações necessárias ou acréscimo de atributos e métodos? Esse mecanismo traria que tipos de vantagens no desenvolvimento de sistemas?

#### Metodologia:

O professor deve iniciar a aula apresentando o conceito de herança, abordando sobre superclasse e subclasse e sua relação, optando por utilizar exemplos do mundo real. Após a apresentação teórica, deve apresentar ao aluno a palavra-chave extends e uma aplicação simples de herança na linguagem Java, como sugestão o professor pode apresentar a seguinte situação: Uma indústria necessita de um sistema para cadastrar produtos produzidos. Ela produz Telefones cujos atributos são: código, número de série, modelo, peso e dimensão. Essa mesma empresa começou a produzir telefones sem fio que

compartilham as mesmas características do telefone com fio, porém possuem novas características que são: frequência, quantidade de canais e distância de operação. O professor deve codificar a situação proposta explicando-a aos alunos. Em seguida deve apresentar o conceito de polimorfismo no qual duas ou mais classes derivadas de uma mesma superclasse podem invocar métodos que têm a mesma identificação (assinatura) mas comportamentos distintos. Como sugestão, o professor pode apresentar a seguinte situação: Crie uma classe chamada Forma2D que possui os atributos `dimensao1` e `dimensao2`; também possui o método `area` que retorna a área da figura. Crie duas classes que herdam as características e atributos da classe Forma2D: classe Retângulo e Quadrado. Os métodos de cálculo de área devem ser adaptados e executados para demonstrar o conceito de polimorfismo.

Atividade verificadora de aprendizagem:

Nessa aula, o professor deve dividir a turma em grupos e propor para cada um deles os seguintes problemas:

1. Apresente uma situação na qual seria possível e vantajoso a utilização do mecanismo de herança. Faça a sua implementação.
  2. Apresente uma situação em que o polimorfismo poderia ser aplicado. Faça a sua implementação
- A atividade deve ser desenvolvida durante a sala de aula e ao final os grupos devem apresentar as situações propostas.

## 6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual e ambiente de programação Netbeans instalado nos computadores.

## 7 Leitura específica

[1] Os conteúdos abordados nessa aula estão disponíveis no livro FURGERI, Sérgio. Java 8 - Ensino Didático: Desenvolvimento e Implementação de Aplicações. Capítulo 7 - Orientação a Objetos.

Seções 7.1 e 7.2 Disponível em:

<https://integrada.minhabiblioteca.com.br/#/books/9788536519340/cfi/28!/4/2@100:0.00>

## 8 Aprenda +

Assista ao vídeo:

- Orientação a Objetos: Herança. Disponível em: <https://www.youtube.com/watch?v=MOXLCjL4Ik4>

Atividade Autônoma Aura:

1) A empresa de desenvolvimento de software "Atlas Development" foi contratada por uma empresa de seguros para desenvolver um software na Linguagem Java que irá avaliar o perfil de dos contratantes de seguro para automóveis. A Atlas já desenvolveu sistemas parecidos para duas outras empresas. Em um dos módulos do sistema, um desenvolvedor verificou que se herdasse métodos e atributos de duas outras classes, poderia então desenvolver a classe necessária para ser utilizada em parte do módulo em questão. Nesse caso, que tipo de herança está sendo realizada e se Java suportaria esse mecanismo.

- a) Herança por prototipagem, Java implementa
- b) Herança múltipla, Java implementa
- c) Herança simples, Java implementa
- d) Herança múltipla, Java não implementa

e) Herança por prototipagem, Java não implementa

2) Um programador está desenvolvendo um software que irá controlar aceleradores de motores de forma automática, evitando por exemplo, que o motorista acelere se houver um objeto próximo ao veículo. O software deve ser adaptado para diversos tipos de veículos como automóveis, caminhões e motos. No entanto, sabemos que esses veículos possuem características distintas, mesmo apresentando o comportamento de aceleração do motor. Nesse caso, o programador deverá implementar o método `acelerar()` para todos os veículos, utilizando o mesmo nome mas com implementações distintas. Que conceito de orientação a objetos o problema acima está se referindo.

a) Herança

b) Abstração

c) Polimorfismo

d) Encapsulamento

e) Construtor



## Plano de Aula

### 1 Código e nome da disciplina

ARA0075 PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

### 2 Semana/Tema

Semana 3: Tema - 1. INTRODUÇÃO À PROGRAMAÇÃO OO EM JAVA

### 3 Objetivos

Agrupar objetos, utilizando a infraestrutura Collections, para implementar estruturas de dados como Listas e Filas, de forma a organizar e manipular um conjunto de objetos.

### 4 Tópicos

1.3 IMPLEMENTAÇÃO DE AGRUPANDO OBJETOS

### 5 Procedimentos de ensino-aprendizagem

Organizar objetos em agrupamentos é útil quando precisamos manipular, ordenar e realizar outras operações necessárias nesse conjunto. Java apresenta várias formas de agrupamentos as quais serão abordadas nesta aula. Como sugestão, o professor pode seguir o roteiro abaixo:

#### - Situação-problema:

Um programador foi contratado para desenvolver um bloco de notas que deve funcionar da seguinte forma: o usuário pode criar um conjunto de objetos notas que possui os seguintes atributos: nome, data de criação e informação. Nesse caso, verificamos que o Bloco de Notas é composto por um agrupamento de várias notas. Como podemos organizar esse agrupamento? Existe apenas uma forma de agrupá-los?

#### - Metodologia:

O professor deve iniciar a aula apresentando o conceito de collections e seus tipos: set (conjunto), List (lista), Queue (fila) e Map (mapa). Após a apresentação inicial, o professor deve, a partir de exemplos práticos, apresentar os métodos principais de cada um dos tipos de collections. Como sugestão, o professor pode implementar uma lista com os alunos da sala, que devem ser modelados como uma Classe contendo os atributos nome e matrícula. Deve utilizar o método add e remove e também deve criar um método que imprime um elemento da lista e todos os elementos da lista.

#### - Atividade verificadora de aprendizagem:

O professor deve retornar a situação-problema e deve propor que os alunos implementem um Bloco de Notas utilizando collections. Os alunos devem decidir qual é o tipo mais adequado para o problema apresentado. Lembrando que deve ser possível criar, excluir e alterar uma nota. Os alunos devem entregar a atividade pelo Sava, MS Teams ou Similar até a aula seguinte.

## 6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual e ambiente de programação Netbeans instalado nos computadores.

## 7 Leitura específica

[1] MACHADO, Rodrigo Prestes; FRANCO, Márcia Hafele Islabão; BERGANOLLI, Silva de Castro. Desenvolvimento de Software III: programação de sistemas web orientada a objetos em Java. 1ª Ed.. Porto Alegre: Bookman, 2016. Capítulo 3.  
Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788582603710/.00>

## 8 Aprenda +

Assista ao vídeo:

- Usando ArrayList em Java - Exemplo de Cadastro de Contatos. Disponível em:  
<https://www.youtube.com/watch?v=ISFI-gfBdgA>

Leia o artigo:

- Java Collections: Como utilizar collections: <https://www.devmedia.com.br/java-collections-como-utilizar-collections/18450>

### Atividade Autônoma Aura:

1) Um órgão público trabalha com tramitação de processos de forma eletrônica. Os processos (objeto Processo) devem ser armazenados em um ArrayList para que possam ser posteriormente manipulados. Supondo que você precise inserir um novo processo no ArrayList e que este processo já foi devidamente instanciado e foi denominado p1, seus atributos já foram preenchidos e que você deva inseri-lo no ArrayList já instanciado chamado de listaProcessos, qual instrução você utilizaria?

