



Roteiro de Aula Prática

Linguagem Orientada a Objetos

Disciplina: Linguagem Orientada a Objetos

ROTEIRO DE AULA PRÁTICA 1

Unidade:

Aula (White Label)/Seção (KLS):

SOFTWARE

☒ Software / ☐ Acesso on-line

☐ Pago / ☒ Não Pago

Infraestrutura:

Computador;

Software IDE NetBeans com Java JDK instalado e configurado.

Descrição do software:

NetBeans é uma IDE de desenvolvimento gratuita e de código aberto. Útil no desenvolvimento de softwares utilizando as linguagens: Javascript, HTML5, PHP, C/C++ e Java. Ferramenta multiplataforma podendo ser instalada no Linux, macOS e Windows.

ATIVIDADE PRÁTICA 1

Atividade proposta:

Configurar um servidor que será necessário para que a IDE NetBeans execute os projetos desenvolvidos sobre essa ferramenta. Para essa atividade, faremos a instalação do Tomcat como servidor padrão para rodar as aplicações e para testar, você deverá “levantar” uma aplicação web simples para testar esse servidor.

Objetivos:

- Criar familiaridade com ambiente de desenvolvimento NetBeans.
- Configurar servidor Tomcat para exibir aplicações desenvolvidas em Java;
- Compreender como testar o servidor tomcat.

Procedimentos para a realização da atividade:

Para cumprir com o proposto dessa aula prática, é necessário ter previamente instalado o Netbeans e o Java JDK.

1 - Acessar a página do servidor de internet Tomcat. Sua escolha se dá por se tratar de um servidor também desenvolvido pela Apache, mesma criadora do NetBeans. O que assegurará total integração entre as ferramentas. Para isso, acesse a página do Tomcat em: <<https://tomcat.apache.org/download-native.cgi>>.

2 - Trabalharemos com a versão 8.x do servidor. Procure na página por essa versão para ter

acesso à área de download.

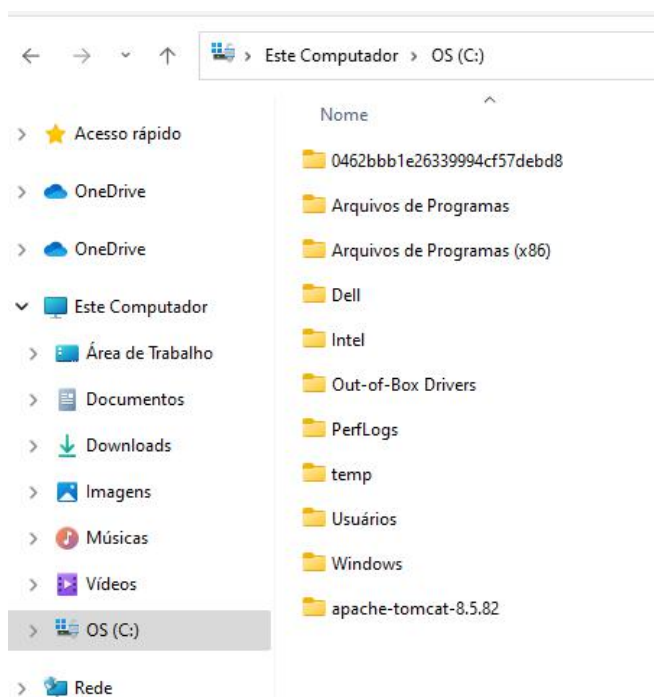
3 - Ao acessar a área de download, procure pela sessão “Core” da página. Nela, nós teremos acesso a versões distintas desse servidor. Atenção à arquitetura de sua máquina para que você não escolha a versão errada. De modo geral, a versão para arquitetura 32bits está sendo descontinuada porque essa arquitetura de processadores foi descontinuada. Muito provavelmente você utilizará a versão 64bits, mas tenha certeza. Para isso, você pode acessar a propriedade do “Meu Computador” e encontrar a informação de arquitetura de seu processador.

4 - Faça o download da versão de arquivos .zip, ou seja, faremos o download da versão zipada de arquivos, ou seja, 64-bit Windows.zip. Veja:

- Core:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
 - [32-bit Windows zip \(pgp, sha512\)](#)
 - [64-bit Windows zip \(pgp, sha512\)](#)
 - [32-bit/64-bit Windows Service Installer \(pgp, sha512\)](#)

5 - Após o download ser realizado, descompacte o arquivo em um local no seu computador. Pode ser na pasta Downloads mesmo, pois vamos mudar a localização da pasta.

