



Aula 4 – Exercício Recursão

■ Depuração de Código.

- *Conceito.*
- *Utilização da ferramenta de Debug no Netbeans.*
- *Break Point.*

■ Recursão de métodos.

- *Utilização de recursão.*
- *Exercícios.*

Depuração de Código



A depuração ou “Debug” é um processo essencial na construção de aplicações com maior complexidade.



Possibilitando encontrar e eliminar erros de um código de forma sistemática e organizada.



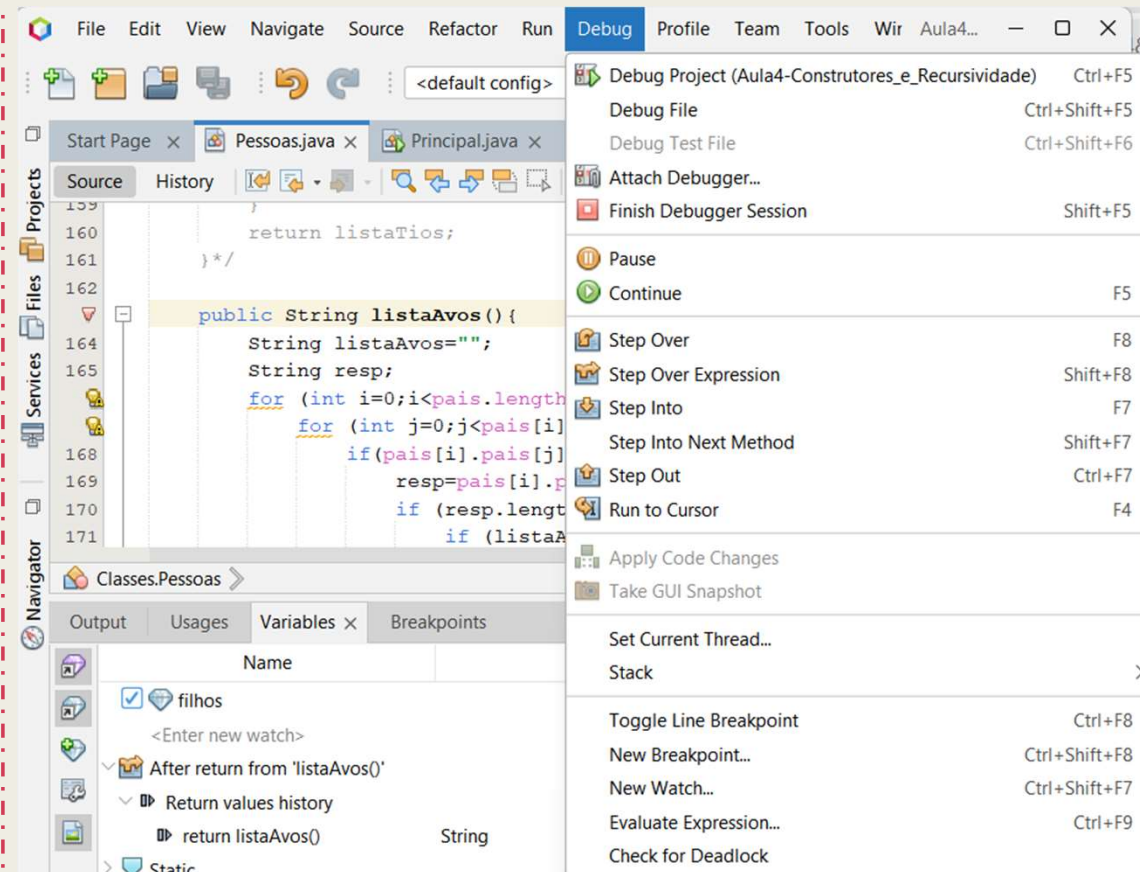
Otimizar e melhorar o código já desenvolvido.



Validar os requisitos de projeto, verificando os métodos e recursões no código.

Depuração de Código no NetBeans

- O primeiro passo para iniciar a depuração do código é definir os pontos de parada “break points”.
- Break points devem ser adicionados em pontos do código na qual o programador tem o interesse de investigar o estado das variáveis.
- As ferramentas de desenvolvimento de códigos, normalmente possuem recursos de “Debug” incorporadas.
- No Netbeans a ferramenta de depuração se encontra na aba **Debug**.



