

Algoritmos

Prof. Ricardo Luiz de Freitas

ricardo.freitas@academico.domhelder.edu.br



Conteúdo 3

- Programas
- Compiladores/IDE
- Compilador **IntelliJ**
- Projetos/Pacotes
- Dados
- Identificadores
- Variáveis
- Exercícios

Algoritmo x Programas

■ Programas:

- Algoritmo traduzido para uma linguagem de programação.

■ Linguagem de programação:

- Conjunto de símbolos e regras de sintaxe (Pascal, C++, Java, etc.).

■ Programa fonte:

- Composto por uma sequência de comandos escritos em uma linguagem de programação de alto nível (Java, Python, etc.).

■ Programa objeto:

- Composto por uma sequência de comandos escritos em linguagem de máquina (baixo nível).

Exemplo 1 - Programa fonte em Pascal

```
program CalculaFatorial;  
{ Função: calcular o fatorial de um número }  
var Numero, Fatorial, Contador : integer;  
begin  
    write('Informe um número: ');  
    readln(Numero);  
    Fatorial := 1;  
    for Contador := 2 to Numero do  
        begin  
            Fatorial := Fatorial * Contador;  
        end;  
    write('Fatorial = ', Fatorial);  
end.
```

Exemplo 2 - Programa fonte em Python

```
def main():  
    ...  
    Função: Calcular o fatorial de um número  
    ...  
    print("Cálculo do fatorial de um número\n")  
    # leia o valor de n  
    n = int(input("Digite um número inteiro não-negativo: "))  
    # inicialização da variável que armazena os fatoriais  
    n_fat = 1  
    # calcule n!  
    for i in range(2,n+1):  
        n_fat = n_fat * i  
    print("%d! = %d" %(n, n_fat))
```

Exemplo 3 - Programa fonte em Portugol

```
algoritmo "CalculaFatorial"  
  // Função : calcular o fatorial de um número  
  var numero, fatorial, contador : inteiro  
  inicio  
    escreva ("Informe um número: ")  
    leia (numero)  
    fatorial <- 1  
    para contador de 2 ate numero faca  
      fatorial <- fatorial * contador  
    fimpara  
    escreval ("Fatorial = ", fatorial)  
finalgoritmo
```

Exemplo 4 - Programa fonte em Java

```
import java.util.Scanner;

public class CalculaFatorial {
    // Função : Calcular o fatorial de um número
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        System.out.print("Informe um número: ");
        long numero = teclado.nextLong();
        long fatorial = 1L;
        for(long contador = 2; contador <= numero; contador++) {
            fatorial = fatorial * contador;
        }
        System.out.println("Fatorial = " + fatorial);
        teclado.close();
    }
}
```

Tipos de Programação

- **PI (Programação Imperativa):**
 - Modelo de programação onde sequências de comandos mudam o estado (variáveis) de um programa (Portugol);
- **POO (Programação Orientada a Objetos):**
 - Modelo de análise, projeto e programação baseado na composição e interação entre diversas unidades chamadas de “objetos” (Java, C#, Python).

Tipos de Linguagens de Programação

■ Linguagens Procedurais:

- ❑ Leem uma linha por vez;
- ❑ A linguagem **C** é procedural.

■ Linguagens orientadas a objetos

- ❑ Leem uma linha por vez;
- ❑ Modelam objetos por meio do código;
- ❑ Enfatizam a interação do objeto;
- ❑ Permitem uma interação sem uma ordem prescrita;
- ❑ **Java** e **C++** são linguagens orientadas a objetos.

Compiladores/IDE

- São “ambientes” integrados de desenvolvimento (*IDE Integrated Environment Development*), ou programas, utilizados para se criar programas;
 - ❑ Suportam apenas uma linguagem de programação;
 - ❑ A maioria dos compiladores/*IDE* transformam o Programa FONTE (linguagem de programação), digitado pelo programador, em um Programa OBJETO (linguagem de máquina) que roda diretamente em cima do SO (Sistema Operacional) (por exemplo: Delphi, C++, VB, etc.);
 - ❑ Os demais compiladores/IDE interpretam e executam o código fonte, utilizando uma máquina virtual, e independem do SO (por exemplo: Portugol, Java, C#, Python, etc.).
- Exemplos:
 - ❑ VisuAlg / G-Portugol = Portugol
 - ❑ PascalZIM / Delphi = Pascal
 - ❑ IntelliJ / Eclipse / JBuilder / JDK / NetBeans = Java

Compiladores/IDE

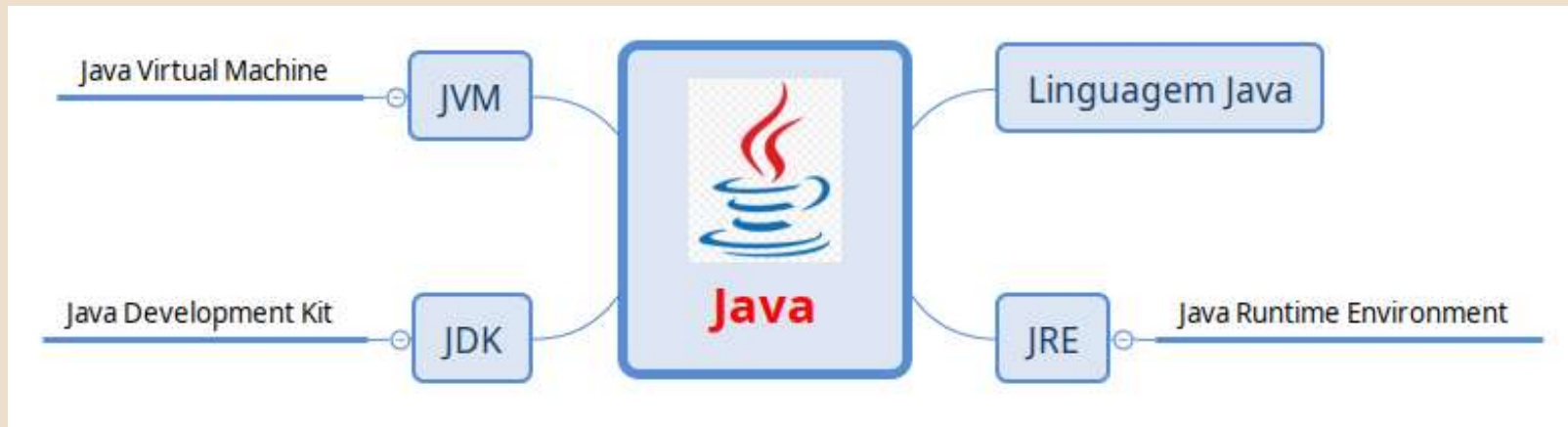
■ Etapas:

- ❑ **Edição**: digitação e alteração do programa;
- ❑ **Compilação**: verificação de erros de sintaxe do programa;
- ❑ **Execução**: rodar o programa, entrando com os dados e obtendo os resultados.

■ Erros que podem aparecer:

- ❑ **Sintaxe**: identificado pelo compilador na etapa de compilação;
- ❑ **Lógica**: identificado pelo programador na etapa de execução;
- ❑ **Run-time** : identificado pelo Java na etapa de execução (necessário tratamento das exceções).

Ecosystem Java



- O compilador Java gera um **bytecode** que roda dentro da JVM que está dentro da JRE, ou seja, o compilador Java não gera código em linguagem máquina, diferente do Delphi ou do C#, onde o código gerado roda diretamente em cima do SO (Sistema Operacional). Por isso um mesmo programa escrito em Java e compilado pode rodar em Windows, Linux ou Mac, basta os SOs terem a JRE instalada.

Java SE Development Kit (JDK)

■ Instalação do Java:

- Acesse o link:

<https://www.oracle.com/technetwork/java/javase/downloads/index.html>;

- Instale a última versão da plataforma “Java SE Development Kit ... downloads”:

- Clique na aba referente ao seu SO (Linux, macOS ou Windows);
- Clique em “... Installer”;
- Execute o executável baixado;
- Clique em <ENTER> para todas as telas.

Compilar/Executar Programas

- Via barra de comando (CMD):
 - Java programa.java >>> compila;
 - Java programa >>> executa.

- Via plataforma on-line:
 - GDB online Debugger (setar para Java);
 - <https://www.onlinegdb.com>.

- Via IDE (exemplo: IntelliJ, Eclipse, etc.).

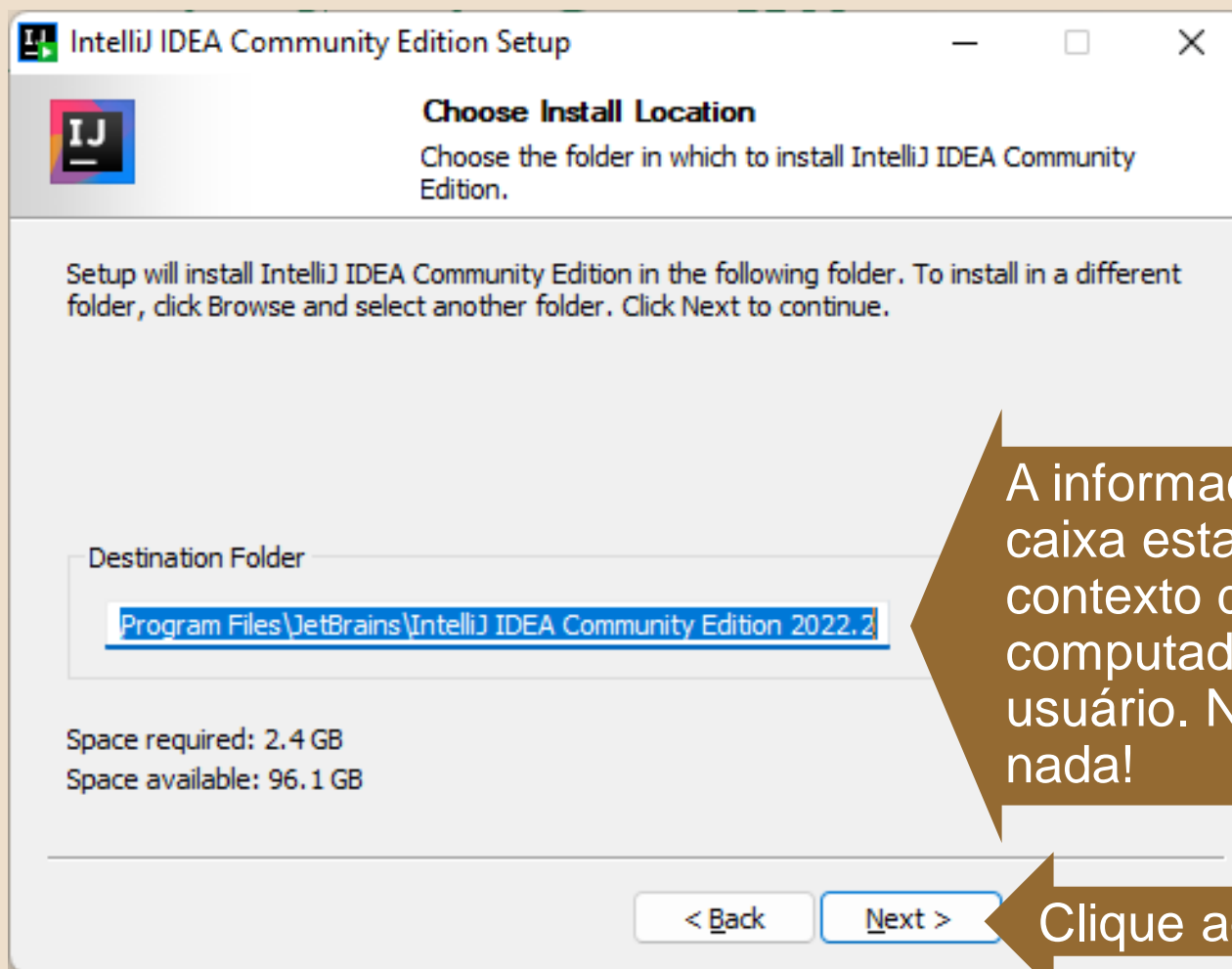
Compilador/IDE IntelliJ

- Download do arquivo para instalação
<https://www.jetbrains.com/pt-br/idea/download/#section=windows>:
 - Baixar a versão Community Edition (gratuita).
- Após o download do arquivo de instalação, execute-o para iniciar a instalação;
- Siga as instruções dos slides seguintes até concluir a instalação.