6 - Agora, vamos mover essa pasta apache-tomcat-8.5.82 para o Disco Local C: de nosso computador. Vamos colocá-lo solto, como mostra a imagem a seguir:



7 - Agora, abra o NetBeans e crie um novo projeto, um projeto Java Web. Para isso, será necessário escolher essa opção na categoria Java with Ant > Java Web > Web Application.

8 - Escolha o local onde o projeto será salvo, dê um nome para o projeto e Click no botão “Next”.

9 - Na tela que surgirá, será necessário escolher o servidor Apache Tomcat or TomEE e configurar a opção “Server Location” que é a pasta que acabamos de colocar no disco local C:

Nessa mesma tela pode-se definir um usuário e senha para acessar o servidor.

10 - Na tela seguinte será exibida a versão do Java EE que está sendo utilizada e você deverá avançar, clicando no botão “Next”. Na tela seguinte, selecione o framework Spring Web MVC e clique no botão “Finish”. Ao escolher essa opção, uma estrutura em HTML será criada graças ao framework escolhido na etapa anterior. Execute a aplicação e acesse o link gerado pelo Tomcat no navegador para ver sua aplicação.

Checklist:

- 1 - Download das ferramentas e preparação do ambiente de desenvolvimento com NetBeans.
- 2 - Download do servidor Tomcat no site oficial do Apache.
- 3 - Organizar pasta do servidor no computador e iniciar o NetBeans;
- 4 - Criar um novo projeto do tipo Java Web, e realizar as etapas de configuração do servidor.
- 5 - Instalar o framework Spring MVC para construir uma projeto simples web para executar o servidor e testar a aplicação no navegador.

Resultado: Aluno, você deverá entregar:

Configuração do servidor Tomcat na versão 8.x para servidor às aplicações web desenvolvidas na IDE Netbeans. Você deverá configurar, ao criar um novo projeto web, o servidor tomcat e instalar o framework Spring MVC para construção de um projeto simples web utilizando o framework em questão para testar a configuração do servidor tomcat.

Referências:

Tutorial Tomcat - Instalação e configuração Básica. Disponível em: <<http://www.mhavila.com.br/topicos/java/tomcat.html>>. Acesso em: 20, setembro de 2022.

ROTEIRO DE AULA PRÁTICA 2

Unidade: Digite aqui

Aula (White Label)/Seção (KLS): Digite aqui

SOFTWARE

☒ Software / ☐ Acesso on-line

☐ Pago / ☒ Não Pago

Infraestrutura:

Computador;

Software IDE NetBeans com Java JDK instalado e configurado.

Descrição do software:

NetBeans é uma IDE de desenvolvimento gratuita e de código aberto. Útil no desenvolvimento de softwares utilizando as linguagens: Javascript, HTML5, PHP, C/C++ e Java. Ferramenta multiplataforma podendo ser instalada no Linux, macOS e Windows.

ATIVIDADE PRÁTICA 2

Atividade proposta:

Utilizando os principais conceitos do paradigma de Orientação a Objetos, crie uma pequena aplicação de gerenciamento bancário que possibilite ao usuário informar seu nome, sobrenome e CPF. Além disso, a aplicação deverá possibilitar ao usuário consultar saldo, realizar depósitos e saques. Esses procedimentos devem se repetir até que o usuário escolha encerrar o uso da aplicação.

Objetivos:

Fixar os conceitos de programação orientada a objetos;

Empregar práticas de instanciação de classes Java;