- a) listaProcessos.insert(p1);
- b) listaProcessos.set(p1);
- c) listaProcessos.add(p1);
- d) new listaProcessos.insert(p1);
- e) new listaProcessos.add(p1);

2) Um programador deve implementar um ranking dos melhores jogadores de um RPG que está fazendo sucesso nas lojas de aplicativos Mobile. Para isso ele pode utilizar as interfaces do Collection. Analise as afirmações a seguir e marque a alternativa correta.

I - Para implementar o ranking não podemos ter elementos duplicados, por isso o programador poderia utilizar a interface Set.

II - Caso o programador tente inserir um objeto duplicado em um Set, ele irá aceitar mas emitirá um aviso dizendo que há um registro duplicado.

III - O programador poderia utilizar a interface Map a qual estende Collections, para implementar o referido ranking.

Está(ão) correto(s)

- a) I e II
- b) I



- c) II
- d) II e III
- e) III



## Plano de Aula

### 1 Código e nome da disciplina

ARA0075 PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

### 2 Semana/Tema

Semana 4: Tema - 1. INTRODUÇÃO À PROGRAMAÇÃO OO EM JAVA

### 3 Objetivos

Organizar e configurar um ambiente de desenvolvimento em Java, conhecendo os principais componentes da arquitetura e utilizando os principais ambientes de desenvolvimento integrados para desenvolver aplicações na linguagem Java.

### 4 Tópicos

1.4 AMBIENTE DE DESENVOLVIMENTO EM JAVA

### 5 Procedimentos de ensino-aprendizagem

Para que uma linguagem de programação possa ser utilizada de forma adequada e produtiva, é necessário que ela tenha um ambiente preparado e se possível, um desenvolvimento integrado (IDE) que permita ao programador ter acesso a algumas funcionalidades como um sistema que autocompleta um comando, um sistema que permita testar o software, organizar suas versões, integrar a um banco de dados entre outras funcionalidades importantes. Nesta aula o professor deve abordar alguns ambientes de desenvolvimento bem como suas funcionalidades. Como sugestão, segue o roteiro abaixo:

#### - Situação-problema:

Um programador foi contratado para desenvolver um sistema de frente de caixa para uma farmácia. Foi escolhida a linguagem Java para a implementação do referido sistema. No entanto, o programador comprou um novo computador para utilizar como ferramenta de trabalho. Quais os passos que ele deverá seguir para configurar adequadamente o ambiente de programação? Quais ambientes de desenvolvimento ele tem disponível para desenvolver aplicações em Java?

#### - Metodologia:

O professor deve fazer uma revisão sobre linguagens interpretadas, após isso deve apresentar as diferenças entre a JRE e a JDK e a necessidade da instalação de ambas. Após o download e instalação, o professor deve apresentar o conceito de variáveis de ambiente (JAVA\_HOME e CLASSPATH) e como configurá-las adequadamente (será utilizado o sistema operacional Windows). Nesse ponto o Professor deve testar, utilizando o prompt de comando, se as variáveis foram configuradas adequadamente. Em seguida o professor deve apresentar as principais IDEs do mercado que são utilizadas pelos desenvolvedores em Java (com ênfase no Netbeans e Eclipse).

- Atividade verificadora de aprendizagem:

O professor deve solicitar aos alunos que realizem uma pesquisa com relação as IDEs Netbeans e Eclipse. O resultado da pesquisa deve apontar as similaridades e diferenças entre elas e seu percentual de uso no mercado e esse resultado deve ser apresentado.

## 6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual e ambiente de programação Netbeans e Eclipse instalados nos computadores. Além disso os computadores devem estar liberados para que os alunos possam proceder tanto a instalação da JRE e JDK como a criação das variáveis de ambiente.

## 7 Leitura específica

[1] FURGERI, Sérgio. Java 8 - Ensino Didático: Desenvolvimento e Implementação de Aplicações. 1ª Ed.. São Paulo: Érica, 2015. Capítulo 1.

Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788536519340/>

## 8 Aprenda +

Leia o artigo:

Preparação do ambiente para desenvolvimento em Java: Disponível em:

<https://www.devmedia.com.br/preparacao-do-ambiente-para-desenvolvimento-em-java/25188>

Atividade Autônoma Aura:

1) Uma das vantagens de linguagens interpretadas é que os programas que são desenvolvidos nelas podem executar em qualquer sistema operacional para qual haja suporte. Tomando como exemplo a linguagem Java, relacione as colunas com a explicação sobre:

I - JRE

II - JVM

III - JDK

( ) O ambiente de execução Java, é o mínimo que você precisa ter instalado para poder rodar um aplicativo Java

( ) Possui todo o ambiente necessário para desenvolver e executar aplicativos em Java

( ) Máquina virtual implementada pela JRE.

a) I, II, III

b) II, I, III

c) III, I, II

d) I, III, II

e) II, III, I

2) A preparação de um ambiente de desenvolvimento é parte essencial dos preparativos para que um programador possa utilizar uma linguagem de programação. Algumas linguagens necessitam que compiladores sejam instalados para um determinado sistema operacional, outras necessitam que máquinas virtuais sejam instaladas. No caso da linguagem Java, é necessário que o desenvolvedor instale o Kit de desenvolvimento para o sistema operacional no qual irá desenvolver a aplicação, além disso ele necessitará configurar algumas variáveis de ambiente que serão utilizadas pelo sistema

operacional para encontrar a máquina virtual. A partir dos seus conhecimentos sobre variáveis de ambiente, relacione as colunas com as explicações sobre:

I - JAVA\_HOME

II - PATH

III - CLASSPATH

( ) Indica o diretório onde está instalado o JDK

( ) Indica o caminho onde estão as bibliotecas java - pasta lib

( ) Indica onde se encontram os arquivos executáveis do jdk - pasta bin

a) I, II, III

b) II, I, III

c) III, I, II

d) I, III, II

e) II, III, I



## Plano de Aula

### 1 Código e nome da disciplina

ARA0075 PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

### 2 Semana/Tema

Semana 5: Tema - 2. APROFUNDAMENTO DE HERANÇA E POLIMORFISMO EM JAVA (ATIVIDADE PRÁTICA SUPERVISIONADA)

### 3 Objetivos

- Executar o particionamento e a agregação de classes para que partes menores possam ser reaproveitadas em outros projetos permitindo melhorar a manutenção e legibilidade do projeto.

### 4 Tópicos

2.1 HIERARQUIA DE HERANÇA EM JAVA

### 5 Procedimentos de ensino-aprendizagem

Nesta aula, estaremos conectados com o conteúdo digital. O aluno explora e estuda, previamente, o conteúdo digital disponível em seu ambiente virtual.

### 6 Recursos didáticos

A aula será realizada no ambiente virtual de aprendizagem.

### 7 Leitura específica

O aluno deverá consultar a bibliografia proposta no tema.

### 8 Aprenda +

O aluno deverá aprofundar os seus estudos navegando no explore + disponível no tema digital.