Instalação do IntelliJ



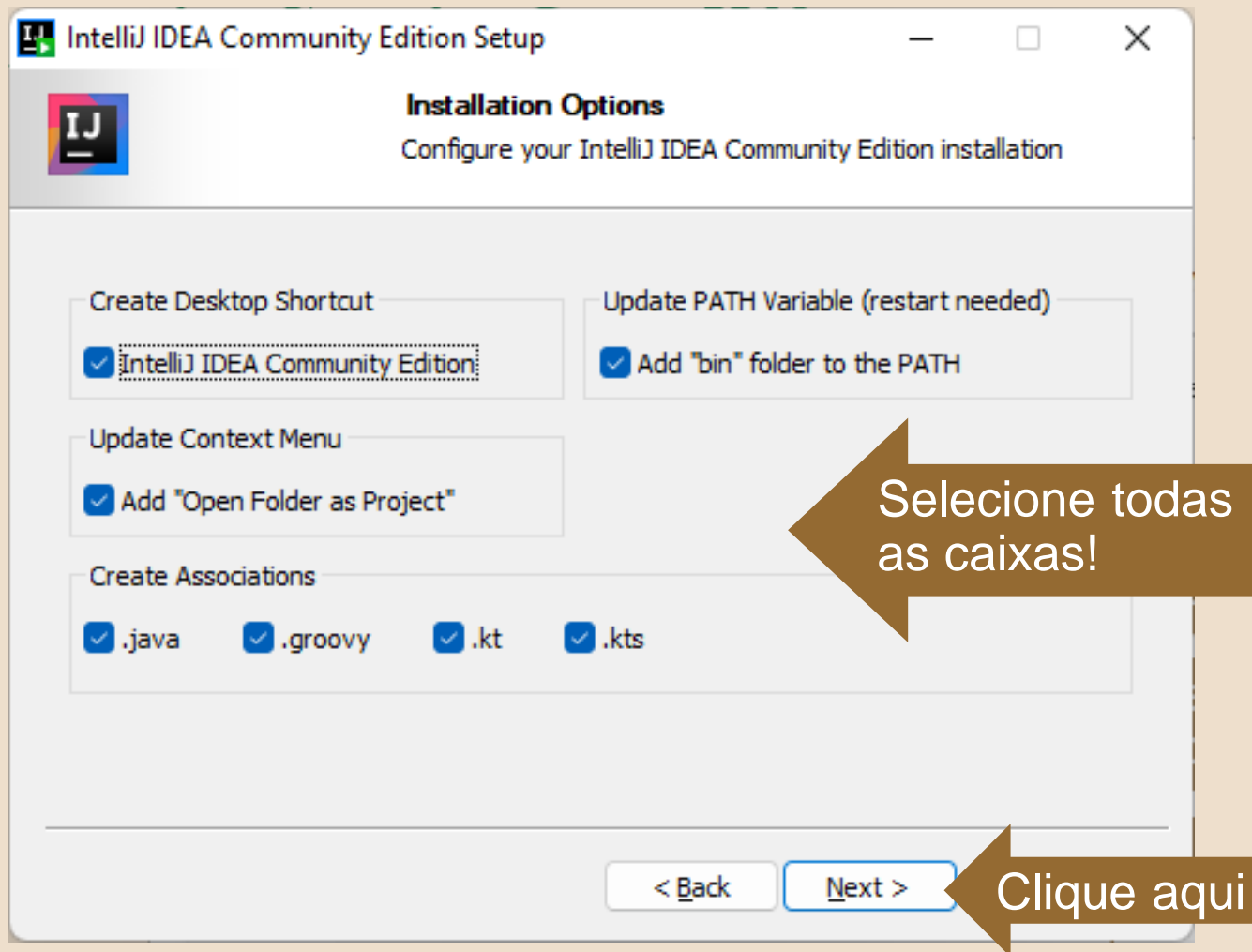
Instalação do IntelliJ



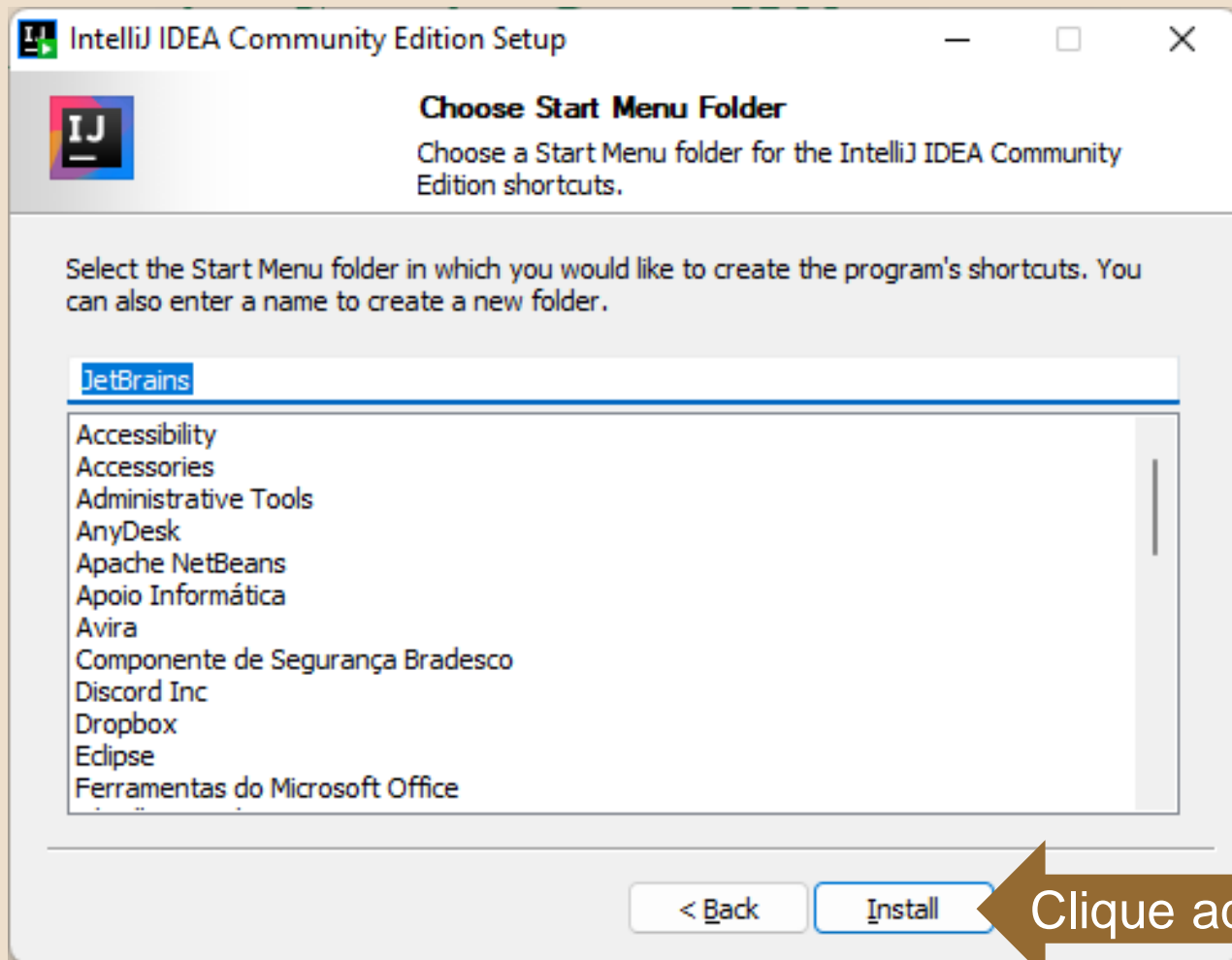
A informação desta caixa estará no contexto do seu computador e seu usuário. Não altere nada!

Clique aqui

Instalação do IntelliJ



Instalação do IntelliJ



Instalação do IntelliJ

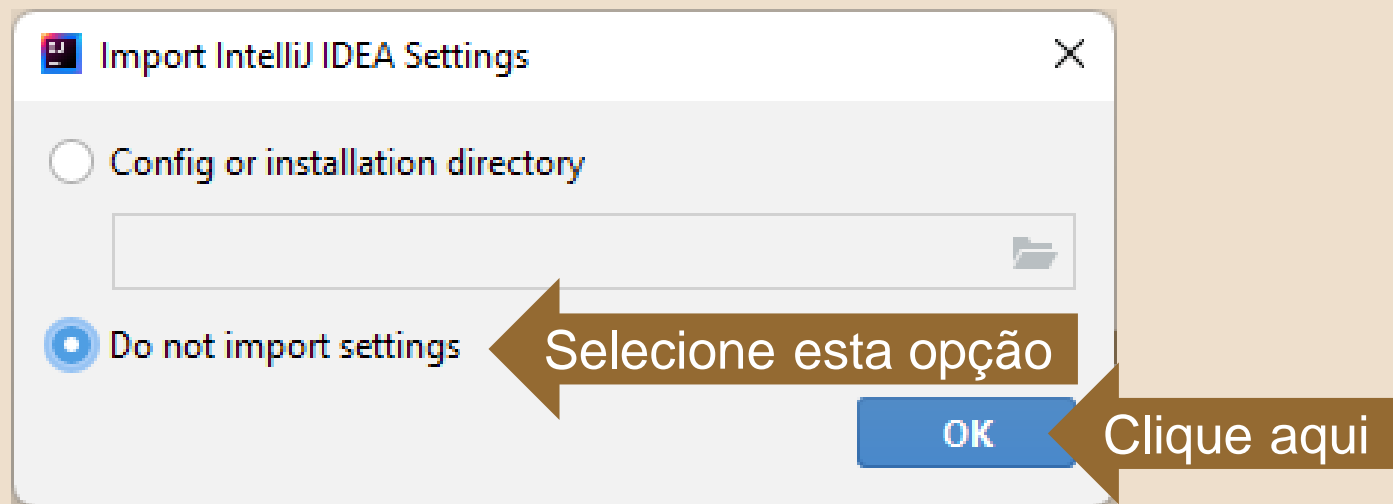


Execução (inicial) do IntelliJ

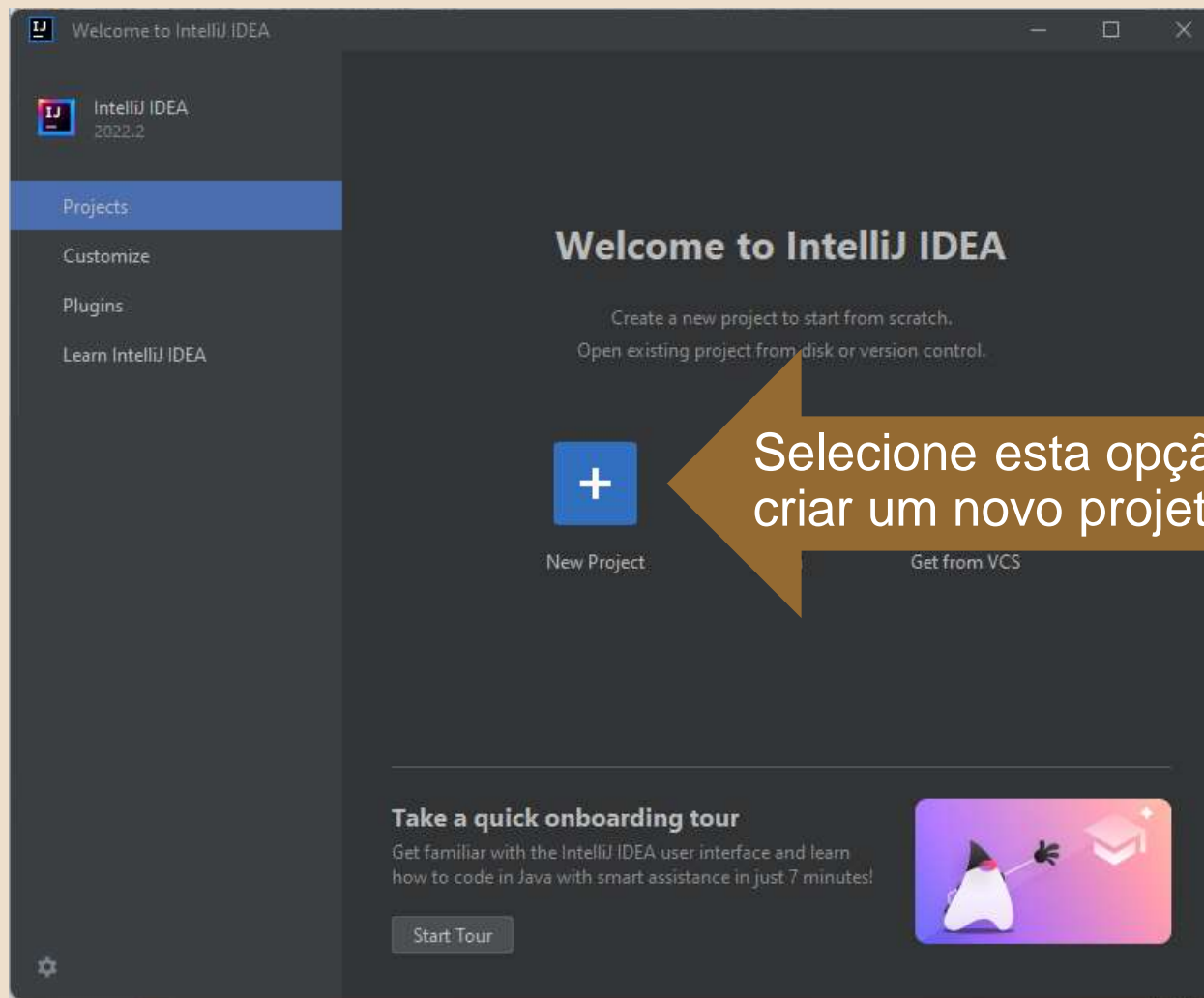


- Procure o ícone do IntelliJ na área de trabalho do seu computador e clique nele para executá-lo.