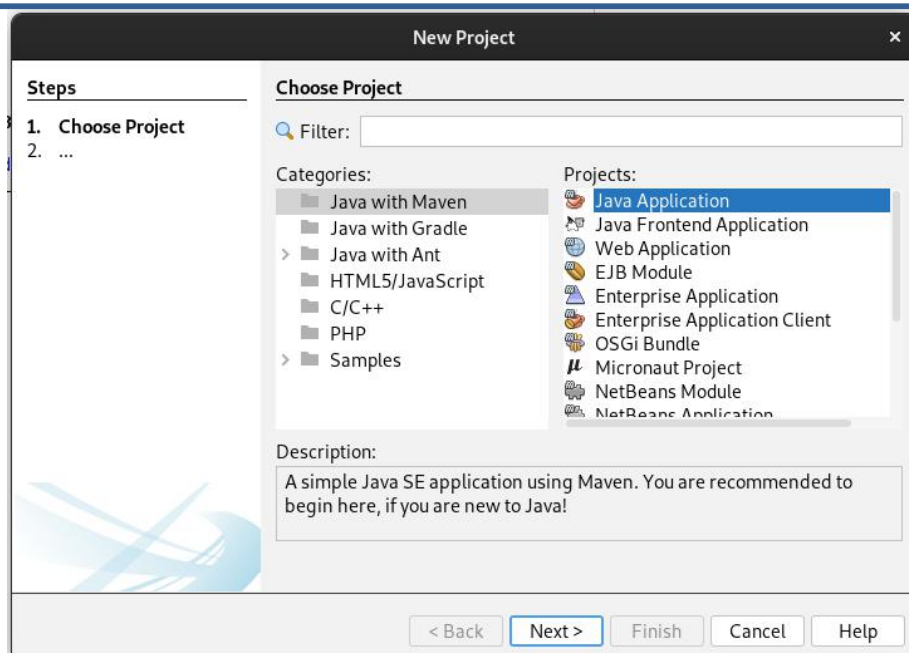
Utilizar métodos e atributos em Java.

Utilizar pacotes da linguagem java para implementação de software com paradigma de orientação a objetos.

Procedimentos para a realização da atividade:

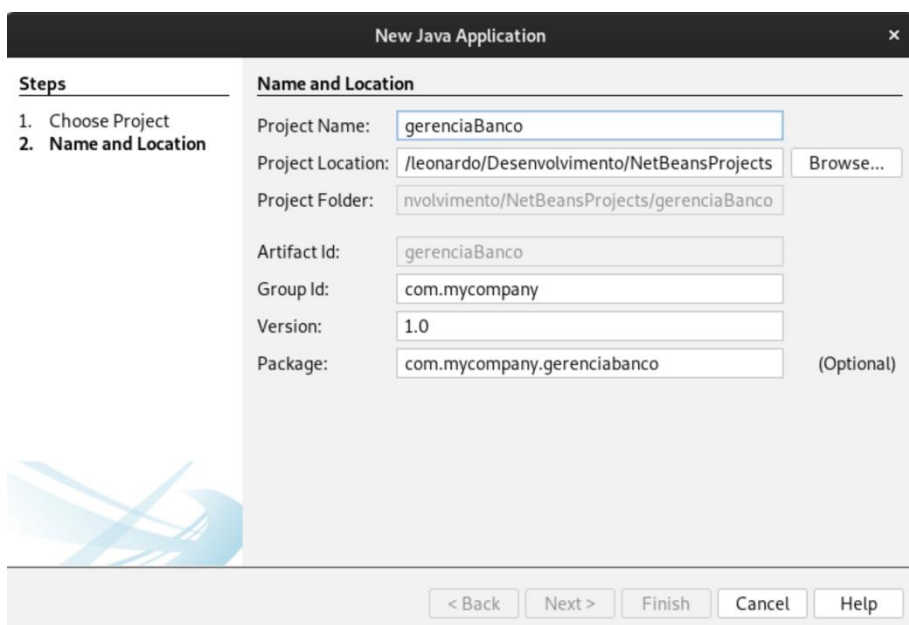
Será necessário ter devidamente instalado e configurado o ambiente de desenvolvimento com IDE NetBeans para a construção de um projeto novo. Para essa atividade, Sugere-se utilizar o nome gerenciaBanco para a aplicação. Para iniciar é necessário criar um novo projeto.

1 - Vá ao menu File e em seguida clique sobre New Project. Na sequência, deve-se escolher a categoria Java with Maven > Java Application. Veja:



Fonte: Elaborado pelo Autor.

2 - Em seguida, deve-se avançar e, na tela seguinte, dar o nome ao projeto. Como já mencionado, sugere utilizar o nome gerenciaBanco. Veja:



Fonte: Elaborado pelo Autor.

3 - Construa a aplicação em um único arquivo do tipo java main Class. Isso porque o método principal, que é chamado pela maquina virtual, deve estar nesse mesmo arquivo.

4 - No seu código você deverá construir, basicamente:

- a) a classe principal
- b) classe para dados pessoais e operações bancárias
- c) método para exibição do menu.

5 - Para a exibição do menu, será necessário utilizar uma estrutura de decisão para tratamento

das escolhas do usuário. Sugere-se utilizar a estrutura do...while e switch..case.

Checklist:

- 1 - Ambiente de desenvolvimento NetBeans e JDK devidamente configurado e funcionando.
- 2 - Criação de um projeto do tipo Java Application, em Java with Maven.
- 3 - Escrita do código em Java utilizando-se de conceitos da programação orientada a objetos.