## Plano de Aula

### 1 Código e nome da disciplina

ARA0075 PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

### 2 Semana/Tema

Semana 6: Tema - 2. APROFUNDAMENTO DE HERANÇA E POLIMORFISMO EM JAVA (ATIVIDADE PRÁTICA SUPERVISIONADA)

### 3 Objetivos

- Apresentar o conceito de encapsulamento e sua importância para proteger a complexidade e o acesso a variáveis e métodos de classes. o mecanismo de herança, de forma a reaproveitar códigos já escritos.

### 4 Tópicos

2.2 PRINCIPAIS MÉTODOS DE OBJETOS EM JAVA

### 5 Procedimentos de ensino-aprendizagem

Nesta aula, estaremos conectados com o conteúdo digital. O aluno explora e estuda, previamente, o conteúdo digital disponível em seu ambiente virtual.

### 6 Recursos didáticos

A aula será realizada no ambiente virtual de aprendizagem.

### 7 Leitura específica

O aluno deverá consultar a bibliografia proposta no tema.

### 8 Aprenda +

O aluno deverá aprofundar os seus estudos navegando no explore + disponível no tema digital.







## Plano de Aula

### 1 Código e nome da disciplina

ARA0075 PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

### 2 Semana/Tema

Semana 7: Tema - 2. APROFUNDAMENTO DE HERANÇA E POLIMORFISMO EM JAVA (ATIVIDADE PRÁTICA SUPERVISIONADA)

### 3 Objetivos

- Utilizar classes abstratas para definir um comportamento padrão para um grupo de classes com o objetivo de padronizar o desenvolvimento do sistema.

### 4 Tópicos

2.3 POLIMORFISMO EM JAVA

### 5 Procedimentos de ensino-aprendizagem

Nesta aula, estaremos conectados com o conteúdo digital. O aluno explora e estuda, previamente, o conteúdo digital disponível em seu ambiente virtual.

### 6 Recursos didáticos

A aula será realizada no ambiente virtual de aprendizagem.

### 7 Leitura específica

O aluno deverá consultar a bibliografia proposta no tema.

### 8 Aprenda +

O aluno deverá aprofundar os seus estudos navegando no explore + disponível no tema digital.





## Plano de Aula

### 1 Código e nome da disciplina

ARA0075 PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

### 2 Semana/Tema

Semana 8: Tema - 2. APROFUNDAMENTO DE HERANÇA E POLIMORFISMO EM JAVA (ATIVIDADE PRÁTICA SUPERVISIONADA)

### 3 Objetivos

Especificar interfaces definindo atributos, métodos e assinatura de métodos, com o objetivo de definir um contrato na qual uma classe se compromete a fornecer o comportamento publicado.

### 4 Tópicos

2.4 CRIAÇÃO E O USO DE INTERFACES EM JAVA

### 5 Procedimentos de ensino-aprendizagem

Nesta aula, estaremos conectados com o conteúdo digital. O aluno explora e estuda, previamente, o conteúdo digital disponível em seu ambiente virtual.

### 6 Recursos didáticos

A aula será realizada no ambiente virtual de aprendizagem.

### 7 Leitura específica

O aluno deverá consultar a bibliografia proposta no tema.

### 8 Aprenda +

O aluno deverá aprofundar os seus estudos navegando no explore + disponível no tema digital.





## Plano de Aula

### 1 Código e nome da disciplina

ARA0075 PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

### 2 Semana/Tema

Semana 9: Tema - 3. IMPLEMENTAÇÃO DE TRATAMENTO DE EXCEÇÕES EM JAVA

### 3 Objetivos

Apresentar o conceito de exceção e sua estrutura, para que um programa possa tratar e se recuperar de erros durante sua execução.

### 4 Tópicos

3.1 TIPOS DE EXCEÇÕES  
3.2 CLASSE EXCEPTION

### 5 Procedimentos de ensino-aprendizagem

O tratamento de erros que possam ocorrer durante a execução de um programa é essencial para que um sistema não para a sua execução, para isso, é necessário que a exceção seja tratada de forma adequada. Para esta aula, sugere-se que o professor siga o seguinte roteiro.

#### - Situação-problema:

Um sistema de transações bancárias precisa ser muito robusto pois milhares e até milhões de pessoas fazem sua utilização diariamente. Esse sistema está conectado um banco de dados que contém as informações dos correntistas e suas transações. Imagine que em um determinado instante, há a perda de conexão desse sistema com o banco de dados e não houvesse tratamento para esse tipo de intercorrência. Nesse caso o sistema simplesmente cairia, pois seria lançada uma exceção que não possuiria tratamento, fazendo com que todos os correntistas tivessem seu acesso ao sistema interrompido. Como resolver esse tipo de situação?

#### - Metodologia:

A aula inicia com a apresentação do conceito de exceções e a dinâmica do seu funcionamento. O professor deve abordar a classe `java.lang.Exception` abordando sua hierarquia. Em seguida, o professor deve explicar a estrutura `try-catch-finally`, explicitando a função de cada um dos blocos. Nesse ponto, o professor deve fazer um exemplo onde um programa não tenha tratamento de exceção e dessa forma terá seu funcionamento interrompido, em seguida, o professor realiza o tratamento da exceção e demonstra que o programa continua funcionando após o devido tratamento. Como exemplo, o professor pode testar a inserção de objetos em um array de tamanho fixo.

#### - Atividade verificadora de aprendizagem:

Um sistema de vendas deve inserir produtos em um vetor de tamanho fixo (10 posições). Crie uma classe Produto que possua os atributos, nome, descrição e preço. Crie um vetor de 10 posições e simule inserções e retiradas nesse vetor. Faça inserções que extrapolem o tamanho do vetor e retiradas de elementos que não existem, observe a saída do seu programa. Posteriormente, implemente um tratamento de exceção que trate os problemas observados de forma a permitir com que o programa continue funcionando.

## 6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual e ambiente de programação Netbeans e Eclipse instalados nos computadores. Além disso os computadores devem estar liberados para que os alunos possam proceder tanto a instalação da JRE e JDK como a criação das variáveis de ambiente.

## 7 Leitura específica

[1] FURGERI, Sérgio. Java 8 - Ensino Didático: Desenvolvimento e Implementação de Aplicações. Capítulo 3, Seção 3.2 ? Orientação a Objetos. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788536519340/cfi/28!/4/2@100:0.00>

## 8 Aprenda +

Leia o artigo:

- 11 Erros que desenvolvedores Java cometem quando usam Exceptions. Disponível em: <https://www.oracle.com/br/technical-resources/article/java/erros-java-exceptions.html>

Assista ao vídeo:

- Exceptions: try, catch. Disponível em: <https://www.youtube.com/watch?v=ld2C4GcAtsg>

Atividade Autônoma Aura:

Questão 1: Estamos no auge da pandemia de covid 19 e ainda longe de uma solução definitiva para o problema. A principal arma para enfrentar essa situação é informação. Vamos supor que um programador está desenvolvendo um software que irá realizar a coleta das mortes diárias por covid 19 que ocorrem em uma cidade. Todo dia ele insere de forma fixa, o dado de mortes. Supondo que até o momento há apenas seis amostras. Abaixo segue um trecho do programa desenvolvido.

```
public static void main(String[] args) {  
  
    int mortesCovid [] = {2, 35, 57 ,80 , 28, 31};  
  
    float soma = 0;  
    for (int i = 0; i < mortesCovid.length; i++){  
        soma = soma +mortesCovid[i];  
    }  
    float media = soma/mortesCovid.length;  
    System.out.println("A média de mortes é: "+media);  
}
```

```
System.out.println(mortesCovid[6]);  
}
```

Qual exceção lançada caso esse programa fosse executado?