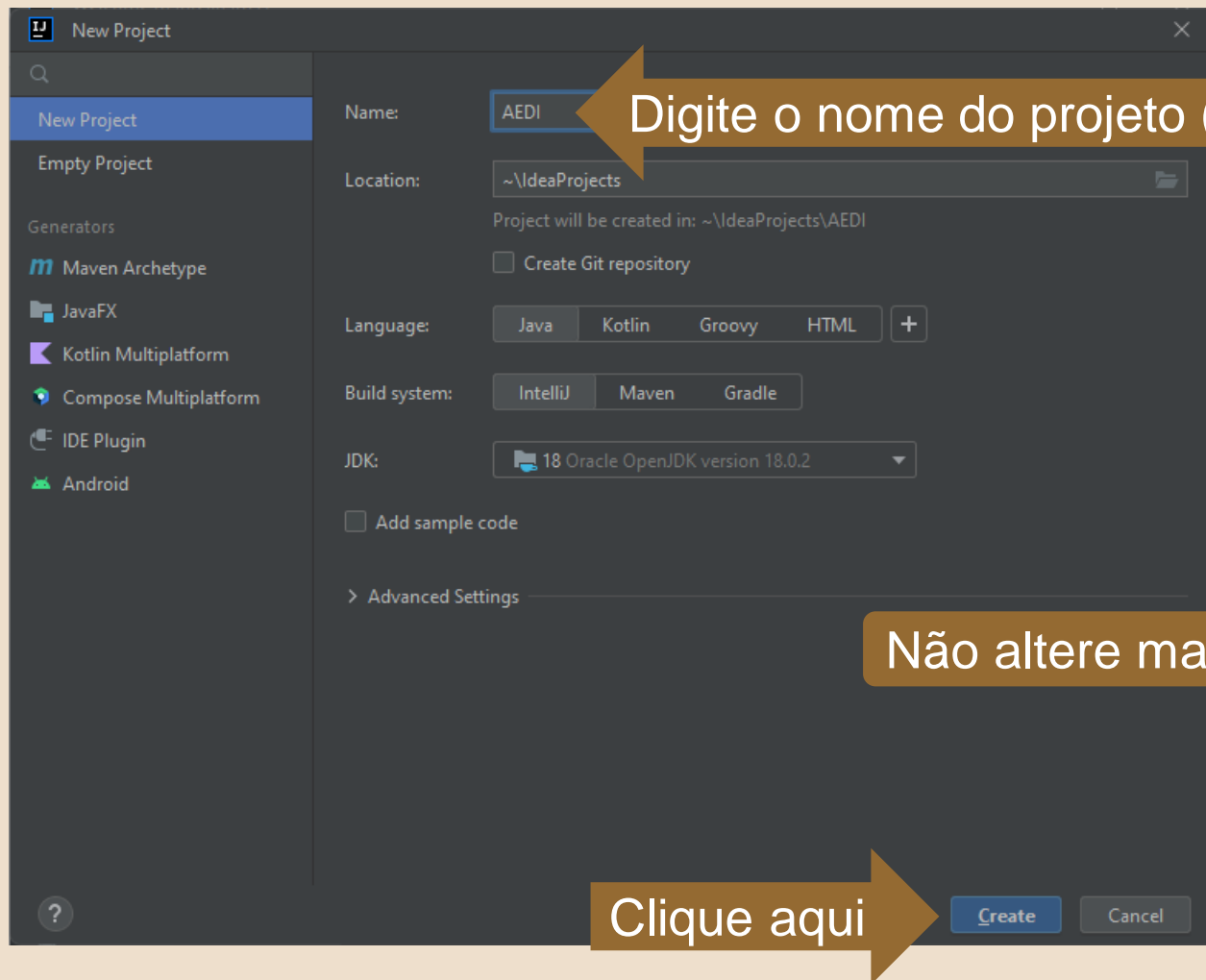
Execução (inicial) do IntelliJ



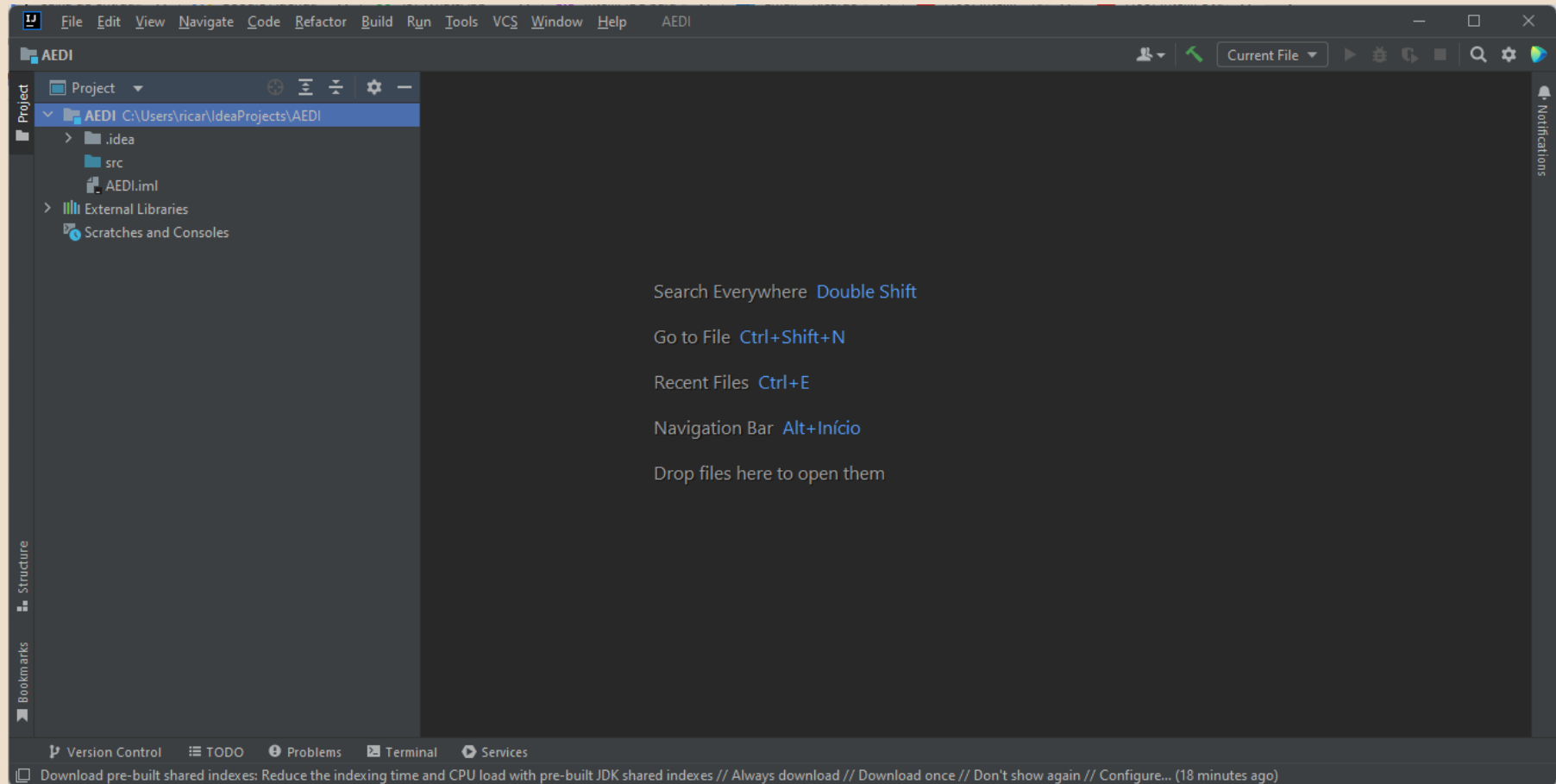
Execução (inicial) do IntelliJ



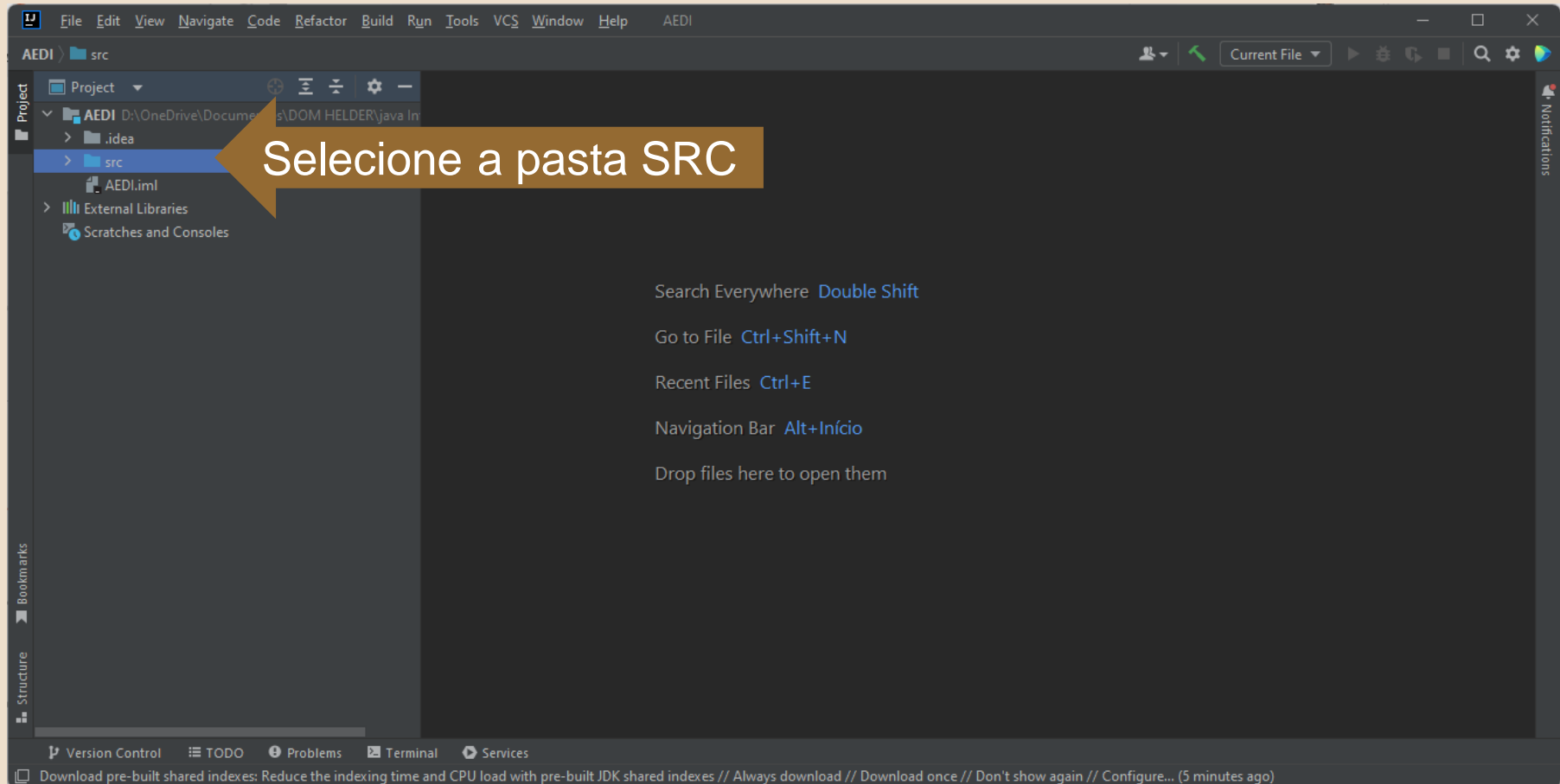
Execução (inicial) do IntelliJ



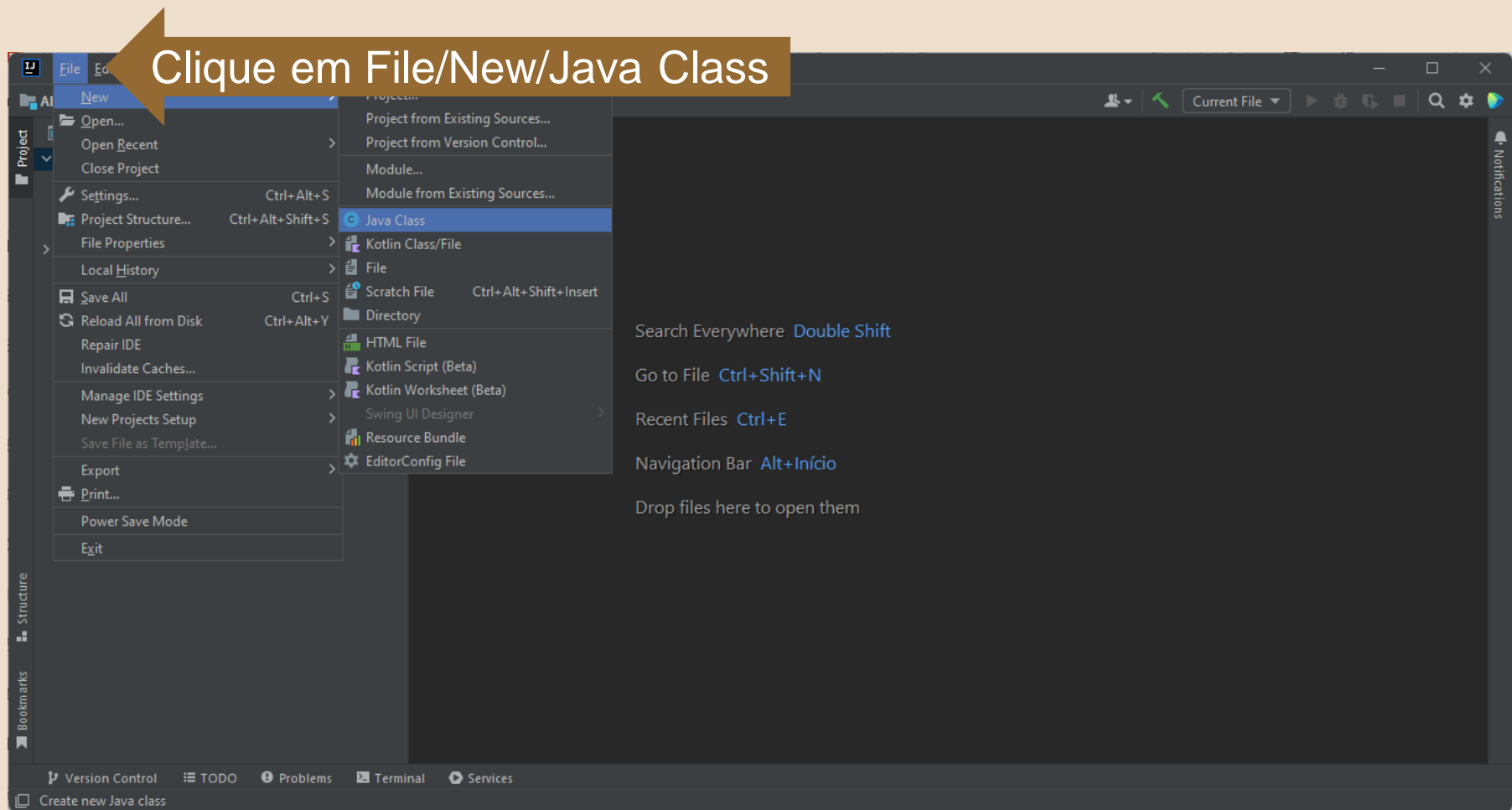
Tela principal do IntelliJ



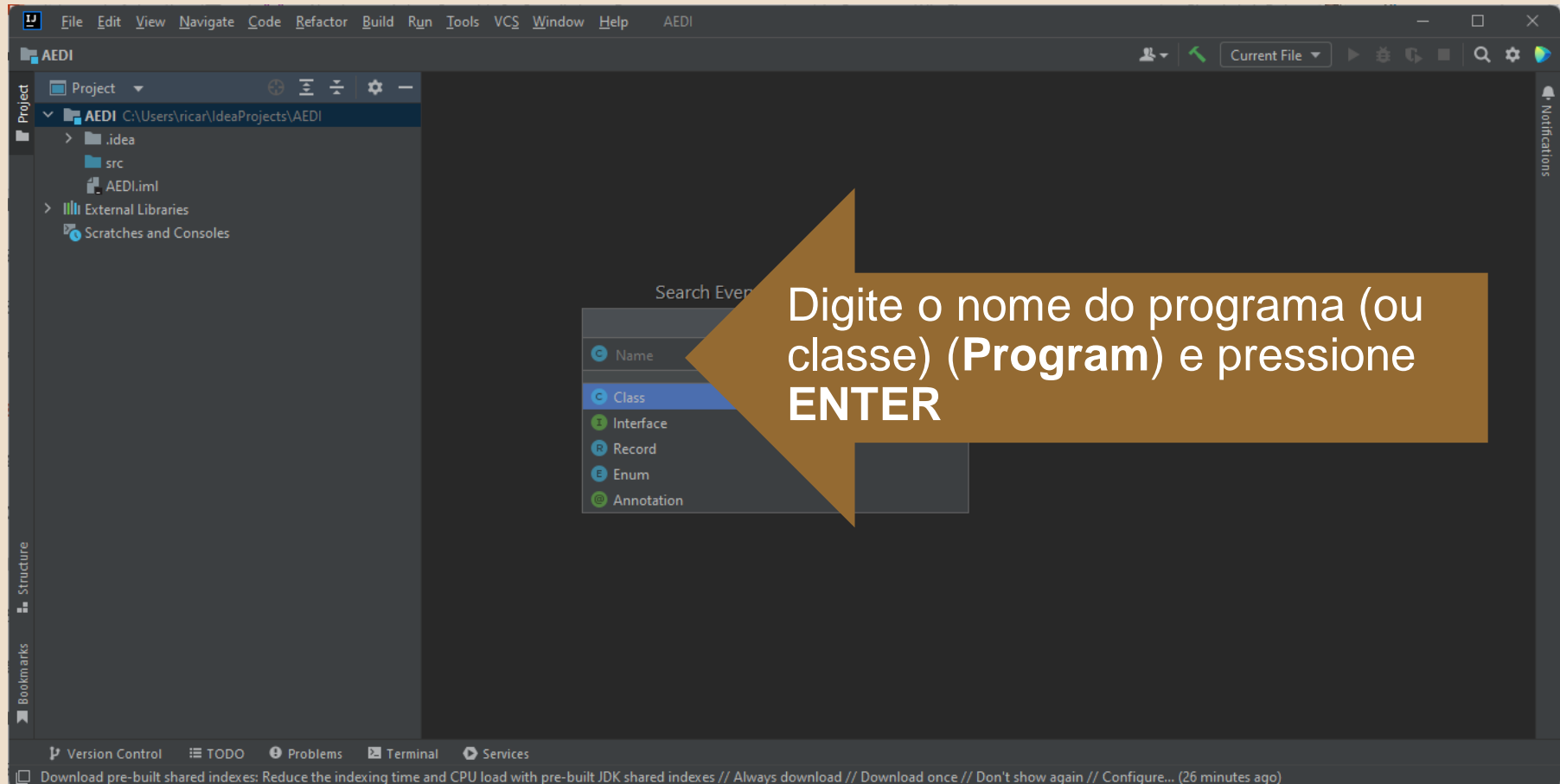
Criar programa (classe)



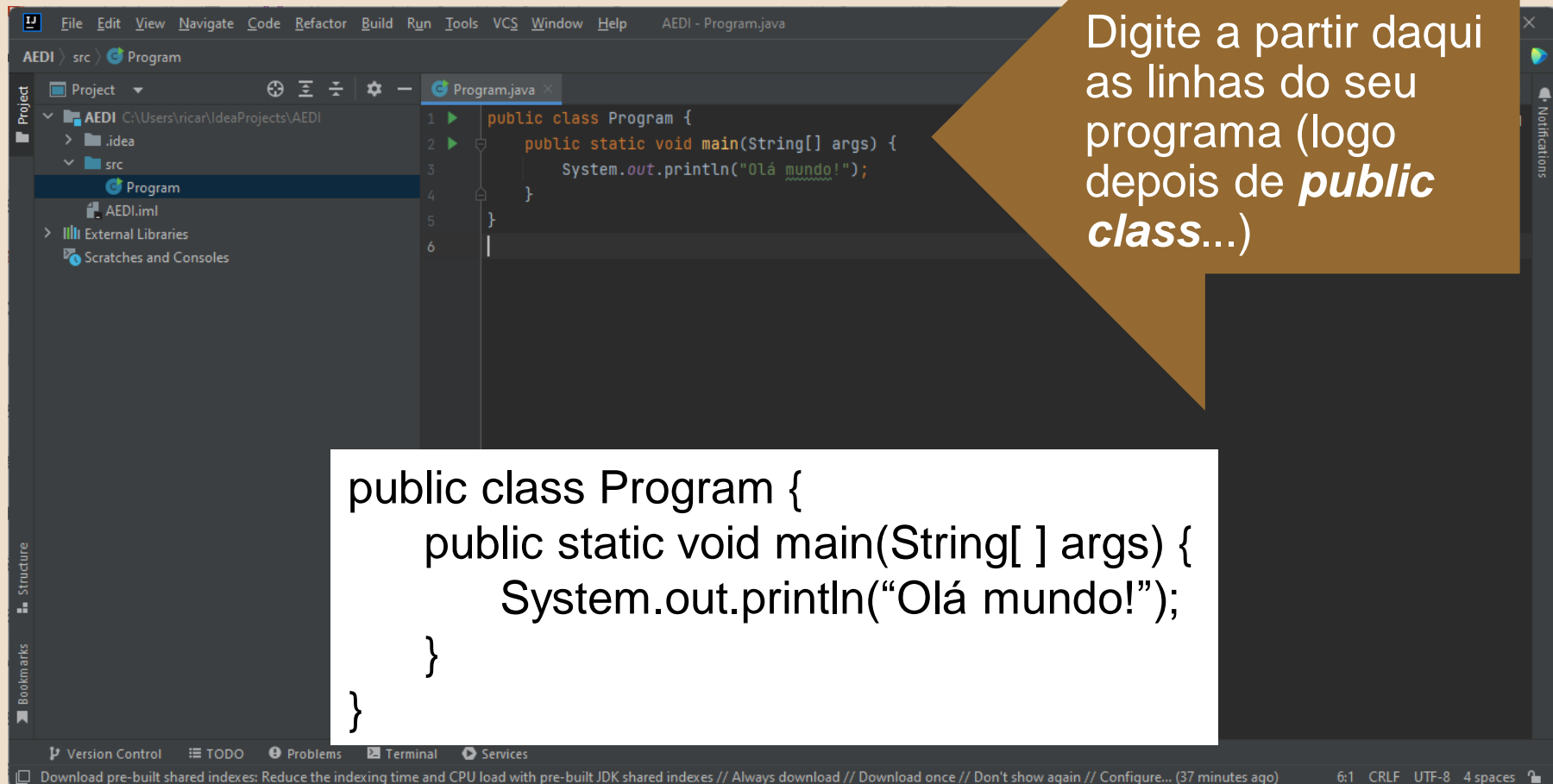