Depuração de Código no NetBeans

O *break point* é adicionado ao clicar sobre o número da linha de código

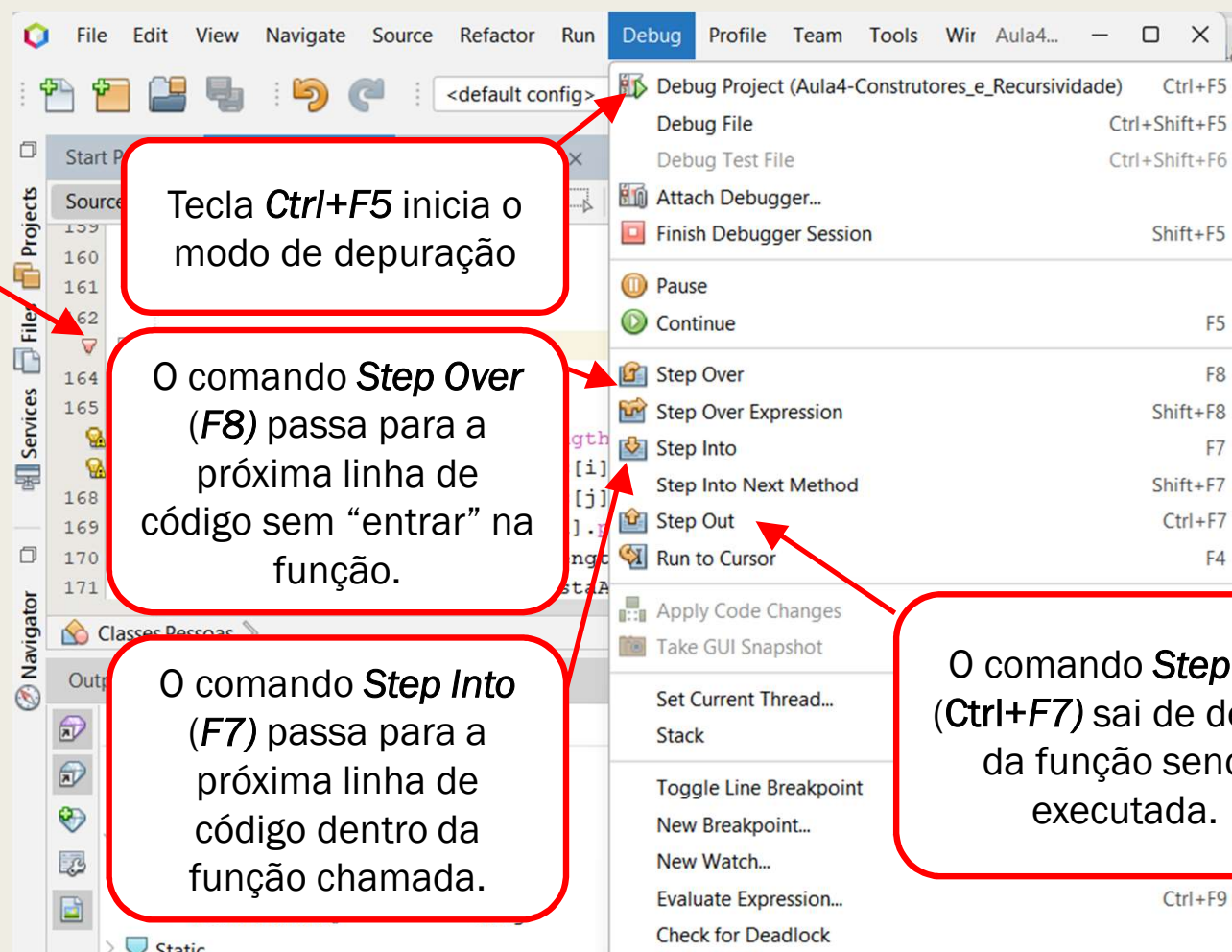
Em modo de depuração a execução do código será interrompida quando atingir um *break point*.