- a) Exception
- b) ArrayIndexOutOfBoundsException
- c) IllegalAccessException
- d) IndexOutOfBoundsException
- e) RuntimeException

Questão 2 - Em uma implementação para gerenciar uma carteira de criptomoedas usando a linguagem Java, considere que um método retirar( ) lança uma exceção Quantidade Insuficiente quando o quantidade de criptomoeda for menor do que a quantidade solicitada. Nesta situação, ao se definir o método retirar ), para identificar o nome da exceção que pode ser lançada, qual palavra chave da linguagem deve-se usar ?

- a) try
- b) static
- c) throws
- d) catch
- e) finally



## Plano de Aula

### 1 Código e nome da disciplina

ARA0075 PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

### 2 Semana/Tema

Semana 10: Tema - 3. IMPLEMENTAÇÃO DE TRATAMENTO DE EXCEÇÕES EM JAVA

### 3 Objetivos

Apresentar o conceito de exceção e sua estrutura, para que um programa possa tratar e se recuperar de erros durante sua execução.

### 4 Tópicos

3.3 SINALIZAR, LANÇAR, RELANÇAR E TRATAR EXCEÇÕES

### 5 Procedimentos de ensino-aprendizagem

A sinalização, o lançamento e o tratamento de exceções em Java é a maneira adequada para informar que houve um comportamento anormal do programa durante sua execução. Java apresenta uma estrutura bem definida para a realizar o tratamento de exceções. Como sugestão, o professor pode seguir o roteiro abaixo.

#### - Situação-problema:

Um sistema de busca de preços na web faz a varredura de diversos sites para apresentar os valores desses produtos em diversas lojas. Suponha que no momento do acesso de um certo site, o mesmo encontra-se fora do ar ou seu endereço é inválido. Que estruturas o programador Java deverá utilizar para sinalizar, lançar e tratar as possíveis exceções que esse problema pode gerar sem que o sistema para de funcionar?

#### - Metodologia:

A aula inicia com a explicação sobre o mecanismo de captura e tratamento de exceções a partir do bloco try-catch-finally e das instruções throws/throw explicitando suas diferenças. Em seguida o professor deve demonstrar como um programador pode criar a sua própria exceção a partir da herança da Classe Exception.

#### - Atividade verificadora de aprendizagem:

Suponha que o método "saca" da classe Conta vai ser reescrito de forma a lançar uma exceção criada por você, cuja classe é ContaExcecao (extends Exception). A exceção é lançada sempre que o saldo da conta for inferior ao valor sacado. Implemente a classe ContaExcecao. Implemente o método saca que lança a exceção. O exercício deve ser resolvido durante a aula.



## 6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual e ambiente de programação Netbeans e Eclipse instalados nos computadores. Além disso os computadores devem estar liberados para que os alunos possam proceder tanto a instalação da JRE e JDK como a criação das variáveis de ambiente.

## 7 Leitura específica

[1] FURGERI, Sérgio. Java 8 - Ensino Didático: Desenvolvimento e Implementação de Aplicações. Capítulo 3, Seção 3.2 - Orientação a Objetos. Disponível em:  
<https://integrada.minhabiblioteca.com.br/#/books/9788536519340/cfi/28!/4/2@100:0.00>

## 8 Aprenda +

Leia o artigo:

- 11 Erros que desenvolvedores Java cometem quando usam Exceptions. Disponível em:  
<https://www.oracle.com/br/technical-resources/article/java/erros-java-exceptions.html>

Assista ao vídeo:

- Exceptions: stacktrace e throws. Disponível em: <https://www.youtube.com/watch?v=CW7DER5TqLM>

Assista ao vídeo:- Exceptions: Tipos de erros em Java. Disponível em:  
<https://www.youtube.com/watch?v=5R8pw1V2H-g>

Assista ao vídeo:- Exceptions: Criando sua própria exceção. Disponível em:  
<https://www.youtube.com/watch?v=Xt0CpwJT5yY>

Atividade Autônoma Aura:

Questão 1) As exceções são recursos muito importantes que são utilizados por algumas linguagens de programação. A linguagem Java possui uma forma simples de capturar e lançar exceções. A seguir são feitas algumas afirmações sobre efeitos de uma exceção:

I - Nenhum valor de retorno é retornado

II - O método de lançamento de exceção termina prematuramente

III - Um método pode capturar uma exceção para tentar recuperar-se do erro ou enviar uma mensagem mais amigável para o usuário.

Qual(s) afirmativa(s) é (são) verdadeira(s)?

a) I e II

b) I e III

c) I

d) II e III

e) Todas as afirmativas estão correta

Questão 2) É possível que um método possa gerar múltiplas exceções, nesse caso, como por exemplo, em um método de inserção de dados sobre uma pessoa em um banco de dados. Nesse caso, pode haver problemas com conexão com o banco, com a SQL de inserção, com incompatibilidade dos dados inseridos entre outros. Em relação ao tratamento de múltiplas exceções, julgue os itens a seguir:

I - As cláusulas catch são verificadas na ordem em que são escritas

II - Java não possui suporte a tratamento de múltiplas exceções

III - Somente uma exceção poderá ser lançada

Está(ão) correta(s):

- a) I
- b) I e III
- c) II e III
- d) II
- e) I, II e III



## Plano de Aula

### 1 Código e nome da disciplina

ARA0075 PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

### 2 Semana/Tema

Semana 11: Tema - 4. PROGRAMAÇÃO PARALELA EM JAVA: THREADS

### 3 Objetivos

- Apresentar os conceitos de processamento paralelo e de concorrência e sua aplicação em situações que demandam que processos sejam executados "ao mesmo tempo".

### 4 Tópicos

4.1 O CONCEITO DE THREADS E SUA IMPORTÂNCIA PARA O PROCESSAMENTO PARALELO

### 5 Procedimentos de ensino-aprendizagem

O desenvolvimento de processadores com múltiplos núcleos foi uma solução encontrada para a necessidade atual dos sistemas conseguirem executar tarefas de forma paralela. Nessa aula o professor deve abordar o conceito de threads e sua importância para o processamento paralelo. Como sugestão o professor pode seguir o roteiro abaixo:

- Situação-problema:

Um programador deve desenvolver um software comercial que mantém atualizado os dados de estoque de produtos de uma empresa. Esse sistema deve realizar conexões com servidores via redes, acessar banco de dados e processar requisições web. No entanto, todas essas operações são importantes e devem ser executadas de forma simultânea em um servidor. Como podemos fazer com que todos esses processos sejam executados de forma paralela?

- Metodologia:

A aula inicia com o professor apresentando o conceito de paralelismo e a diferença entre paralelismo e concorrência seguido da sua importância, em seguida, deve ser apresentada a diferença entre processos e threads e como uma thread seria executada em um computador com um processador de um único núcleo e um processador de vários núcleos. Em seguida deve ser apresentado os estados possíveis que uma thread pode estar e as possíveis transições entre os estados.