Resultados da aula prática: Aluno, você deverá entregar:

Criação, configuração e implementação de projeto java application configurado. Aplicação de gerencia de banco com procedimentos básicos como informação de usuário (nome, sobrenome e CPF), e procedimentos bancários como consulta de saldo, depósito e retirada e interrupção da execução do código, contendo uma mensagem de despedida.

Referências:

Construtores em Java: Primeiros passos. Disponível em:
<<https://www.devmedia.com.br/construtores-em-java-primeiros-passos/28618#:~:text=Em%20Java%20apenas%20as%20Interfaces%20n%C3%A3o%20possuem%20construtores.&text=O%20construtor%20sempre%20tem%20a,%3A%20public%2C%20private%20ou%20protected.>>. Acesso em: 20, setembro de 2022.

ROTEIRO DE AULA PRÁTICA 3

Unidade: [Digite aqui](#)

Aula (White Label)/Seção (KLS): [Digite aqui](#)

SOFTWARE

☒ Software / ☐ Acesso on-line

☐ Pago / ☒ Não Pago

Infraestrutura:

Computador;

Software IDE NetBeans com Java JDK instalado e configurado.

Descrição do software:

NetBeans é uma IDE de desenvolvimento gratuita e de código aberto. Útil no desenvolvimento de softwares utilizando as linguagens: Javascript, HTML5, PHP, C/C++ e Java. Ferramenta multiplataforma podendo ser instalada no Linux, macOS e Windows.

ATIVIDADE PRÁTICA 3

Atividade proposta:

Implementar um novo módulo à aplicação de gerenciamento bancário criado na atividade anterior. Agora, nesse novo módulo, será implementado um módulo bancário voltado para aplicações financeiras. Nesse novo módulo, deverá ser disponibilizado ao usuário duas aplicações de renda fixa. Um CDB e um LCI.

Objetivos:

Compreender e utilizar conceitos de programação orientada a objetos para implementação de módulo de investimentos a ser integrado à aplicação de gerenciamento bancário criado na atividade anterior.

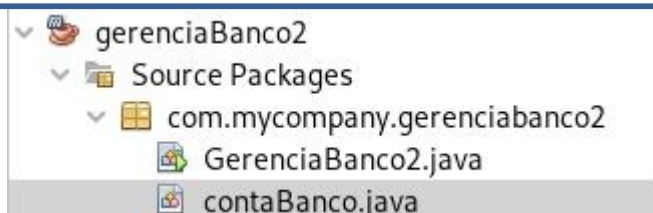
Procedimentos para a realização da atividade:

Para realização dessa atividade, será necessário duplicar o projeto criado na atividade anterior. Isso porque queremos ter separado os projetos, na intenção de termos as versões em separado com a finalidade de estudo.

1 - Para cumprir com a sugestão do enunciado é possível exportar o projeto ou, simplesmente, criar um novo projeto e copiar o código. Esse último será o procedimento a ser adotado. Crie um novo projeto, semelhante à construção realizada nas atividades anteriores. Vá em File > New Project. Em seguida, escolha Java with Maven > Java Application.

2 - Dê o nome de gerenciaBanco2 ao projeto, o que criará um arquivo gerenciaBanco2.java.

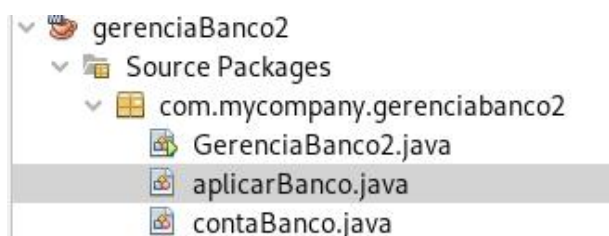
3 - Separe o projeto em classes distintas. Agora, você terá que criar uma classe contaBanco.java. A árvore de diretórios deverá ficar semelhante ao exibido na imagem a seguir:



Fonte: Elaborado pelo autor.

4 - Transfira o código classe contaBanco para o novo o arquivo contaBanco.java.

5 - Teste a aplicação e crie um novo arquivo chamado aplicarBanco.java. Para isso, clique sobre java source package com o botão direito do mouse, escolha New > Java Class. A árvore de diretório deverá passar a mostrar uma nova estrutura, veja:



Fonte: Elaborado pelo autor.