Criar programa (classe)



Criar programa (classe)



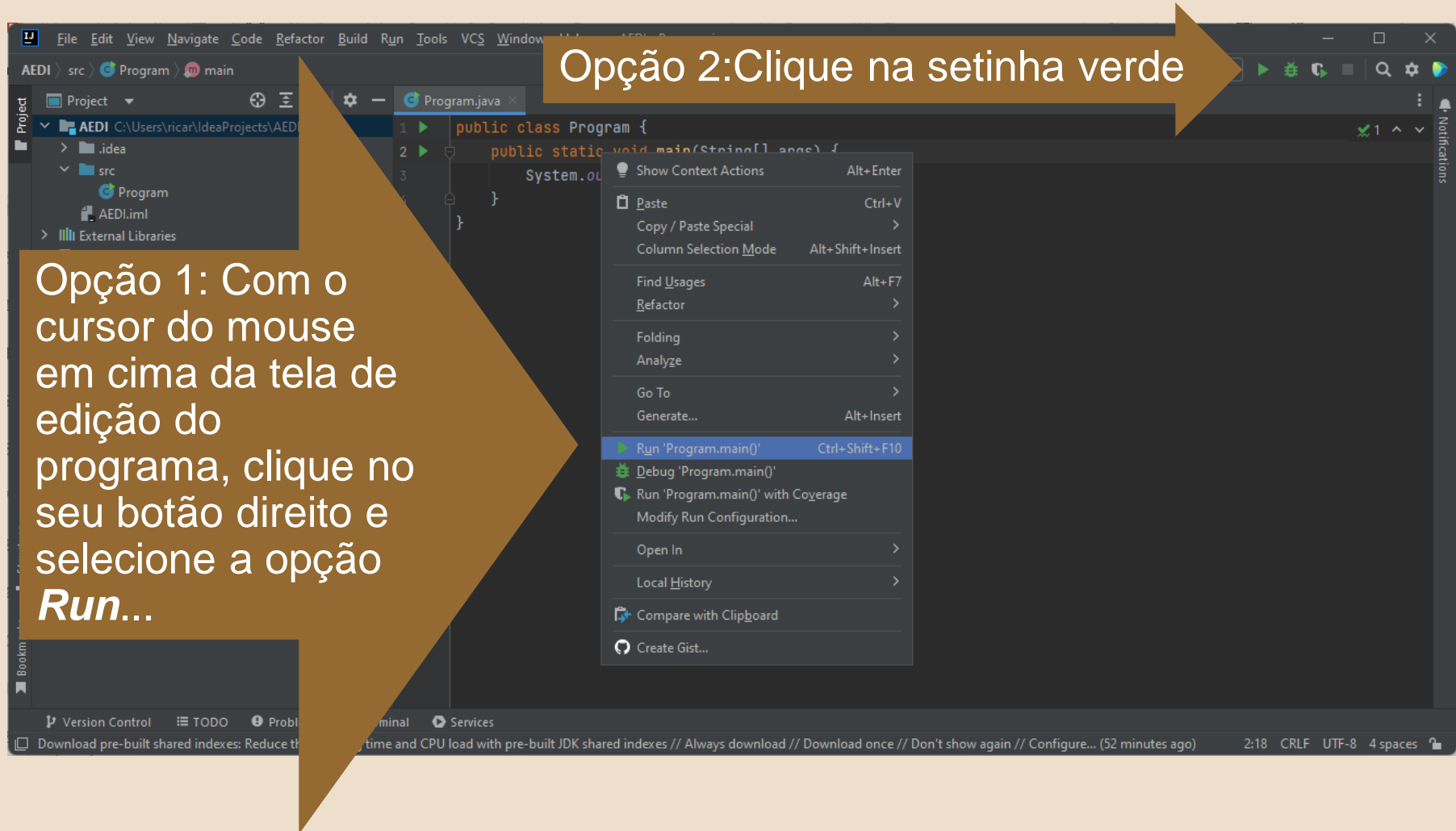
Criar programa (classe)



Digite a partir daqui as linhas do seu programa (logo depois de **public class...**)

```
public class Program {  
    public static void main(String[] args) {  
        System.out.println("Olá mundo!");  
    }  
}
```

Rodar (executar) programa (classe)