Tecla **Ctrl+F5** inicia o modo de depuração

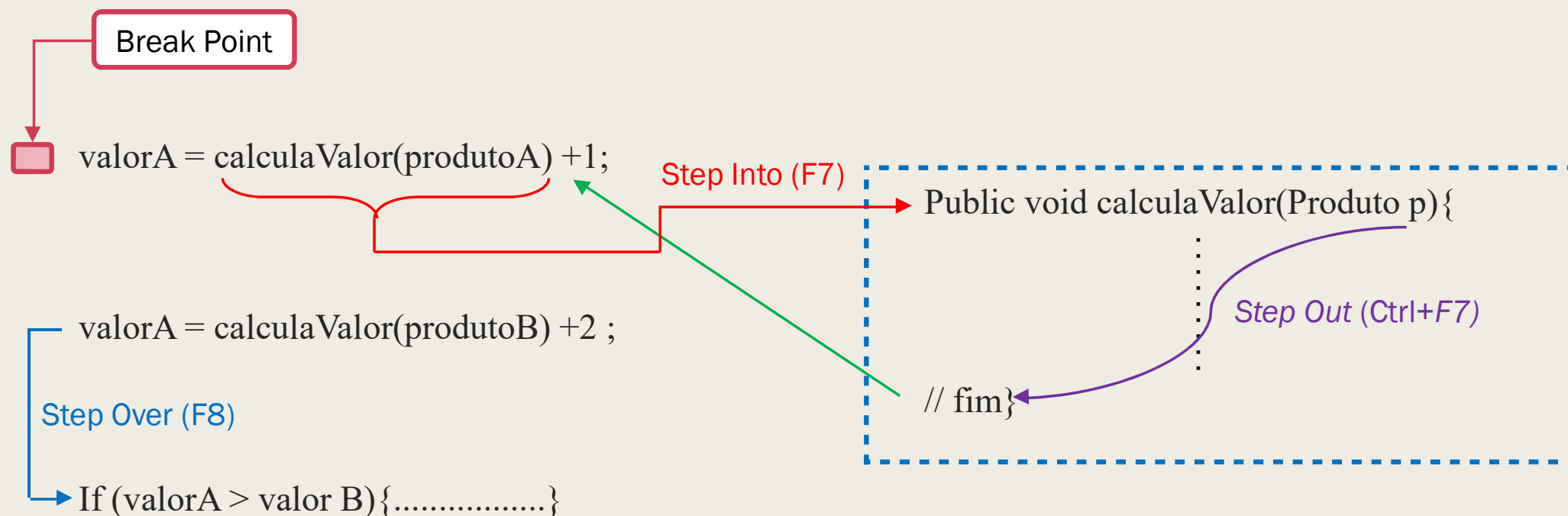
O comando **Step Over (F8)** passa para a próxima linha de código sem “entrar” na função.

O comando **Step Into (F7)** passa para a próxima linha de código dentro da função chamada.