- Atividade verificadora de aprendizagem: Um programador resolveu implementar usando threads um novo servidor web. O servidor é um processo daemon que possui uma thread bloqueada a espera de requisições http que cheguem via rede. Ao receber uma requisição, esse thread se desbloqueia, cria uma nova thread para atender a requisição que chegou e volta a se bloquear até que uma nova

requisição chegue. Essa solução baseada em threads funciona para máquinas monoprocessadas e para máquinas multiprocessadas? Justifique sua resposta. O exercício deve ser respondido durante a aula.

## 6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual e ambiente de programação Netbeans e Eclipse instalados nos computadores. Além disso os computadores devem estar liberados para que os alunos possam proceder tanto a instalação da JRE e JDK como a criação das variáveis de ambiente.

## 7 Leitura específica

[1] SCHILDT, Herbert. Java para Iniciantes. 6ª Ed.. Porto Alegre: Bookman, 2015. Capítulo 11 - Programação com várias threads. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788582603376/cfi/381>

## 8 Aprenda +

Assista ao vídeo:

- Thread (entenda como sua aplicação funciona) // Dicionário do Programador. Disponível em: <https://www.youtube.com/watch?v=xNBMNKjpJzM>

Atividade Autônoma Aura:

Questão 1) Os sistemas computacionais modernos têm evoluído para enfrentar as novas demandas por recursos computacionais. Uma das estratégias utilizadas é o desenvolvimento de sistemas multitarefas e sistemas multiprocessados. A multitarefa pode trabalhar de forma diversa em sistemas monoprocessados e multiprocessados. A partir dessas diferentes formas de implementação da multitarefa, julgue os itens que se seguem:

I - Concorrência e paralelismo possuem o mesmo sentido técnico.

II - Concorrência trata da execução sequencial e disputada de um conjunto de tarefas independentes.

III - Paralelismo trata da execução de mais uma tarefa ao mesmo tempo, um por vez, de forma simultânea, a depender da quantidade de núcleos disponíveis.

Está(ão) correta(s)?

- a) I
- b) I e II
- c) II e III
- d) I e III
- e) nenhuma

Questão 2) Os threads definem como um processador funciona, recebendo e executando instruções. Isso acontece muito rapidamente e passa a sensação de que as ações são simultâneas. Portanto, uma CPU com um thread tem apenas uma linha de trabalho e realiza uma ação por vez. Logo, processadores multithread são mais vantajosos, já que dão a possibilidade de operar em diversas frentes ao mesmo tempo. Disponível em: <https://www.techtudo.com.br/noticias/2019/01/o-que-sao-threads-e-para-que-servem-em-um-processador.ghtml>

Em relação as threads, julguem os itens que se seguem:

I - Thread é uma abstração que permite que uma aplicação execute mais de um trecho de código

(método) simultaneamente.

II - Multithreading refere-se à habilidade de um SO em suportar múltiplas threads de execução em vários processos.

III - A JVM suporta um processo com várias threads.

Está(ão) correta(s)?

- a) I
- b) I e II
- c) II e III
- d) I e III
- e) nenhuma



## Plano de Aula

### 1 Código e nome da disciplina

ARA0075 PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

### 2 Semana/Tema

Semana 12: Tema - 4. PROGRAMAÇÃO PARALELA EM JAVA: THREADS

### 3 Objetivos

Aplicar os conceitos de programação paralela para resolver um problema que exija que atividades sejam executadas nessa arquitetura.

### 4 Tópicos

4.2 IMPLEMENTANDO THREADS EM JAVA

### 5 Procedimentos de ensino-aprendizagem

Na aula anterior foram apresentados os conceitos de programação paralela e concorrente juntamente com o conceito de Thread. Nesse aula o professor deve apresentar na prática, como criar threads a partir da interface Runnable e também a partir da herança da classe Thread introduzindo sua sintaxe e principais métodos. Como sugestão, segue o roteiro abaixo:

#### - Situação-problema:

Um programador está desenvolvendo um app para o Android utilizando a linguagem Java. O App é utilizado para automação residencial, dessa forma ele envia e recebe dados para um microcontrolador Arduino no qual estão conectados sensores de temperatura e emissores de infravermelhos que tem o objetivo de controlar a temperatura de uma residência. Esse app exibe em tempo real a temperatura dos vários cômodos da casa e precisa enviar dados para o Arduino realizar o controle da temperatura. Nesse caso, a interface gráfica precisa ser atualizada ao mesmo tempo em que comandos são enviados para o Arduino. Como essa operação pode ser feita levando em consideração que o processo de atualização da interface gráfica precisa executar o tempo todo?

#### - Metodologia:

A aula inicia com o professor apresentando as possíveis formas de criação de uma Thread: a partir da interface Runnable e a partir da herança da classe Thread, apresentando, em seguida, os métodos run(), start(), sleep(). Nesse ponto, o professor pode apresentar um exemplo simples, como sugestão o exemplo que está na página 366 do livro sugerido na leitura específica. Em seguida, o professor deve demonstrar como criar múltiplas threads, determinando quando uma thread termina e como priorizar uma thread.

#### - Atividade verificadora de aprendizagem:

Um estudante deseja criar um simulador de corrida, para isso ele cria uma Classe Piloto que é uma subclasse da classe Thread. Essa classe possui o atributo nome e no seu método run() é implementada uma simulação de corrida. Como por exemplo:

```
System.out.println("****LARGADA ****");
for (int i=1;i<4;i++){// três voltas
for (int cont = 0; cont < 100000; cont++) {
};
System.out.println("Volta "+ i + ": "+ nome);
}
System.out.println(nome + " -> Final da Corrida!");
```

Em seguida, ele implementa um classe chamada Corrida que deve simular uma corrida com 4 Pilotos: Felipe Massa, Hamilton, Schumacher e Sena. Ao final o programa deve exibir quando cada piloto completa sua volta e quando ele termina a corrida.

## 6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual e ambiente de programação Netbeans e Eclipse instalados nos computadores. Além disso os computadores devem estar liberados para que os alunos possam proceder tanto a instalação da JRE e JDK como a criação das variáveis de ambiente.

## 7 Leitura específica

[1] SCHILDT, Herbert. Java para Iniciantes. 6ª Ed.. Porto Alegre: Bookman, 2015. Capítulo 11 - Programação com várias threads. Disponível em:  
<https://integrada.minhabiblioteca.com.br/#/books/9788582603376/cfi/381>

## 8 Aprenda +

Assista ao vídeo:

- Criando Threads + método start, run e sleep Disponível em: <https://www.youtube.com/watch?v=v5l30QMKv6c>

Atividade Autônoma Aura:

Questão 1)(COPESE-UFPI, 2017) As unidades concorrentes em Java são métodos denominados run, cujo código pode estar em execução simultânea com outros métodos (de outros objetos) e com o método main. Uma das maneiras de definir uma classe com um método run é definir uma subclasse da classe Thread predefinida e substituir o método run. Sobre a classe Thread da linguagem Java, para cada afirmativa abaixo, informe se é verdadeira (V) ou falsa (F). Em seguida, marque a opção que corresponde à sequência CORRETA.

( ) A classe provê diversos métodos para controle de execução de threads, sendo o método start responsável por iniciar a execução de thread.

( ) O método sleep é utilizado para forçar um método a atrasar sua execução até que o método run de outra thread tenha completado sua execução.

( ) O método yield, que não possui parâmetros, é um pedido de thread de execução para entregar o processador voluntariamente.