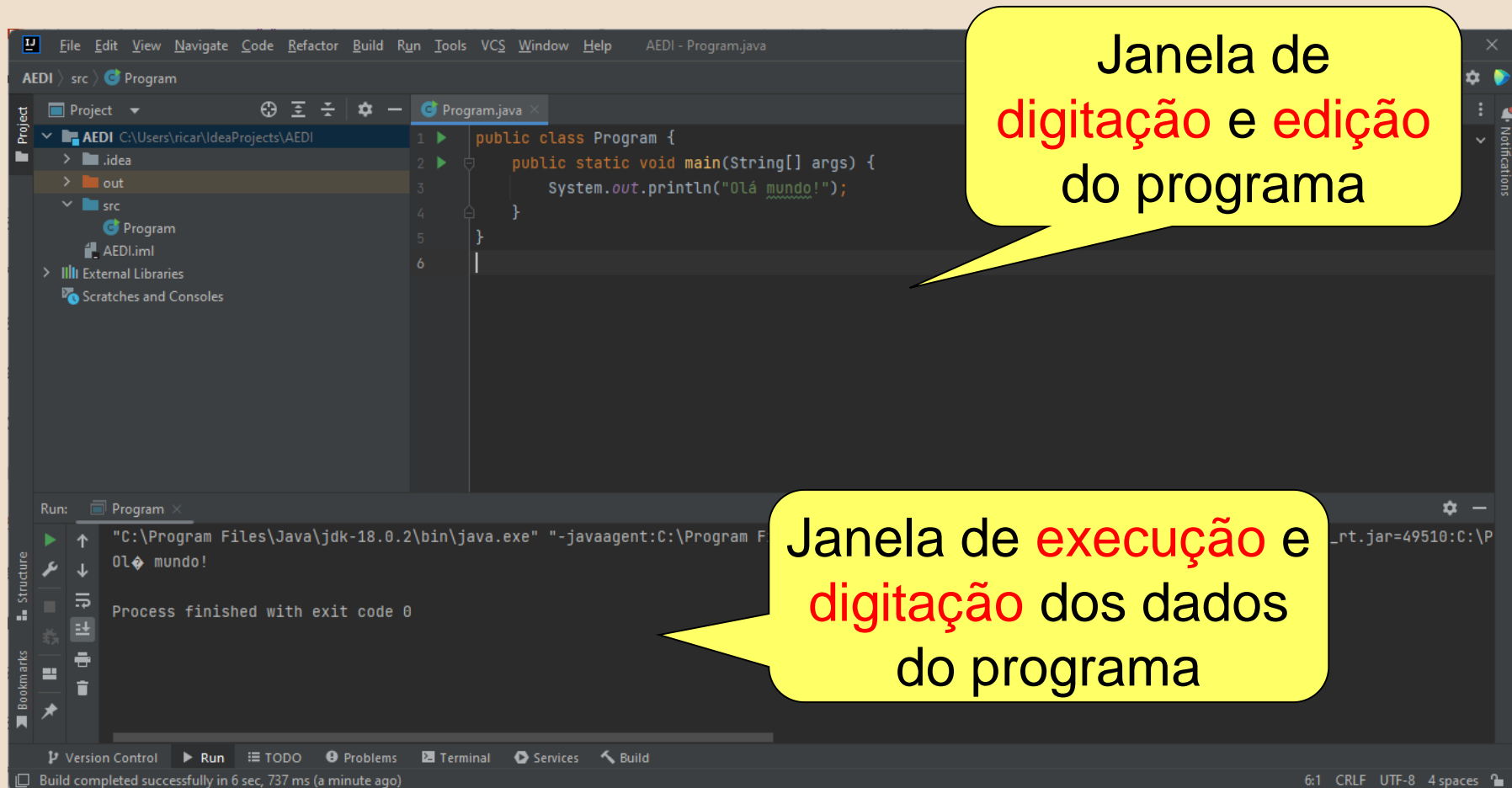


Opção 2: Clique na setinha verde

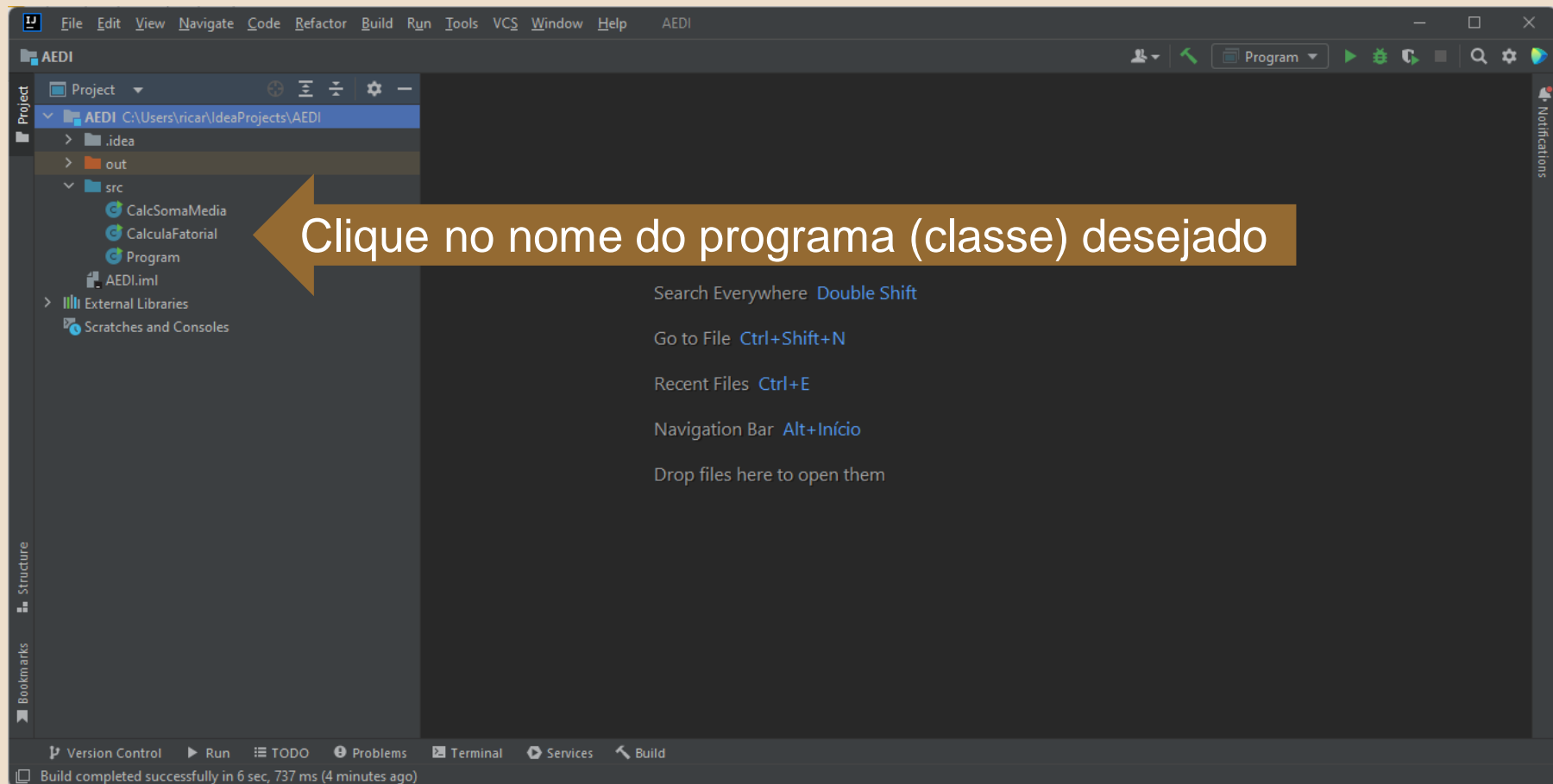
Opção 1: Com o cursor do mouse em cima da tela de edição do programa, clique no seu botão direito e selecione a opção **Run...**

```
1 public class Program {  
2     public static void main(String[] args) {  
3         System.out.println("Hello, World!");  
4     }  
5 }
```

Compilador/IDE IntelliJ



Abrir programa (classe) já criado



Compilador/IDE IntelliJ

- Na pasta “.../IdeaProjects/AEDI/src” tem os **fontes** (arquivos .Java);
- Na pasta “.../IdeaProjects/AEDI/out/production/AEDI” tem os **bytecodes** (arquivos .class);
- “...” é o disco (C: por exemplo) seguido da pasta de “Usuários” e da pasta do usuário logado no computador.

IntelliJ (dicas 1)

- Digite **Ctrl** mais **Shift** mais **A** para acessar a ajuda;
- Digite **Ctrl** mais **Alt** mais **S** para acessar as configurações (menu **File/Settings...**);
- Se você precisar criar o método **main** no modo de edição é só digitar a palavra main e depois **Enter**;
 - O método **main** transforma uma classe Java em um programa executável Java;
- No modo de edição, se digitar a palavra:
 - **sout** e depois **Enter** vira **System.out.println()**;
 - **jo** e depois **Enter** vira **JOptionPane.**;
- Digite **Alt** mais **Insert** (ou **Ins**) para inserir os métodos constructor, setter, getters, etc., automaticamente;

IntelliJ (dicas 2)

- Para aumentar ou diminuir os fontes do editor, vá até o menu **File/Settings/Editor/Font** e mude o valor da caixinha Size;
 - Marque a caixinha da opção “**Change font size with...**” em **File/Setting/Editor/General** para que se possa fazer este alteração no tamanho do fonte pressionando-se a tecla **Ctrl** mais o movimento do botão de rolagem do mouse;
- Marque a caixinha da opção “**Optimizei imports on the fly**” no menu **File/Setting/General/Auto Import** para que a importação das bibliotecas sejam inseridas automaticamente no código;
 - Exemplo: `import Java.util.Scanner;`

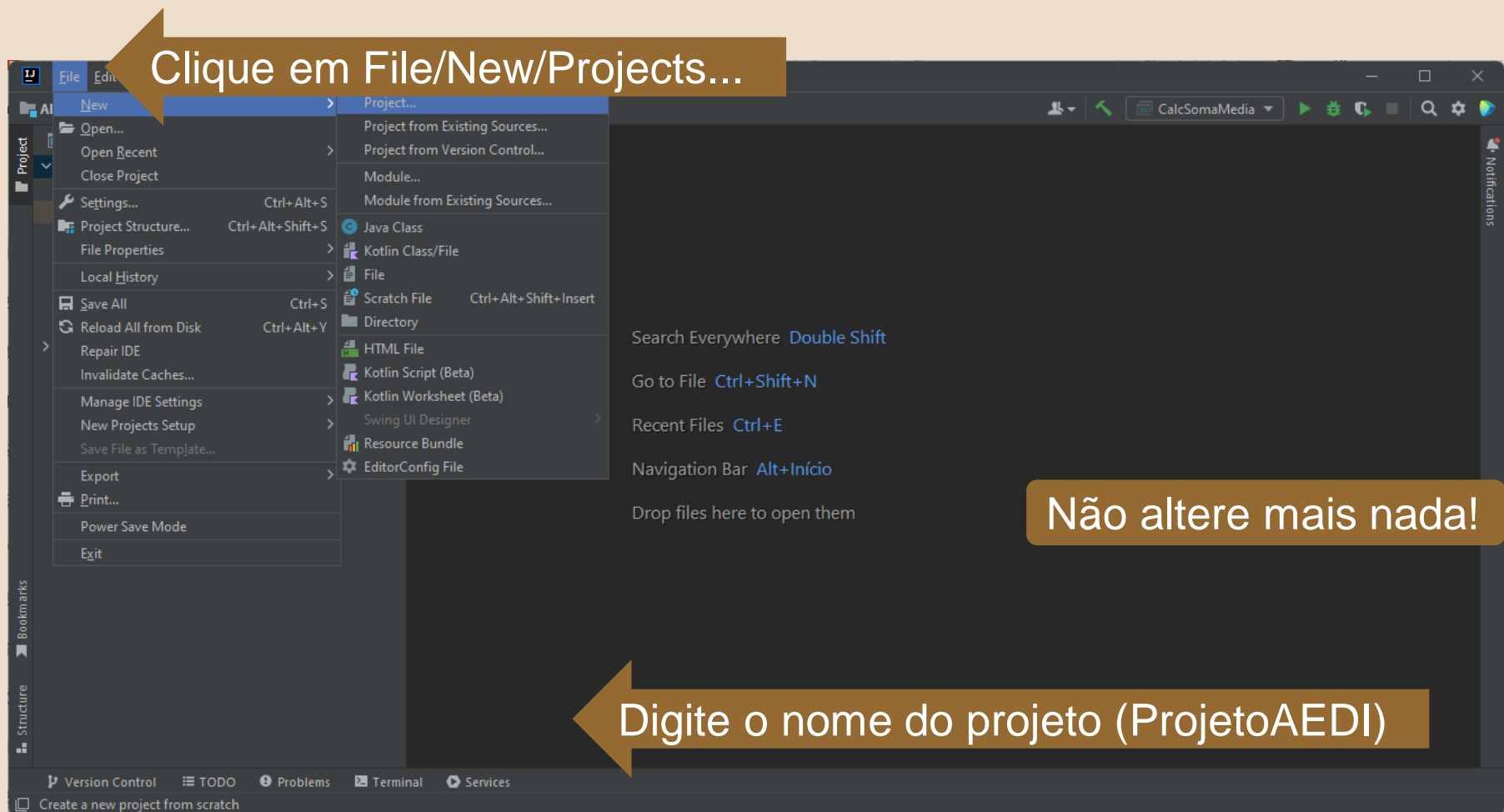
Intellij (dicas 3)

- Para utilizar caracteres acentuados, vá até o menu **File/Setting/Editor/Code Style/File Encodings** e selecione a opção “**ISO-8859-1**” nas caixas Global Encodings e Project Encodings;