6 - A classe que foi escrita na atividade anterior chamada contaBanco, que está no arquivo gerenciaBanco.java daquela atividade, deverá ser separado, ou seja, será retirado e inserido no arquivo contaBanco.java, que passa a integrar esse projeto.

7 - Deverá ser criada a classe aplicarBanco.java que receberá todo o código relacionado à disponibilidade dos dois tipos de investimentos que nossa aplicação deverá ter: CDB e LCI. Note que um rende mais que o outro porque o CDB, o que rende mais, tem recolhimento de imposto de renda do rendimento. No entanto, não é uma informação a ser considerada no desenvolvimento desse módulo. Basta saber o motivo de um investimento render mais que o outro, afinal, não faria o menor sentido ter dois investimentos rendendo a mesma coisa, não é mesmo?

8 - Muita atenção no desenvolvimento desse módulo pois, o CBD rende 10% ao ano e o LCI 8% ao ano. Isso significa que o rendimento mensal precisa ser calculado. Imagine que o usuário queira investir valores mensais de 200 reais por 6 meses no CDB. Será necessário descobrir qual é o rendimento mensal dessa aplicação para fazer os cálculos de rendimento dos 6 meses investidos pelo usuário.

9 - O resultado deverá ser algo semelhante ao que já foi implementado anteriormente. Coisas como estrutura de decisão utilizada para a escolha dos procedimentos já existentes na aplicação, podem e devem ser aproveitados na implementação do módulo de investimentos. Veja:

```

Scanning for projects...

-----< com.mycompany.gerenciabanco2:gerenciaBanco2 >-----
[Building gerenciaBanco2 1.0-SNAPSHOT]
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ gerenciaBanco2 ---
#####
*****
#### Sistema de controle bancário ####
*****
#####
Informe seu nome:
Leonardo

Informe seu sobrenome:
Rocha

Informe seu CPF:
666555544433
você deseja realizar que tipo de transação:
a) Transações comuns:
b) Investimentos:

```

Fonte: Elaborado pelo autor.

Checklist:

- 1 - Ambiente de desenvolvimento NetBeans e JDK devidamente configurado e funcionando.
- 2 - Criação de um projeto e organização dos arquivos e pastas criados.
- 3 - Separação do código das classes já escritas na atividade anterior em duas, ou seja, a classe principal chamada gerenciaBanco2.java e contaBanco.java.
- 4 - A classe contaBanco() deverá ser transferida para o novo arquivo contaBanco.java
- 5 - Criação de uma nova classe chamada aplicarBanco.java, onde será escrito o código para implementação do módulo de investimentos em CDB e LCI.
- 6 - Montar os cálculos para descobrir o rendimento mensal de cada uma das aplicações para considerar na fórmula que será base para exibição do valor investido acrescido do rendimento desse valor ao longo do tempo investido.

Resultados da aula prática: Aluno, você deverá entregar:

Criação, estrutura, organização e codificação de uma aplicação que simula um sistema bancário onde o usuário poderá informar nome, sobrenome e CPF. Além disso, poderá definir que tipo de procedimento ele deseja realizar. Entre eles estarão:

- a) Transações comuns
- b) Investimentos.

Dentro do primeiro deverá ter:

- a) Verificar saldo
- b) Depositar valor
- c) Retirar valor
- x) Sair

Dentro do segundo deverá ter:

- a) CDB (10% ao ano)
- b) LCI (8% ao ano)

c) Sair

O módulo de investimentos deverá permitir que o usuário informe: valor a ser investido
quantidade de parcelas (quantidade de meses). Com base nessas informações, deverá exibir:

O rendimento desse investimento no ano

A quantidade de parcelas informada pelo usuário

Exibir um relatório final que mostre o valor total investido acrescido do rendimento obtido.

Referências:

Como formatar um double com duas casas decimais? Disponível em:
<<https://www.devmedia.com.br/forum/como-formatar-um-double-com-2-casas-decimais/564740>>. Acesso em: 22, setembro de 2022.

Como calcular juros compostos em java. Disponível em:
<<https://www.arquivodecodigos.com.br/dicas/1179-java-calculando-juros-compostos-e-montante-usando-java.html>>. Acesso em: 22, setembro de 2022.

Entendendo e aplicando herança em java. Disponível em:
<<https://www.devmedia.com.br/entendendo-e-aplicando-heranca-em-java/24544>>. Acesso em: 22, setembro de 2022.

ROTEIRO DE AULA PRÁTICA 4

Unidade: [Digite aqui](#)

Aula (White Label)/Seção (KLS): [Digite aqui](#)

SOFTWARE

☒ Software / ☐ Acesso on-line

☐ Pago / ☒ Não Pago

Infraestrutura:

Computador;

Software IDE NetBeans com Java JDK instalado e configurado.

