

Instruções:

- Poderá ser realizado em **Dupla**.
- Postar no Teams os arquivos txt compactados.
- Data da entrega até: 17/03/2022

Criar um projeto:

Os programas no NetBeans precisam estar em projetos. Os projetos são estruturados em pacotes (que são pastas) e classes (que são arquivos), assim como mostra a . Para criar um projeto com esta estrutura você precisa fornecer os nomes do projeto, pacote e classe na tela de criação do projeto assim como mostra a Figura 2. OBS: As telas são do Netbeans versão 13, porém são iguais a versão 8, a única diferença é que estão em Inglês.

Para fazer os exercícios a seguir utilize um único projeto e ele deverá ter os seguintes nomes:

- Projeto: **Pratica1**
- Pacote: **aula**
- Classe: **Principal**
-

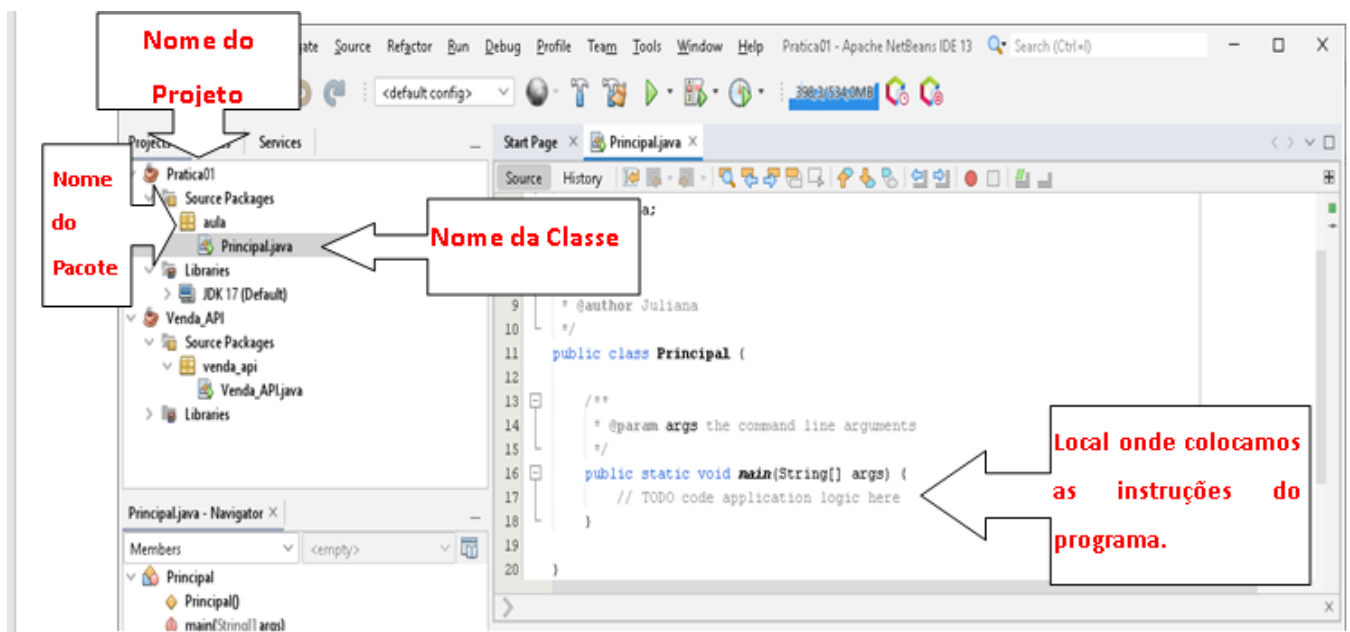


Figura 1 - Estrutura um projeto Java no NetBeans Versão13)