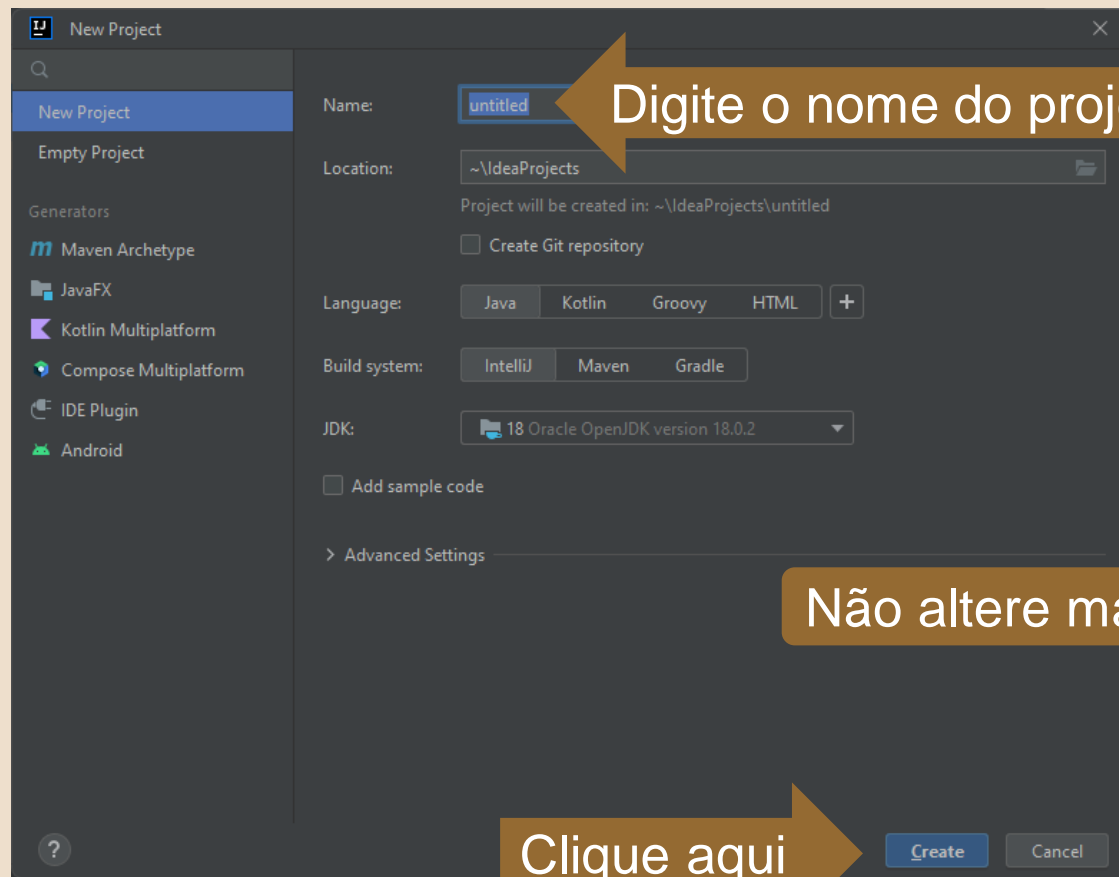
Projetos (*Projects*)

- Projetos (**projects**) são a forma utilizada no Java para agrupar os programas (classes) e/ou pacotes (**packages**) que compõe um sistema.
 - Normalmente tem-se uma classe principal (com método **main**) e várias outras classes que são chamadas a medida que o usuário vai navegando nas opções do sistema.
 - Exemplo:
 - Projeto: Sistema de Folha de Pagamento
 - Pacotes: pessoa, processos, relatórios, etc.
 - Funcionalidades (classes): cadastro de pessoal, cálculo da folha de pagamento, manutenção da tabela de IR, etc.

Criando Projetos (*Projects*) no IntelliJ



Criando Projetos (*Projects*) no IntelliJ



Digite o nome do projeto (ProjetoAEDI)

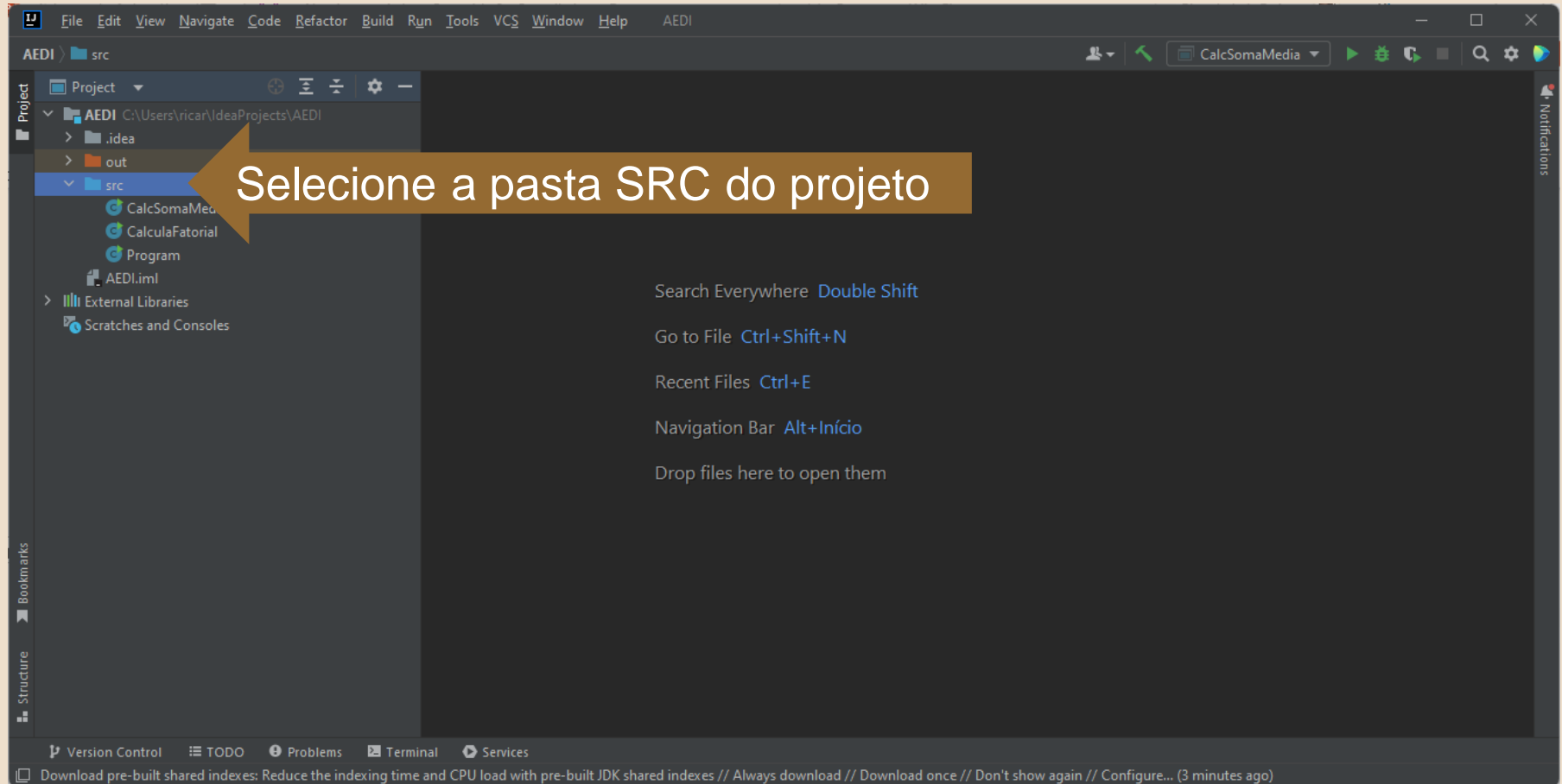
Não altere mais nada!

Clique aqui

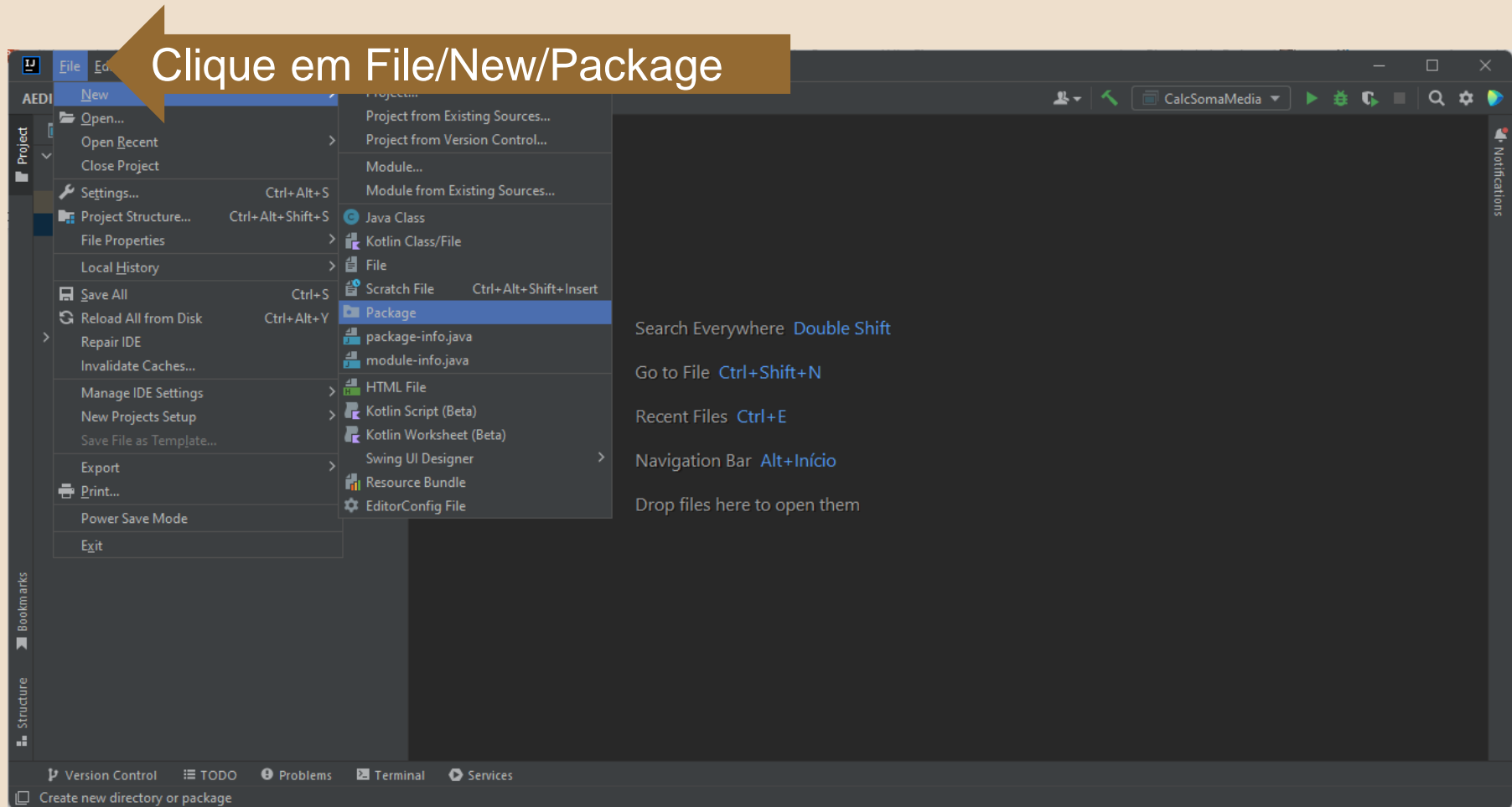
Pacotes (*Packages*)

- Pacotes (***packages***) são uma forma utilizada no Java para subagrupar os programas (classes) que compõem um projeto e que tem o mesmo objetivo ou atuam numa mesma entidade.
 - Exemplo:
 - Projeto: Sistema de Folha de Pagamento
 - Pacote 1: pessoa
 - Funcionalidades (classes): cadastros de funcionário, dependentes, fornecedores, etc.
 - Pacote 2: processos
 - Funcionalidades (classes): cálculo da folha de pagamento, manutenção da tabela de IR, etc.