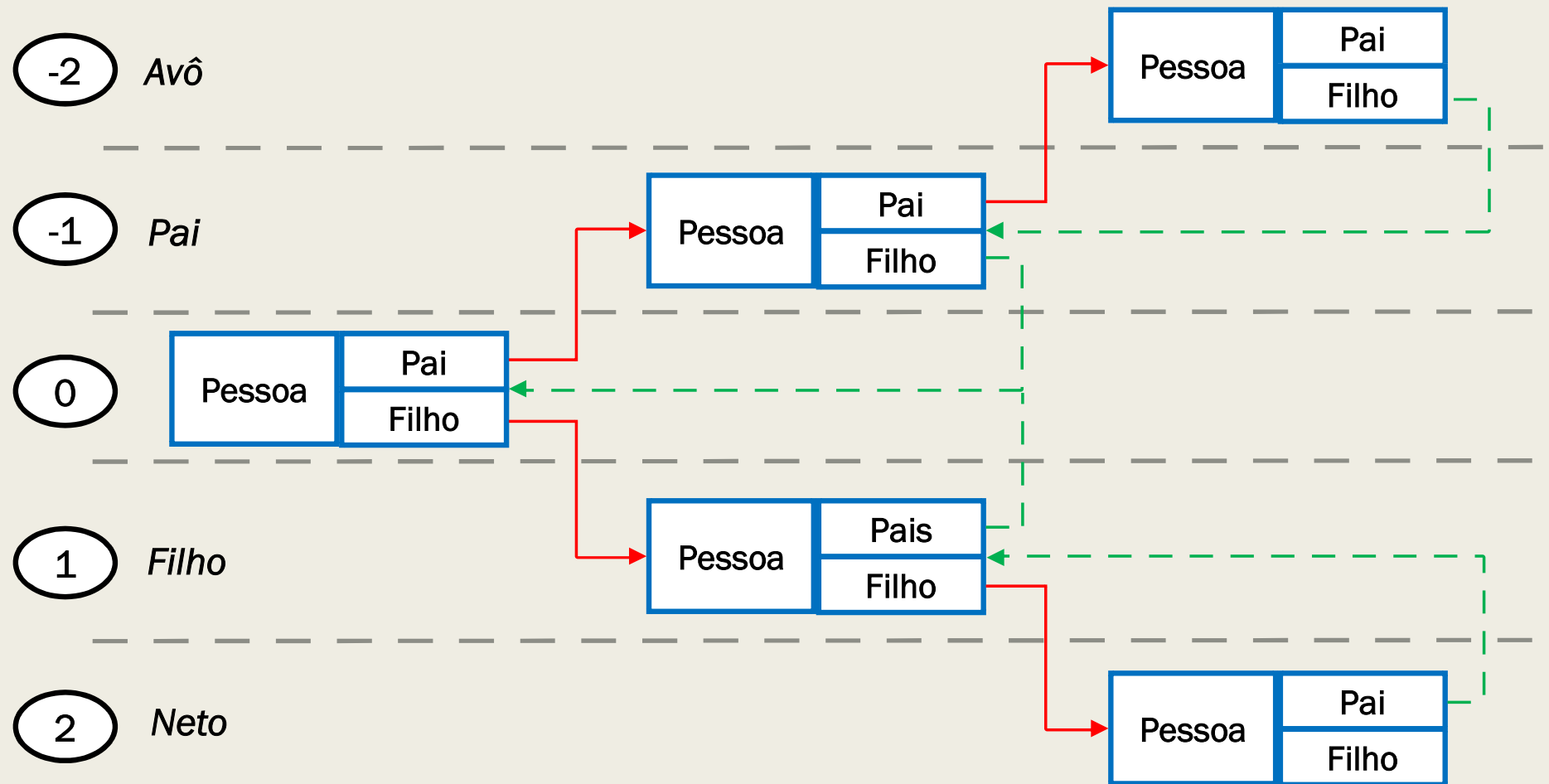
O comando **Step Out (Ctrl+F7)** sai de dentro da função sendo executada.