- a) F-V-V
- b) V-V-F
- c) F-F-V
- d) V-F-F
- e) V-F-V

Questão 2) Um programador precisa desenvolver uma aplicação em que um usuário possa imprimir um pdf e a barra de progresso que indica a porcentagem da impressão deve ser atualizada ao mesmo tempo. Abaixo segue um trecho de código desenvolvido por esse programador:

```
public class MeuPrograma {  
  
    public static void main (String[] args) {  
  
        GeraPDF gerapdf = new GeraPDF();  
  
        Thread threadDoPdf = new Thread(gerapdf);  
  
        threadDoPdf.start();  
  
        BarraDeProgresso barraDeProgresso = new BarraDeProgresso();  
  
        Thread threadDaBarra = new Thread(barraDeProgresso);  
  
        threadDaBarra.start();  
  
    }  
  
}
```

Para que o código apresentado seja funcionar, que método deve ser implementado nas classes GeraPDF e Barra DeProgresso supondo que elas implementam a interface Runnable?

- a) start()
- b) run()
- c) init()
- d) execute()
- e) do()





## Plano de Aula

### 1 Código e nome da disciplina

ARA0075 PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

### 2 Semana/Tema

Semana 13: Tema - 4. PROGRAMAÇÃO PARALELA EM JAVA: THREADS

### 3 Objetivos

- Coordenar as atividades de duas ou mais Threads em situações nas quais eles precisam acessar um recurso compartilhado que só pode ser utilizado por uma Thread de cada vez.

### 4 Tópicos

4.3 SINCRONIZAÇÃO DE THREADS EM JAVA

### 5 Procedimentos de ensino-aprendizagem

Na aula anterior foi apresentado como criar e utilizar Threads para solucionar problemas que precisam ser paralelizados. Nesse aula o professor deve apresentar como sincronizar threads em situações na qual elas compartilham recursos ou quanto uma thread está esperando um evento que é causado por outra Thread. Como sugestão, segue o roteiro abaixo:

#### - Situação-problema:

Uma loja virtual possui vários produtos a venda os quais podem ser comprados pelos consumidores que estão visitando a loja. Imagine que em um determinado instante de tempo um consumidor "A" selecionou um jogo cujo estoque é de uma unidade. Ele insere esse produto no seu carrinho de compras e continua a navegação pelo site em busca de outros produtos. Nesse mesmo instante, o consumidor "B" também seleciona o mesmo jogo, no entanto, ele vai direto para a fase de pagamento, fazendo com que o produto seja retirado do estoque. Algum tempo depois o consumidor "A", após escolher outros produtos, realiza o pagamento. No entanto, o jogo que o consumidor "A" escolheu não está mais em estoque e o sistema da loja vendeu mesmo assim. Como resolver essa situação?

#### - Metodologia:

O Professor inicia a aula apresentando a situação do problema que ilustra a necessidade de sincronização de threads. Em seguida apresenta a sincronização de métodos e a utilização da palavra-chave synchronized. Em seguida, deve abordar a comunicação entre threads utilizando notify(), wait() e notifyAll() e em sequência apresenta a suspensão, retomada e encerramento de threads utilizando os métodos suspend(), resume() e stop(). A aula deve se finalizar com a proposição da atividade verificadora de aprendizagem.

#### - Atividade verificadora de aprendizagem:

Considere um banco com uma só conta. Nessa conta é possível efetuar as operações de depósito e saque. Cada uma dessas operações tem um parâmetro inteiro que é o valor a ser depositado ou sacado. A operação depositar consiste em somar o valor ao saldo atual enquanto a operação de saque consiste em verificar se há saldo suficiente para efetuar a retirada e em caso afirmativo, o valor deve ser subtraído do saldo. Considere que as operações podem ser efetuadas simultaneamente por vários processos.

a) Codifique um programa em Java com as operações sacar e depositar.

b) Discorra sobre a necessidade de sincronização entre os vários processos que executam as operações.

Os alunos devem ser divididos em pequenos grupos e ao final da aula, cada grupo apresenta sua implementação.

## 6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual e ambiente de programação Netbeans e Eclipse instalados nos computadores. Além disso os computadores devem estar liberados para que os alunos possam proceder tanto a instalação da JRE e JDK como a criação das variáveis de ambiente.

## 7 Leitura específica

[1] SCHILDT, Herbert. Java para Iniciantes. 6ª Ed.. Porto Alegre: Bookman, 2015. Capítulo 11 - Programação com várias threads. Disponível em:

<https://integrada.minhabiblioteca.com.br/#/books/9788582603376/cfi/381>

## 8 Aprenda +

Assista ao vídeo:

- Threads: métodos e blocos sincronizados (synchronized) Disponível em:  
<https://www.youtube.com/watch?v=uR0zWNqJ5a8>

Atividade Autônoma Aura:

Questão 1)(CESGRANSRIO-BNDS, 2011) Os usuários de um sistema WEB, desenvolvido em JAVA, reclamam de erros nos dados consultados. A equipe técnica, ao analisar a situação, concluiu que determinado método M está sendo invocado, simultaneamente, por diferentes threads.

Considerando-se que não há manipulação de locks no restante do sistema, o que pode ser feito para que M seja executado, em dado momento, somente por uma thread?

a) Ajustar o firewall de borda para permitir uma conexão por IP.

b) Implementar connection pooling no acesso ao banco de dados.

c) Utilizar a keyword lockable no bloco de M.

d) Declarar o método M como synchronized.

e) Reescrever M em Assembly, necessariamente.

Questão 2)(COSEAC-UFF, 2015) Na linguagem Java, uma thread executável pode entrar em um determinado estado por um intervalo de tempo especificado, retornando ao estado executável quando for notificada por outra thread ou quando esse intervalo expirar. Essa situação caracteriza o estado:

- a) Cancelado
- b) Bloqueado
- c) Espera sincronizada
- d) Novo
- e) Terminado



## Plano de Aula

### 1 Código e nome da disciplina

ARA0075 PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

### 2 Semana/Tema

Semana 14: Tema - 5. INTEGRAÇÃO COM BANCO DE DADOS EM JAVA

### 3 Objetivos

Implementar sistemas que possam realizar operações com bando de dados utilizando a API JDBC.

### 4 Tópicos

5.1 A API JDBC

### 5 Procedimentos de ensino-aprendizagem

Nessa aula o professor deverá explicar a necessidade de desenvolver aplicações que manipulem dados de forma persistente. Em seguida ele deve apresentar a API JDBC e seus principais métodos. Como sugestão, o professor deve seguir o roteiro abaixo:

#### - Situação-problema:

Um empresário contratou uma empresa de desenvolvimento de software para gestão de estoques de um supermercado. As informações dos produtos precisam persistir em um Banco de Dados. Como podemos manipular esses dados utilizando a linguagem de programação Java?

#### - Metodologia:

O Professor inicia a apresentando a API JDBC e como ela pode ser utilizada para manipular os principais banco de dados disponíveis no mercado por meio dos drivers específicos. Em seguida o professor deve criar uma conexão com um Banco de Dados a sua escolha (sugere-se o MySQL) e realizar as principais operações de banco (CRUD).

#### - Atividade verificadora de aprendizagem:

Tomando como base o CRUD que o professor desenvolveu durante a sala de aula, os alunos devem ser divididos em grupos e devem criar uma agenda telefônica que armazene os nomes e telefones de pessoas. As operações que devem ser implementadas são de inserção, deleção, atualização e listagem. Esta é a Atividades Acadêmicas Avaliativas que vale 5 pontos que compõe a nota da AV2 desta disciplina, e a mesma deve ser entregue pelo Sava, MS Teams, ou similar até a data da AV2.