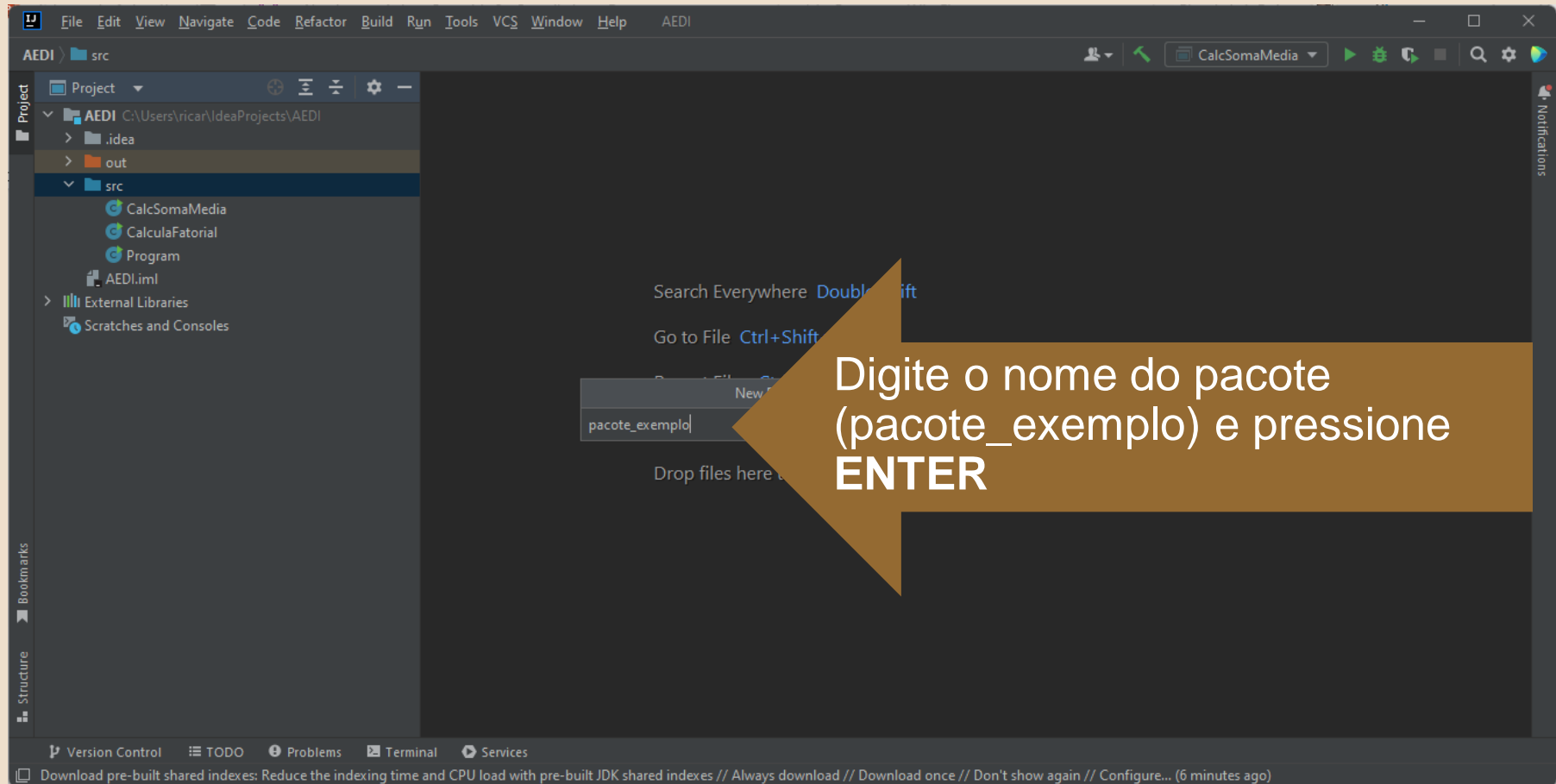
Criar Pacotes (*Packages*) no IntelliJ



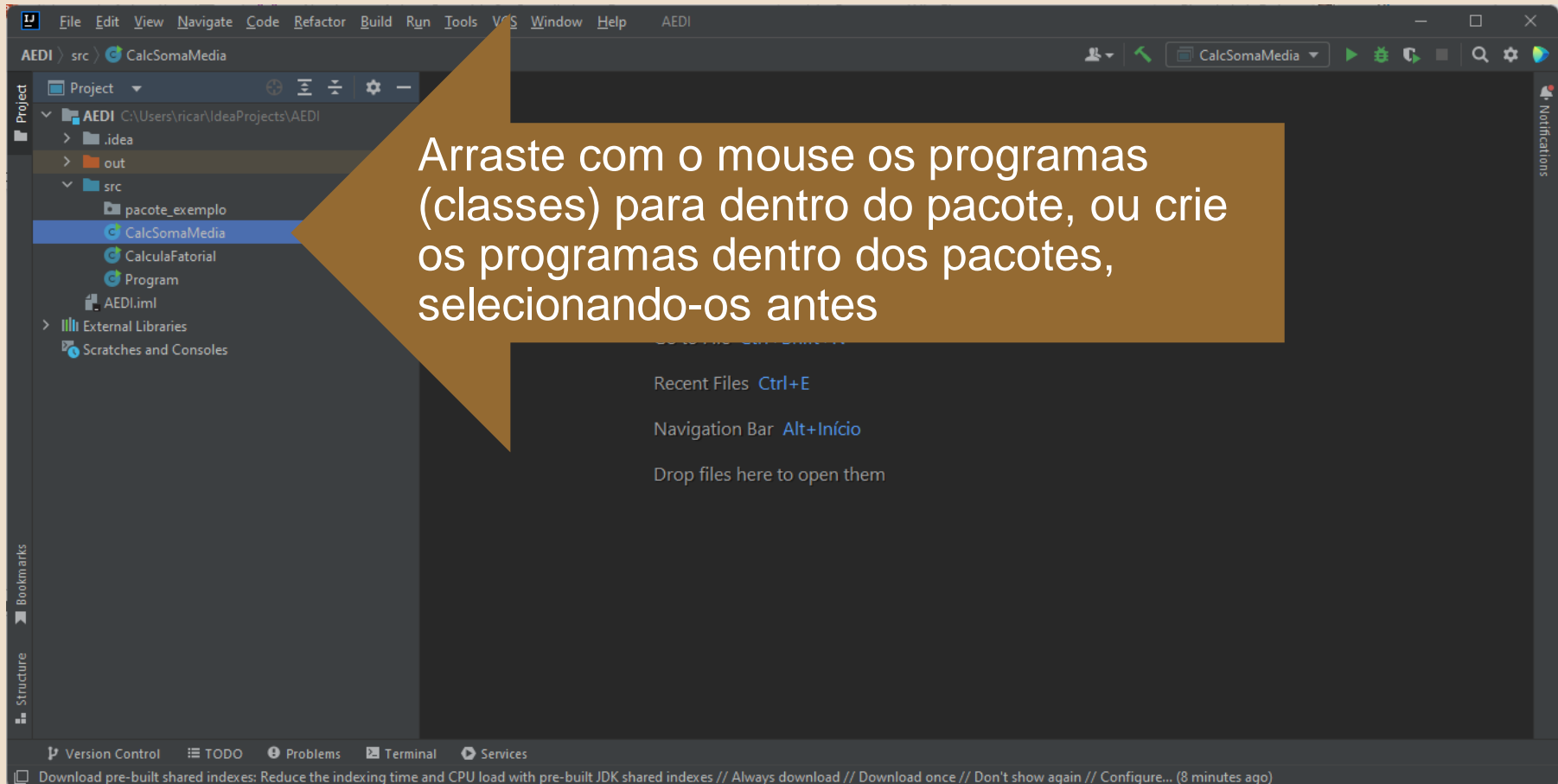
Criar Pacotes (*Packages*) no IntelliJ



Criar Pacotes (*Packages*) no IntelliJ



Criar Pacotes (*Packages*) no IntelliJ



Java (dicas)

- Para comentários utilizamos `//` no início da linha (comentários por linha), ou `/*` fechando com `*/` em outra linha (comentários em bloco);
- Utilizamos `;` (ponto e vírgula) para finalizar as linhas (declarações, comandos, etc.).
- Na entrada de dados via console (método Scanner) utilizamos a vírgula (,) como separador de casas decimais;
 - JOptionPane: ponto (.);
 - Arquivo txt: vírgula (,).

Estrutura básica de um Programa no Java

```
// Classes (pacotes) externas (SE NECESSÁRIO)
import ???;
// Classe principal (NOME INTERNO = NOME EXTERNO)
public class NomePrograma {
    // Função: ??? (OPCIONAL)
    // Autor: Ricardo Luiz de Freitas (OBRIGATÓRIO)
    // Declarações de variáveis globais (SE NECESSÁRIO)
    ???;
    // Método que torna a classe executável
    public static void main(String[] args) {
        // Declarações de variáveis locais
        ???;
        // Comandos do programa (método)
        ???;
    } // Fechar o método main
} // Fechar a classe principal
```

Exemplo 5

```
import java.util.Scanner;
public class CalcSomaMedia {
    // Função: Calcula a média de 3 números inteiros
    // Autor: Seu nome completo
    public static void main(String[] args) {
        long n1, n2, n3, soma;
        double media;
        Scanner teclado = new Scanner(System.in);
        System.out.print("Informe um número: ");
        n1 = teclado.nextLong();
        System.out.print("Informe outro número: ");
        n2 = teclado.nextLong();
        System.out.print("Informe mais um número: ");
        n3 = teclado.nextLong();
        soma = n1 + n2 + n3;
        media = soma / 3.0;
        System.out.println("Média = " + media);
        teclado.close();
    }
}
```

Indentação de Código

- Organizar/estruturar o código do algoritmo;
- Facilitar a legibilidade do código;
- Tornar a interpretação do código mais fácil aos olhos de um terceiro;
- **Mostrar a hierarquia entre as partes e/ou comandos do código.**

Exemplo 6

Código SEM indentação

```
import java.util.Scanner;
public class Somatorio {
public static void main(String[] args) {
int numero, soma = 0, cont = 0, rep;
double media;
Scanner teclado = new Scanner(System.in);
System.out.print("Quantos números serão informados? ");
rep = teclado.nextInt();
for (int aux=1; aux<=rep; aux++) {
System.out.print("Informe um número inteiro: ");
numero = teclado.nextInt();
if (numero >= 1 && numero <= 100) {
soma += numero;
cont++;
}
}
media = (float) soma / cont;
System.out.println("A média deu "+media);
teclado.close();
}
}
```

Exemplo 7

Código COM indentação

```
import java.util.Scanner;

public class Somatorio {
    public static void main(String[] args) {
        int numero, soma = 0, cont = 0, rep;
        double media;
        Scanner teclado = new Scanner(System.in);
        System.out.print("Quantos números serão informados? ");
        rep = teclado.nextInt();
        for (int aux=1; aux<=rep; aux++) {
            System.out.print("Informe um número inteiro: ");
            numero = teclado.nextInt();
            if (numero >= 1 && numero <= 100) {
                soma += numero;
                cont++;
            }
        }
        media = (float) soma / cont;
        System.out.println("A média deu "+media);
        teclado.close();
    }
}
```

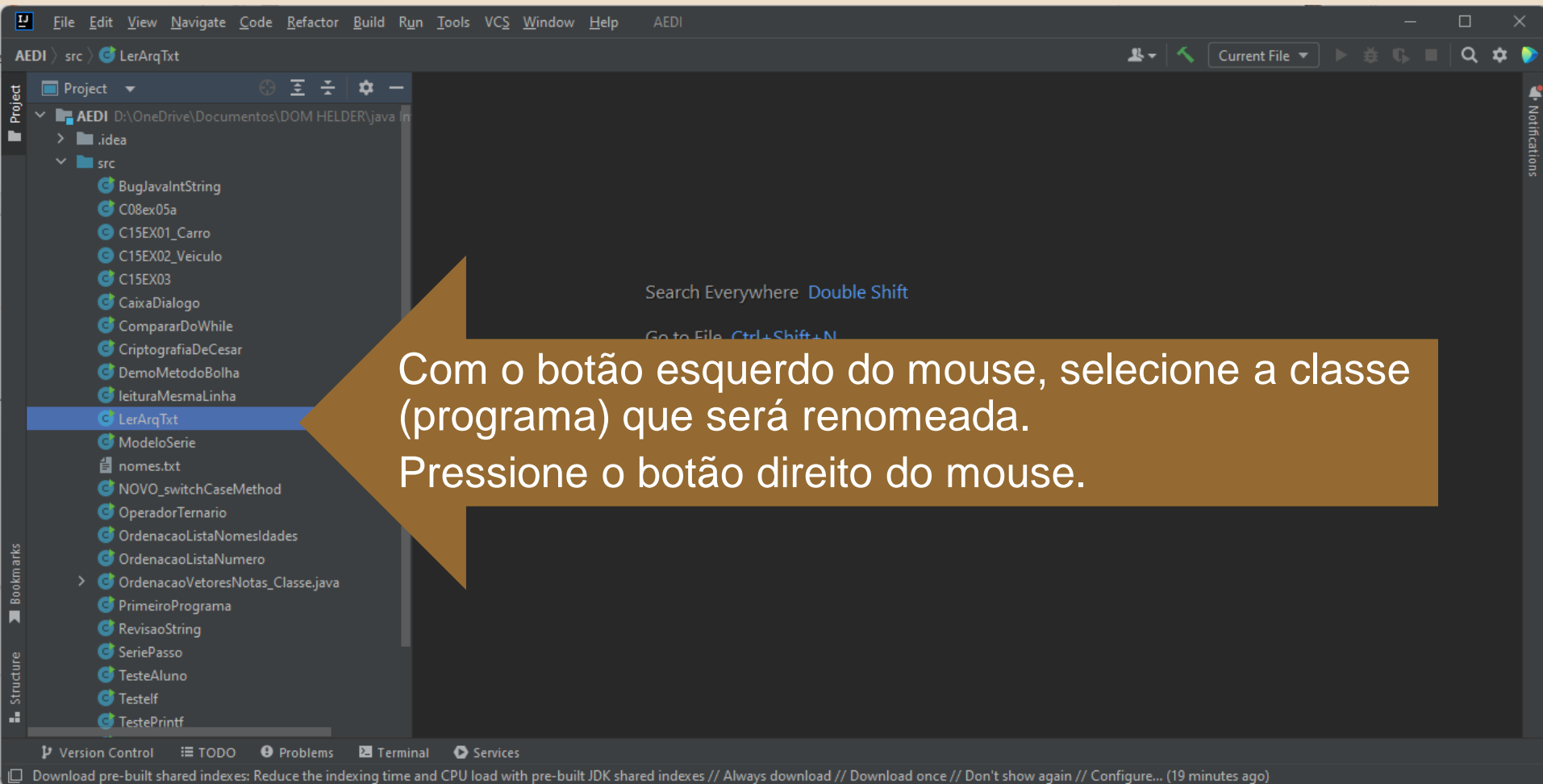
Indentação de Código

- A maioria das IDEs fazem a indentação do código automaticamente.
 - É o caso do IntelliJ.
- No IntelliJ, para fazer manualmente (ao longo ou ao final da digitação do código) pressione simultaneamente as teclas:
 - ***Ctrl Alt L***