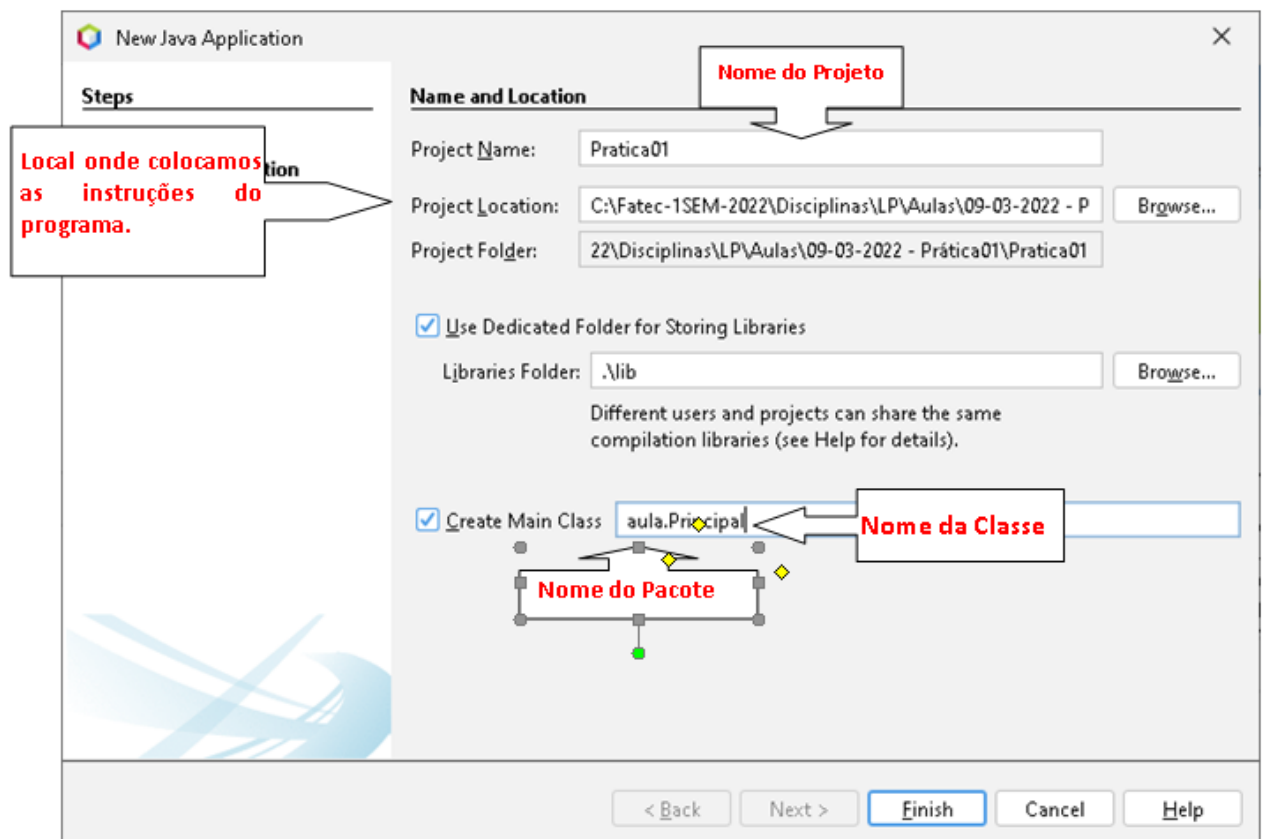


Figura 2 - Tela para fornecer os parâmetros de criação de um projeto Java no NetBeans (Versão 13).

Exercício 1 - Fazer um programa que imprime os números de 1 a 5 na tela. Usando `for(início; teste; incremento) { instruções do bloco }`

Dicas:

- este programa não requer entrada do usuário, ou seja, não precisamos nos preocupar com a leitura do teclado;
- para imprimir na tela um número podemos usar a instrução `System.out.println(1);`
- para imprimir todos os números precisamos repetir o item anterior 5 vezes. Como a diferença de uma linha para outra será somente o conteúdo do método `println`, que será `1, 2, ...`, então isto nos leva a ter de usar uma estrutura de repetição.
- Após finalizar o Exercício 1 copie o código programado para o arquivo TXT e limpe o seu projeto apagando as instruções digitadas, pois iremos aproveitar o mesmo projeto para os demais exercícios.

Exercício 2 - Fazer um programa que imprime os números de 1 a 5 na tela. Usando `while (teste) { instruções do bloco }` Vide Dicas do Exercício 1.

Exercício 3 - Fazer um programa que imprime os números de 1 a 5 na tela. Usando `do { instruções do bloco } while(teste);` Vide Dicas do Exercício 1.

Exercício 4 - Fazer um programa que imprime os números pares que existem no intervalo entre 1 e 15.

Dicas:

- a) este programa não requer entrada do usuário, ou seja, não precisaremos nos preocupar com a leitura do teclado;
- b) para imprimir na tela um número podemos usar a instrução `System.out.println(1);`
- c) antes de imprimir um valor na tela precisaremos checar se ele é par. Para checar precisamos perguntar "você é par?" e a resposta deverá ser "sim ou não". Para checar o resultado de uma pergunta e tomar a decisão de imprimir o valor na tela, só temos uma opção, que é usar a estrutura `if(teste){ bloco de instruções }`
- d) para checar se um valor é par podemos usar o operador (`%`), pois ele retorna o resto da divisão:
 - `3%2` tem como resultado 1, pois 1 é o resto da divisão 3/2como qualquer valor inteiro `x` quando operado com `%` só poderá resultar em `0` ou `1`, e será `0` quando o `x` for **par**, então podemos fazer o seguinte teste:
 - `x%2 == 0`
- e) para imprimir todos os números pares precisaremos repetir o item (b). Para isso podemos usar uma estrutura **for**, acontece que nem todos os valores entre 1 e 15 são pares, então precisaremos checar, ou seja, precisaremos colocar a instrução do item (b) dentro de um **if** com um teste semelhante ao item (d). Desta forma, o programa será composto por um único bloco **for**, que terá dentro um bloco **if** e dentro dele a instrução para imprimir o número;
- f) Após finalizar o Exercício 4 copie o código do programado para um arquivo TXT e limpe o seu projeto apagando as instruções digitadas, pois iremos aproveitar o mesmo projeto para os demais exercícios.