### 6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual e ambiente de programação Netbeans, Eclipse, WAMP e Workbench instalados nos computadores. Além disso os computadores devem estar liberados para que os alunos possam proceder tanto a instalação da JRE e JDK como a criação das variáveis de ambiente.

## 7 Leitura específica

[1] SCHILDT, Herbert. Java para Iniciantes. 6ª Ed. Porto Alegre: Bookman, 2015. Capítulo 12 - Programação com várias threads. Disponível em:  
<https://integrada.minhabiblioteca.com.br/#/books/9788582603376/cfi/381>

## 8 Aprenda +

Leia o tutorial:

- Tutorial do aplicativo CRUD para a plataforma Netbeans. Disponível em:  
[https://platform.netbeans.org/tutorials/nbm-crud\\_pt\\_BR.html](https://platform.netbeans.org/tutorials/nbm-crud_pt_BR.html)

Atividade Autônoma Aura:

Questão 1)(FCC-TRT, 2013) (Adaptada) Considere, abaixo, os métodos encontrados em classes de aplicações Java que acessam banco de dados.

Método 1:

```
public int insert (int varId, String varNome, double varRenda){  
    int retorno;  
    Class.forName("com.mysql.jdbc.Driver");  
    Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/bd","root","1234");  
    PreparedStatement st = conn.prepareStatement("insert into cliente (id, nome, renda) values (?, ?, ?)");  
    st.setInt(1, varId);  
    st.setString(2, varNome);  
    st.setDouble(3, varRenda);  
    retorno = st.executeUpdate();  
    return retorno;  
}
```

Método 2:

```
public int insert (int varId, String varNome, double varRenda){  
    int retorno;  
    Class.forName("com.mysql.jdbc.Driver");  
    Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/bd","root","1234");  
    Statement st = conn.createStatement();  
    retorno = st.executeUpdate("inset into cliente values (" + varId + " , " + varNome + " , " + varRenda + ")");  
    return retorno;  
}
```

Nas classes, nas quais estes métodos se encontram, foram importados todos os recursos necessários para a execução. O banco de dados, a tabela e o driver JDBC existem e funcionam corretamente.

É correto afirmar que:

- a) o Método 1 está incorreto, pois o método executeUpdate da interface PreparedStatement precisa receber como parâmetro a instrução SQL insert a ser executada.
- b) o Método 2 está incorreto, pois o método executeUpdate da interface Statement não pode receber parâmetros. A instrução insert passada como parâmetro nesse método deveria ser passada como parâmetro para o método createStatement da interface Connection.
- c) ambos os métodos estão corretos e executam a mesma operação, apresentando os mesmos resultados.
- d) ambos os métodos estão incorretos, pois o método presente tanto na interface Statement como na interface PreparedStatement para incluir dados na tabela do banco de dados é o método executeInsert e não executeUpdate .
- e) o Método 1 está incorreto, pois a instrução insert passada como parâmetro para o método PreparedStatement da interface Connection está incompleta. No lugar dos pontos de interrogação devem ser colocados os valores que devem ser incluídos nos campos id , nome e renda da tabela.

Questão 2 ) (CONSULPLAN-TRF, 2017) (Adaptada) A grande maioria das aplicações, independente de porte, utiliza a persistência de dados atrelada a seus sistemas. O banco de dados se faz necessário em uma aplicação não só para persistir as informações, como também é preciso nos comunicarmos com ele para recuperar, modificar e apagar informações. Portanto, podemos afirmar que o gerenciamento desses dados é de fundamental importância para o correto funcionamento da aplicação. Quando se trata de persistência de dados em Java, geralmente a forma utilizada para guardar dados é um banco de dados relacional. Para abrir uma conexão com um banco de dados, precisamos utilizar sempre um driver. A classe DriverManager é responsável por realizar essa comunicação, o drive JDBC - Java Database Connectivity é a especificação de como a linguagem Java irá se comunicar com um banco de dados. O parâmetro que é passado é do tipo String contendo URL para localizar o banco de dados que, por sua vez, contém informações para conexão com o banco de dados. Assinale a alternativa correta que contém a URL padrão para a conexão com o banco MySQL, considerando que: usuário, servidor e porta do banco serão padrão, o banco não tem senha e o nome do banco de dados é teste.

- a) ("mysql:jdbc://localhost:3306/teste","", "root").
- b) ("jdbc:mysql://localhost:3306/teste", "root", "").
- c) ("mysql:jdbc://localhost:3306/teste", "root", "").
- d) ("jdbc:mysql://localhost:3306/teste", "", "root").
- e) ("mysql-jdbc://localhost:3306/teste", "root", "").



## Plano de Aula

### 1 Código e nome da disciplina

ARA0075 PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

### 2 Semana/Tema

Semana 15: Tema - 5. INTEGRAÇÃO COM BANCO DE DADOS EM JAVA

### 3 Objetivos

Utilizar o mapeamento objeto relacional para obter o modelo de dados relativo ao projeto de forma automatizada.

### 4 Tópicos

5.2 O MAPEAMENTO OBJETO RELACIONAL

### 5 Procedimentos de ensino-aprendizagem

Nessa aula o professor deverá apresentar princípios do mapeamento objeto relacional e sua importância para o desenvolvimento de sistema que utilizam banco de dados. Como sugestão, o professor deve seguir o roteiro abaixo:

#### - Situação-problema:

Um programador está desenvolvendo um sistema utilizando o paradigma de programação orientada a objeto. Nesse processo, ele realizou a modelagem do sistema e iniciou o seu desenvolvimento. No entanto, como o sistema terá interação com o banco de dados, um modelo de dados deve ser desenvolvido de acordo com as classes que representam os elementos do projeto. Não seria interessante a utilização de algum framework que realizasse o mapeamento dos objetos em tabelas haja vista que sua estrutura pode ser equivalente?

#### - Metodologia:

O Professor inicia a o conceito de ORM e em seguida apresenta a API JPA (Java Persistence API), explicando como criar uma entidade JPA a partir do NetBeans. Em seguida o professor deve apresentar as estratégias de geração de chaves primárias e as anotações que permitem realizar o mapeamento das cardinalidades entre as entidades. Após a criação da entidade, deve apresentar os métodos de inserção, deleção, atualização e listagem finalizando com a criação de uma classe que teste a execução dos referidos métodos.

#### - Atividade verificadora de aprendizagem:

Os alunos devem ser divididos em equipes e então devem criar um cadastro de funcionários. Inicialmente devem criar a base de dados FUNCIONARIOS\_DB no MySQL, em seguida devem criar a tabela FUNCIONÁRIO:

```
CREATE TABLE FUNCIONARIO(  
ID_FUNCIONARIO SERIAL PRIMARY KEY AUTOINCREMENT,  
NOME VARCHAR(80) NOT NULL,  
CPF VARCHAR(14) NOT NULL,  
CATEGORIA VARCHAR(20) NOT NULL,  
SALARIO NUMERIC(8,2) NOT NULL  
);
```

Em seguida, os alunos devem então realizar o mapeamento e implementar os métodos de inserção, deleção atualização e listagem utilizando quaisquer das opções disponíveis pelo Framework. Ao final, as equipes devem apresentar a sua solução.