Descrição do software:

NetBeans é uma IDE de desenvolvimento gratuita e de código aberto. Útil no desenvolvimento de softwares utilizando as linguagens: Javascript, HTML5, PHP, C/C++ e Java. Ferramenta multiplataforma podendo ser instalada no Linux, macOS e Windows.

ATIVIDADE PRÁTICA 4

Atividade proposta:

Construir uma aplicação, uma calculadora para calcular os investimentos feitos na aplicação desenvolvida na atividade anterior mas também, implementar uma operação adicional. A radiciação. Tudo isso, utilizando conceitos de herança.

Objetivos:

- Fixar os procedimentos necessários para construção de um projeto na IDE NetBeans.
- Compreender conceitos relacionados a construção de métodos na programação em Java.
- Aprender a utilizar estruturas de decisão para auxiliar a definição de escolhas na programação em Java.
- Identificar e aplicar conceitos de orientação a objetos com uso de herança.
- Entender como invocar métodos em programação orientada a objetos.

Procedimentos para a realização da atividade:

Será necessário ter devidamente configurado o ambiente de desenvolvimento NetBeans com máquina virtual JDK.

1 - Crie um novo projeto em File > New Project. Em seguida escolha Java with Maven > Java Application.

2 - Atribua o nome de AppCalculadora a esse novo projeto e finalize o processo de criação. Você terá uma tela semelhante a que é exibida na imagem a seguir:

```

1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Project/Maven2/JavaApp/src/main/java/${packagePath}/${mainClassName}.java to edit this template
4  */
5
6  package com.mycompany.teste;
7
8  /**
9   *
10  * @author leonardo
11  */
12  public class ApCalculadora {
13
14      public static void main(String[] args) {
15          System.out.println("Hello World!");
16      }
17  }
18

```

Fonte: Elaborado pelo autor.

3 - Crie uma nova classe java e defina seu nome como calculos.java. Essa será a classe que conterá todos os cálculos de nossa calculadora, incluindo o cálculo de radiciação.

4 - Na classe principal, construa o código que apresentará a aplicação, ou seja, informará ao usuário que se trata de uma calculadora para auxiliar no controle dos investimentos. Não é necessário utilizar interface gráfica para esse projeto.

5 - O projeto deverá solicitar ao usuário que ele informe o seu nome. Isso será necessário pois na classe calculos.java a aplicação iniciará a solicitação dos números que o usuário deverá informar, cumprimentando o mesmo.

6 - Nas operações básicas: adição, subtração, divisão e multiplicação, o programa deverá solicitar que o usuário informe a operação desejada e digite dois números, para que o cálculo da operação escolhida possa ser executado.

7 - Na radiciação, como já se sabe, o usuário deverá informar apenas um número. É sobre esse número que o cálculo de raiz quadrada deverá ser feito.

8 - Não é necessário se preocupar em arredondar os dois números das casas decimais após a vírgula.

9 - Utilize a estrutura de decisão já utilizada nas atividades anteriores. O Switch..case.

Checklist:

1 - Fixar os procedimentos de criação de projetos utilizando a IDE NetBeans.

2 - Memorizar algumas estruturas próprias da linguagem java, como a estrutura de decisão.

3 - Compreender a criação de classes distintas e como elas se relacionam no contexto do projeto.

4 - Escrever métodos construtores para o projeto.

5 - Invocar métodos escritos para realização dos cálculos.

6 - Conhecer métodos prontos da linguagem Java disponíveis para realizar cálculos mais complexos.

Resultados da aula prática: Aluno, você deverá entregar:

Construção de um softwre que simula as funções básicas de uma calculadora, implementando as operações básicas de adição, subtração, multiplicação e divisão. Além disso, implementa um cálculo extra, o de radiciação, utilizando os conceitos fundamentais da programação orientada a objetos utilizando a linguagem Java, sobre a IDE NetBeans.

Referências:

Java sqrt() method with examples. Disponível em: <<https://www.geeksforgeeks.org/java-sqrt-method-examples/>>. Acesso em: 18, agosto de 2022.

Switch Case Java. Disponível em: <<https://blog.betrybe.com/java/switch-case-java/#:~:text=O%20switch%20case%20Java%20%C3%A9,usando%20v%C3%A1rios%20if%20else%20encadeados.>>. Acesso em: 18, agosto de 2022.