Exercício 5 - Fazer um programa que imprime na tela o somatório dos números entre 0 e 10.

Dicas:

- a) este programa não requer entrada do usuário, ou seja, não precisaremos nos preocupar com a leitura do teclado;
- b) o somatório é calculado somando cada um dos valores que compõem o intervalo, como os valores são inteiros, então os valores serão somente: `0+1+2+3+...+9+10`

Acontece que o programa não funciona assim, ele somente opera entre pares de números, então teremos de criar uma variável para estocar os pares de soma, e toda variável precisa iniciar com algum valor, como o primeiro valor a ser estocado é zero, então podemos criar uma variável `s` do tipo inteiro e iniciá-la com o valor zero:

```
int s = 0;
```

Na sequência podemos fazer as seguintes somas:

```
s = s + 1;
```

```
s = s + 2;
```

```
s = s + 3;
```

```
...
```

```
s = s + 9;
```

```
s = s + 10;
```

No final a variável `s` terá o somatório do intervalo de 0 a 10. No entanto, digitar cada uma das instruções de soma seria inviável para intervalos grandes, então podemos colocar cada uma dessas instruções de soma dentro de uma estrutura **for**, pois temos o valor de início, valor de finalização e incremento.

c) desta forma temos que este programa será formado por três instruções:

- uma instrução para a declaração de uma variável do tipo inteiro e a sua inicialização com valor zero;
- uma instrução contendo um bloco **for** e dentro deste bloco teremos uma instrução de soma;
- uma instrução para imprimir na tela o conteúdo da variável que possui o somatório. Observação, esta instrução deverá estar fora do bloco **for**, pois ela deverá ser executada somente após o intervalo ser todo somado.

Exercício 6 - Fazer um programa que imprime na tela a média dos números ímpares que estão no intervalo entre 0 e 100.

Dicas:

a) a média é calculada dividindo o somatório pela quantidade de valores utilizados no somatório. Desta forma teremos de criar duas variáveis inteiras, uma para acumular o somatório e outra para contar a quantidade de vezes que a instrução de soma foi executada. Ambas devem ser iniciadas com zero;

b) desta forma temos que este programa será formado por quatro instruções:

- uma instrução para a declaração de uma variável do tipo inteiro e a sua inicialização com valor zero, esta variável será usada para acumular o somatório;
- uma instrução para a declaração de uma variável do tipo inteiro e a sua inicialização com valor zero, esta variável será usada para contar a quantidade de vezes que a instrução de soma será executada;
- uma instrução contendo um bloco **for** e dentro deste bloco teremos:
 - um bloco **if**, para checar se o valor é ímpar, e dentro desse bloco **if** teremos duas instruções:
 - uma para somar o valor e acumular na variável correspondente;
 - uma para incrementar o valor da variável contadora, pois esta variável irá iniciar com valor zero e este será somada de 1 a cada vez.
- uma instrução para imprimir na tela o valor da média. Para computar a média precisaremos dividir as duas variáveis. Observação, quando dividimos dois valores/variáveis inteiras, o resultado também será inteiro, ou seja, $4/3$ será igual a 1 e não 1.33. Para resolver este problema teremos de converter um dos valores para float, podemos fazer isso das seguintes maneiras:
 - $4/3f$: neste caso transformou-se o valor 3 em 3.0;
 - $4f/3$: neste caso transformou-se o valor 4 em 4.0;
 - $4f/3f$: neste caso transformou-se os valores em 4.0 e 3.0;
 - $4/(float)3$: neste caso transformou-se o valor 3 em 3.0;
 - $(float)4/3$: neste caso transformou-se o valor 4 em 4.0. Atenção, é um erro considerar que isso é equivalente a $(float)(4/3)$, pois esta operação resultaria em $(float)(1) \rightarrow 1.0$

Exercício 7 - Fazer um programa que exibe uma janela de diálogo para o usuário entrar com um nome e após isso o programa imprime este nome na tela de saída, mas somente se o usuário fornecer um nome.