## 6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual e ambiente de programação Netbeans, Eclipse, WAMP e Workbench instalados nos computadores. Além disso os computadores devem estar liberados para que os alunos possam proceder tanto a instalação da JRE e JDK como a criação das variáveis de ambiente.

## 7 Leitura específica

[1] MACHADO, Rodrigo Prestes; FRANCO, Márcia Hafele Islabão; BERGANOLLI, Silva de Castro. Desenvolvimento de Software III: programação de sistemas web orientada a objetos em Java. 1ª Ed. Porto Alegre: Bookman, 2016. Capítulo 4 - Programação com várias threads. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788582603710/cfi/106!/4/4@0.00:0.00>

## 8 Aprenda +

Leia o tutorial:

- Tutorial do Netbeans E-commerce: Adicionando Classes de Entidade e Beans de Sessão. Disponível em: [https://netbeans.org/kb/docs/javaee/ecommerce/entity-session\\_pt\\_BR.html](https://netbeans.org/kb/docs/javaee/ecommerce/entity-session_pt_BR.html)

Atividade Autônoma Aura:

Questão 1)(CETAP, MPC-PA, 2015) Analise o código concernente à tecnologia JPA a seguir

1. @Entity
2. //inserir o código 1 aqui
3. public class Cliente {
- 4.
5. //inserir o código 2 aqui
6. private Long id;
- 7.



8. private String temporario;

9 }

Qual das opções a seguir pode, respectivamente, substituir o comentário 1 e 2 para produzir corretamente a classe Cliente?

a) @Table(name=?cliente") / @Identificator

b) @Table(name="cCliente") / @Id

c) @Table(?cliente")/@Id

d) @Table(?clíente") / @Identificator

e) @Identificator/@Temp

Questão 2) (FUNRIO - IF-PA, 2016) Analise o seguinte código:

```
@Entity
```

```
@NamedQuery(
```

```
name=?encontreTodosOsEmpregadosPeloUltimoNome?,
```

```
query=?SELECT empregados FROM Empregado empregados WHERE empregados.ultimoNome =: ultimonome?
```

```
)
```

```
public class Empregado implements Serializable{
```

```
@Id
```

```
private int empID;
```

```
@Column(name=?SAL?, table=?EMP_SAL?)
```

```
private Long salario;
```

```
....
```

```
}
```

Em relação as annotations do JPA (Java Persistence API) do seguinte código, é correto afirmar que

a) A anotação @Entity define a tabela primária associada a entidade Empregado.

b) A anotação @Column define o método da entidade Empregado que é chamado no momento que o método de persistência do EntityManager é chamado.

c) A anotação @NamedQuery foi utilizada para associar o nome do atributo salario da Entidade empregado a coluna SAL da tabela ?EMP\_SAL?.

d) A anotação @Entity define que a classe Empregado não possui atributo de chave primária.

e) A anotação @Id é utilizada para definir que o atributo empID é a chave primária da entidade Empregado.





## Plano de Aula

### 1 Código e nome da disciplina

ARA0075 PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

### 2 Semana/Tema

Semana 16: Tema - 5. INTEGRAÇÃO COM BANCO DE DADOS EM JAVA

### 3 Objetivos

Aplicar as tecnologias e métodos vistos durante o decorrer do curso para a construção de exemplos.

### 4 Tópicos

5.3 IMPLEMENTAÇÃO DE EXEMPLOS

### 5 Procedimentos de ensino-aprendizagem

Nessa aula o professor deverá orientar os alunos no desenvolvimento de aplicações que utilizem as tecnologias e métodos que foram apresentados durante o curso de forma que eles demonstrem suas habilidades para construir soluções aplicadas a problemas reais. Como sugestão, o professor pode seguir o roteiro abaixo:

#### - Situação-problema:

Um jovem programador foi contratado para desenvolver um sistema que exige um cadastro de filmes. Quais as etapas para criação desse sistema utilizando o paradigma de orientação a objetos e persistência em banco de dados?

#### - Metodologia:

Nessa aula, o professor apresenta um sistema que deve ser implementado pelos alunos, os quais deverão ser divididos em grupos e devem ser orientados pelo professor a desenvolver a atividade sugerida como atividade verificadora de aprendizagem.

#### - Atividade verificadora de aprendizagem:

Os alunos, já divididos em grupos, devem implementar um sistema de cadastro de filmes que é sugerido no capítulo 12, página 247 de [1].

### 6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual e ambiente de programação Netbeans,

Eclipse, WAMP e Workbench instalados nos computadores. Além disso os computadores devem estar liberados para que os alunos possam proceder tanto a instalação da JRE e JDK como a criação das variáveis de ambiente.

## 7 Leitura específica

[1] FURGERI, Sérgio. Java 8 - Ensino Didático: Desenvolvimento e Implementação de Aplicações. 1ª Ed.. São Paulo: Érica, 2015. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788536519340/>. Capítulo 12. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788536519340/>.

## 8 Aprenda +

Assista ao vídeo:

-JDBC CRUD using Java and Mysql. Disponível em: <https://www.youtube.com/watch?v=xYEungntiil&t=10s> Não esqueça de ativar as legendas em Português

Atividade Autônoma Aura:

Questão 1) (NC-UFPR - COPEL, 2015) Quanto a JPA (Java Persistence API) 2.0 e seus modos de carregamento (FetchType) Lazy e Eager, identifique as afirmativas a seguir como verdadeiras (V) ou falsas (F):

- ☐ Eager é o comportamento padrão para relacionamentos muitos-para-muitos.
- ☐ É preciso cuidar do cascadeamento ao usar Eager Load, pois muitos objetos podem ser carregados desnecessariamente.
- ☐ Lazy apresenta maior consumo de processamento e rede durante a inicialização da aplicação quando comparado com Eager.
- ☐ Fazer cache de objetos instanciados via Lazy Load é geralmente desaconselhável, devido ao alto consumo de processamento.
- ☐ Essas formas de carregamento tornaram-se Deprecated na JPA 2.0.

Assinale a alternativa que apresenta a sequência correta, de cima para baixo.

- a) V - F - V - F - V.
- b) F - V - F - V - V.
- c) V - F - V - V - F.
- d) F - F - V - V - F.
- e) F - V - F - F - F.

Questão 2) (FCC- DPE-AM, 2018) Considere o fragmento de código abaixo em uma classe de acesso a dados que utiliza JDBC com objetivo de conectar a aplicação Java a um Banco de Dados PostgreSQL denominado bdaction.

```
Class.forName("org.postgresql.Driver");
```

```
Connection connection =
```

```
DriverManager.getConnection("jdbc:postgresql://127.0.0.1:5432/a0041","bdaction","a45Ab");
```

```
Statement statement = connection.createStatement();
```

Nesse fragmento,

- a) a falta da classe Driver ocasionará uma exceção do tipo SQLException.
- b) a classe Statement deve ser trocada por EntityManager, que permite acesso a métodos para execução de comando SQL.
- c) como o Banco de Dados ao qual se deseja conectar é o bdaction, ocorrerá uma exceção do tipo SQLException.
- d) a porta de comunicação com o PostgreSQL deve ser mudada de 5432 para a porta padrão, que é 336.
- e) não é necessário tratamento de exceção, pois isso é feito automaticamente pela API JDBC.