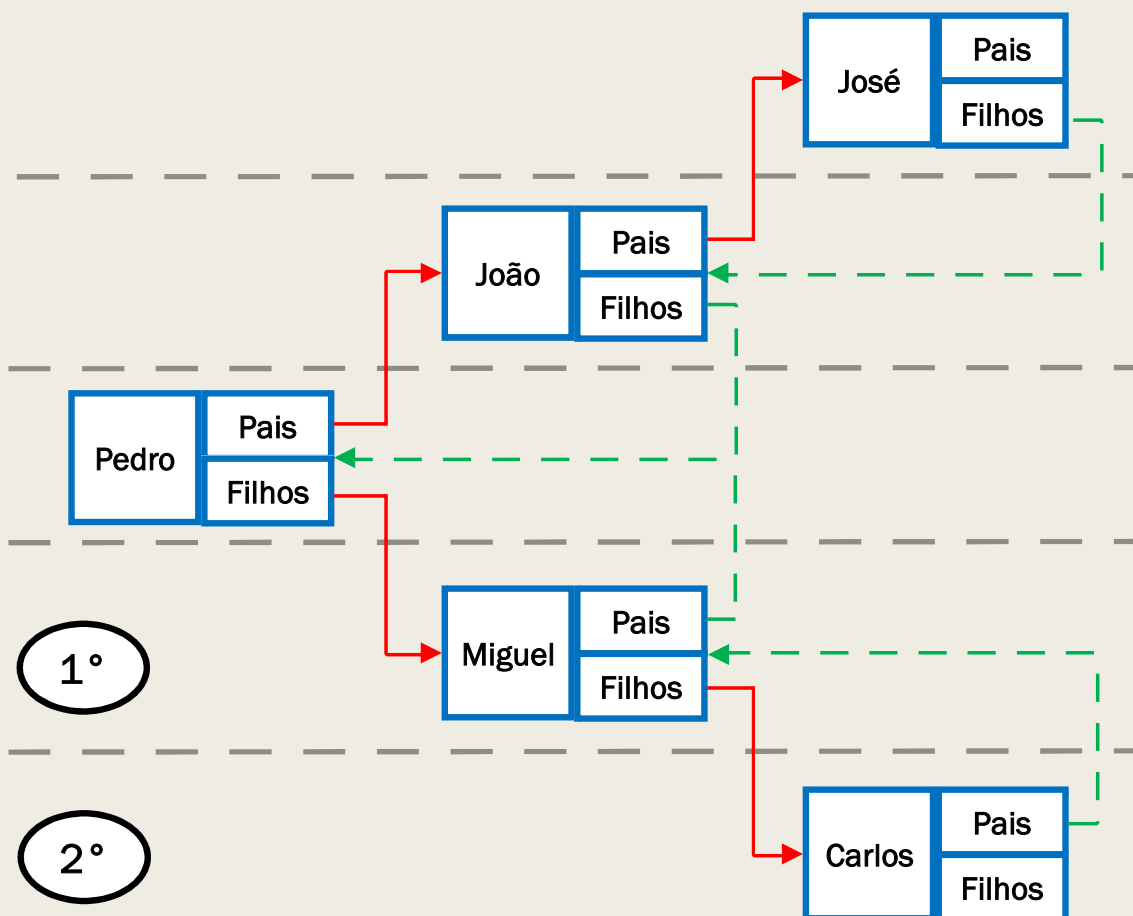
Depuração de Código no NetBeans



Recursividade



Recursividade



// Lista de descendentes paternos.

String **resp**="";

Pessoa **p**;

resp = **this**.nome + " => ";

descendentes(**p**,**resp**);

```
public void descendentes(pessoa,resp){
    if (filho != null){
        resp += filho.nome + " => ";
        descendentes(pessoa,resp);
    }
}
```

Início: **resp** ="Pedro =>"

1° interação: **resp** ="Pedro =>" Miguel,"

2° interação: **resp** ="Pedro =>" Miguel => Carlos"

Bibliografia

- **Java Básico e Orientação a Objeto (livro);**
 - Clayton Escouper das Chagas; Cássia Blondet Baruque; Lúcia Blondet Baruque.
 - Fundação CECIERJ, 2010: <https://canal.cecierj.edu.br/012016/d7d8367338445d5a49b4d5a49f6ad2b9.pdf>
- **Java e Orientação a Objetos (apostila);**
 - Caelum, 2023: <https://www.caelum.com.br/apostila/apostila-java-orientacao-objetos.pdf>
- **Modificadores de acesso em Java(artigo);**
 - DevMedia, 2023: <https://www.devmedia.com.br/modificadores-de-acesso-em-java/18822>
- **Get e Set - Métodos Acessores em Java(artigo);**
 - DevMedia, 2023: <https://www.devmedia.com.br/get-e-set-metodos-acessores-em-java/29241>
- **Sobrecarga e sobreposição de métodos em orientação a objetos(artigo);**
 - DevMedia, 2023: <https://www.devmedia.com.br/sobrecarga-e-sobreposicao-de-metodos-em-orientacao-a-objetos/33066>
- **Análise e modelagem de sistemas com a UML: com dicas e exercícios resolvidos (livro);**
 - Luiz Antônio Pereira, 2011: <https://luizantoniopereira.com.br/downloads/publicacoes/AnaliseEModelagemComUML.pdf>