Dicas:

a) para ler a entrada do usuário podemos usar a janela de diálogo **JOptionPane.showInputDialog** (Figura 3). Esta janela é exibida chamando o método **showInputDialog**, que por sua vez precisa ser precedido do nome da classe

(**JOptionPane**). O método **showInputDialog** precisa receber um parâmetro, que será exibido na janela de diálogo. Como a classe **JOptionPane** não faz parte do pacote padrão do Java, então ela precisa ser importada, acontece que neste caso precisamos colocar o pacote que esta classe se encontra, que é **javax.swing**

b) o método **showInputDialog** retorna algum dos seguintes valores:

- "Ana Maria", ou seja, um valor String: quando o usuário digitar Ana Maria e pressionar o botão ok;
- "", ou seja, um valor String: quando o não digitar algo e pressionar o botão ok;
- "18", ou seja, um valor String: quando o usuário digitar 18 e pressionar o botão ok;
- null, ou seja, a ausência de uma String: quando o usuário digitar qualquer valor e pressionar o botão cancelar.

c) desta forma temos que este programa será formado por duas instruções:

- uma instrução para a declaração de uma variável do tipo String para receber o retorno do método **showInputDialog**, juntamente com a chamada deste método;
- uma instrução contendo um bloco **if** para checar se o conteúdo fornecido é diferente (**!=**) de **null** e também diferente de zero caracteres, que é usando o método **equals** da classe String, ou seja,
 - **nome.equals("") == false** pois o método **equals** retorna **true** quando o parâmetro fornecido é igual ao conteúdo da String **nome**;

Dentro do bloco **if** deverá ter a instrução **System.out.println** para imprimir na tela o conteúdo da variável **nome**.

Exercício 8 - Fazer um programa que exibe uma janela de diálogo para o usuário entrar com um número inteiro e após isso o programa imprime na tela a mensagem dizendo que o número é par ou ímpar.

Dicas:

a) para ler a entrada do usuário podemos usar o método **showInputDialog**, que por sua vez, irá retornar um valor String (texto), independente de o usuário digitar um número. Desta forma, precisaremos fazer uma conversão de texto para inteiro, o método **parseInt**, que pertence a classe Integer, é usado para fazer esta conversão

```
int valor = Integer.parseInt( "18" );
```

- o método **parseInt** precisa ser precedido do nome da classe que o contém;
- o método **parseInt** precisa receber uma String contendo o valor a ser convertido, que neste exemplo, é **"18"**;
- o método **parseInt** retorna um valor do tipo **int**, por este motivo, foi criada a variável **v**.

b) desta forma temos que este programa será formado pelas seguintes instruções:

- uma instrução para a declaração de uma variável do tipo String para receber o retorno do método **showInputDialog**, juntamente com a chamada deste método;
- uma instrução para a declaração de uma variável do tipo int para receber o retorno do método **parseInt**, juntamente com a chamada deste método;
- uma instrução contendo um bloco **if/else** para checar se o conteúdo fornecido possui o resto da divisão por 2 igual 0. Dentro do bloco do **if** terá a instrução **System.out.println** para imprimir que o valor é par e dentro do bloco **else** terá a instrução **System.out.println** para imprimir que o valor é ímpar.

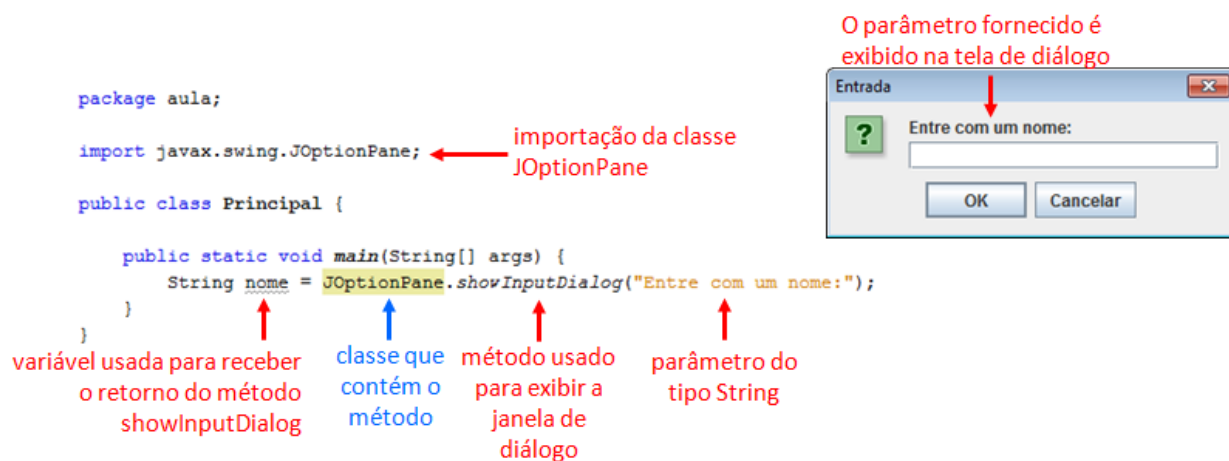


Figura 3 - Exemplo de uso do método showInputDialog.