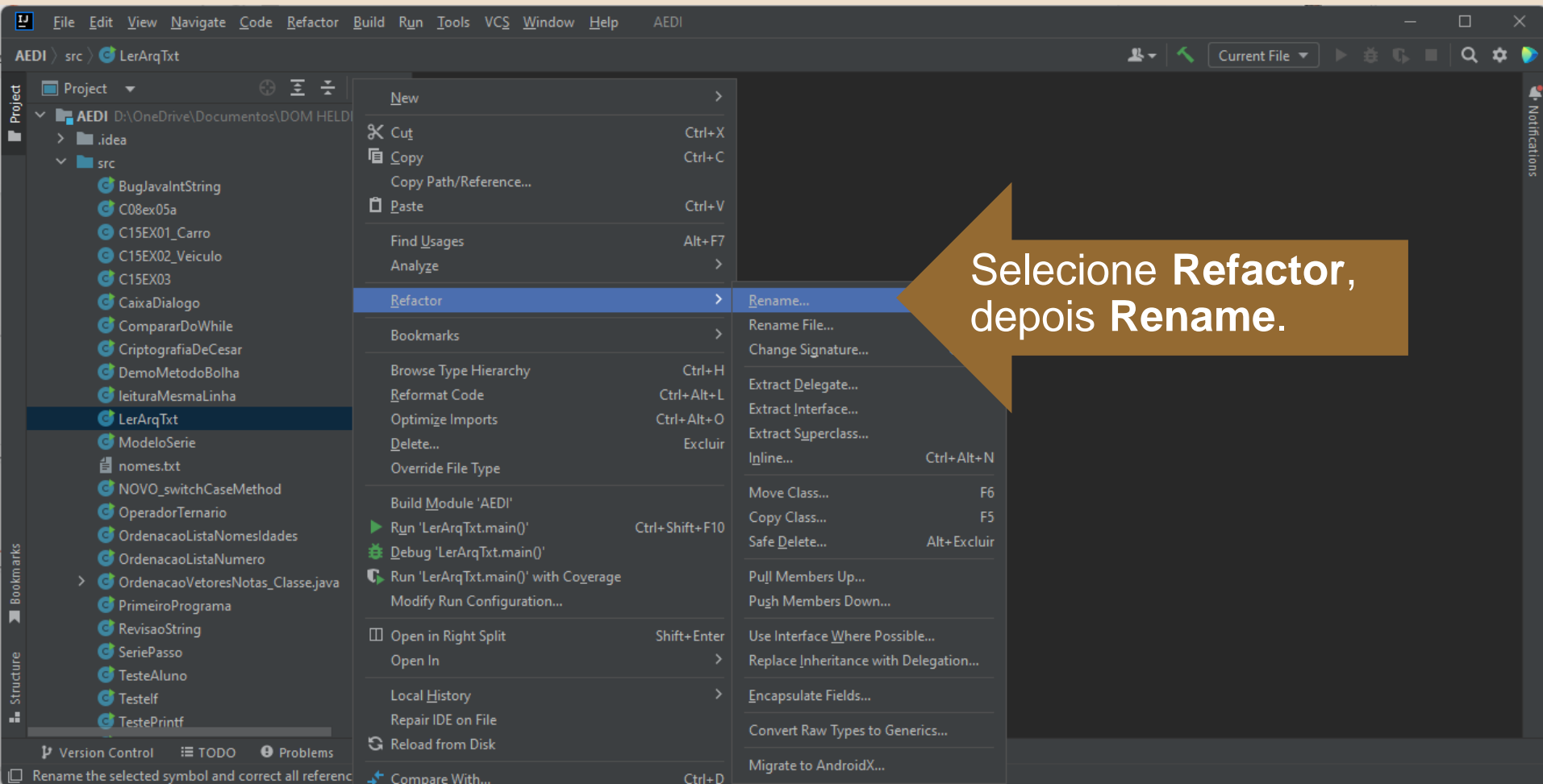
Nomenclatura dos exercícios

- Ao criar os programas (classes) em Java dos exercícios adote o padrão de nome conforme abaixo:
 - CaaEXbb
 - aa = número do conteúdo (ppt);
 - bb = número do exercício.
 - Exemplo: C03EX04
 - Conteúdo 3;
 - Exercício 4.

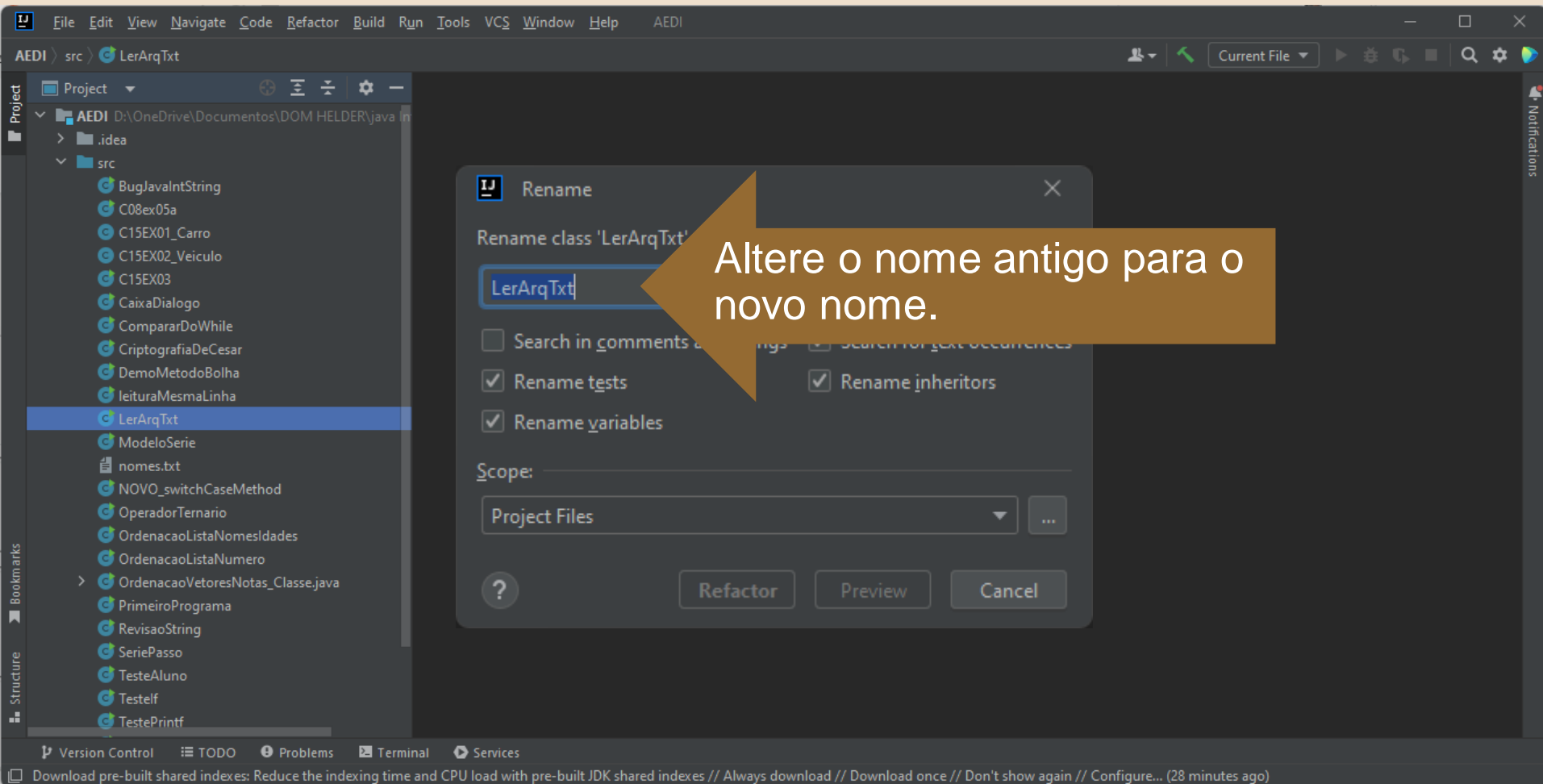
Renomear classes no IntelliJ



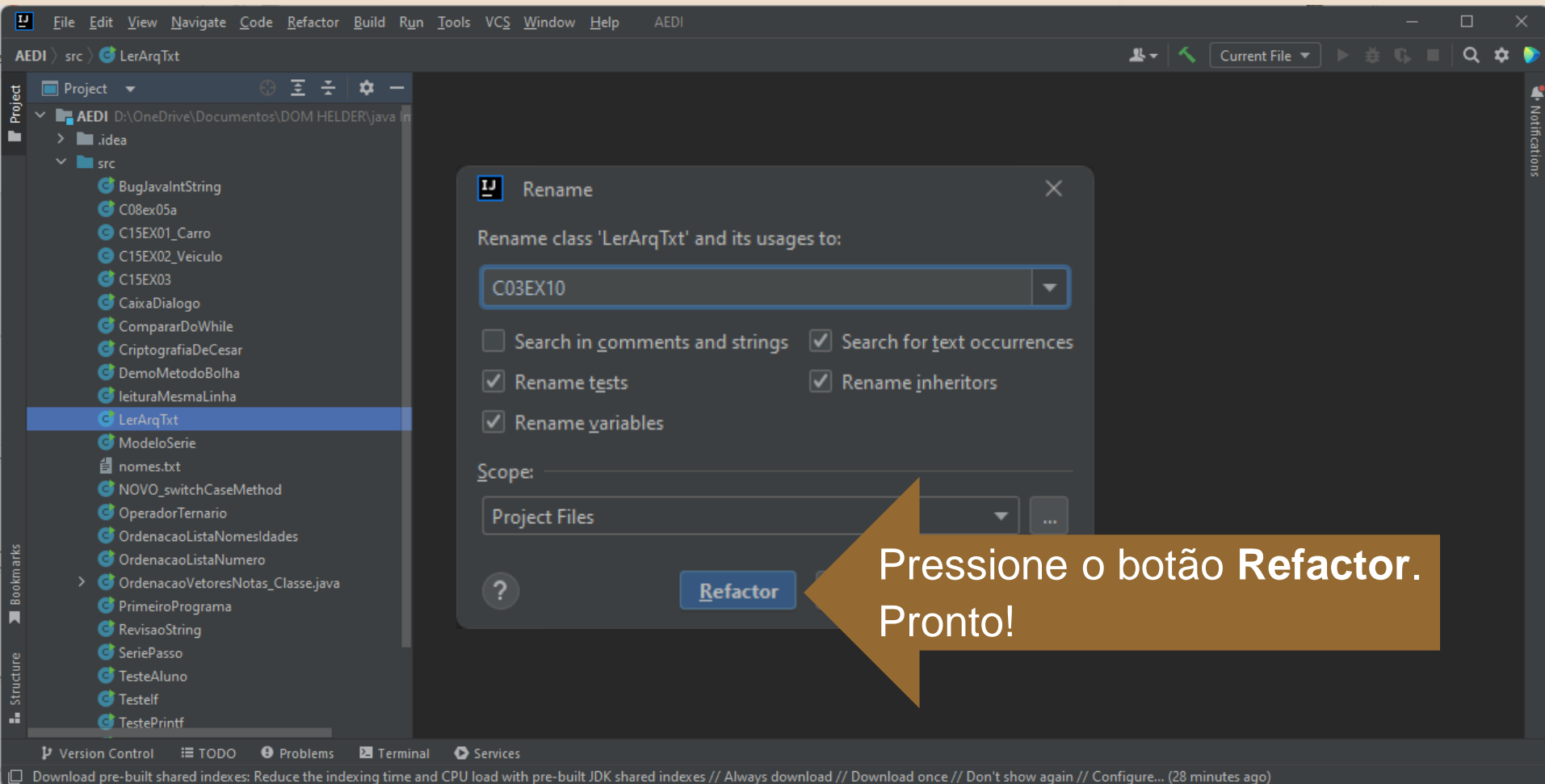
Renomear classes no IntelliJ



Renomear classes no IntelliJ



Renomear classes no IntelliJ



Exercício 1

- Baixe e execute os instaladores do Java e do IntelliJ no seu computador conforme as instruções dos slides anteriores (FAZER EM CASA, SE AINDA NÃO O FEZ).
- Abra o IntelliJ:
 - Digite nele o programa Java do exemplo 4;
 - Execute o programa;
 - Conserte os erros de sintaxe que por ventura apareçam devido a erros de digitação;
 - Entre com o dado 5 para o campo solicitado;
 - Verifique se o resultado será 120, não sendo, volte ao editor.

Exercício 2

- Baixe e execute os instaladores do Java e do IntelliJ no seu computador conforme as instruções dos slides anteriores (FAZER EM CASA, SE AINDA NÃO O FEZ).
- Abra o IntelliJ:
 - Digite nele o programa Java do exemplo 5:
 - Execute o programa;
 - Conserte os erros de sintaxe que por ventura apareçam devido a erros de digitação;
 - Entre com os dados 8, 12 e 63 para cada campo solicitado;
 - Verifique se o resultado será 27,66666..., não sendo, volte ao editor.

Exercício 3

- Baixe o arquivo “CalculaHipotenusa.Java” que está em “Arquivos JAVA dos exemplos dos PPTs de conteúdos das aulas” (programa fonte em Java) da pasta Informações da disciplina no Moodle:
 - Baixe na pasta SRC do projeto AEDI;
- Abra o IntelliJ:
 - Renomeie (Refactor) o programa para o padrão dos exercícios;
 - Execute o programa;
 - Entre com os dados 10 e 15 para cada campo solicitado;
 - Verifique se o resultado será 18,0277..., não sendo, volte ao editor.

Dados

- O dado é a própria essência de um algoritmo ou programa de computador.
- Ele pode ser informado pelo usuário, gerado pelo programa, processado pelo computador, e impresso para o usuário como uma informação.

Tipos de dados

- Tipos de dados são maneiras pelas quais os dados são tratados
 - **Constantes:** são aqueles dados cujos valores **não se alteram** durante a execução do algoritmo
 - **Variáveis:** são aqueles dados cujos valores **podem ser alterados** durante a execução do algoritmo

Tipos de dados

■ Classificação

□ Simples

■ Numéricos

- Sem casa decimal (**byte**, **short**, **int**, **long**)
- Com casa decimal (**float**, **double**)

■ Não Numéricos

- Caracteres (**char**, **String**)
- Lógico (**boolean**)

□ Estruturados

- Listas estáticas (Vetores/Matrizes)
- Listas dinâmicas
- Arquivos (textos, bancos de dados)

Tipos de dados

■ Numéricos

□ Sem casa decimal (**byte**, **short**, **int**, **long**)

- São os valores numéricos inteiros (sem casas decimais)
- Podem ser valores positivos ou negativos
- Ex: 0, -2, 47, 58, 123220, -34235

□ Com casa decimal (**float**, **double**)

- São os valores numéricos que possuem casas decimais
- Podem ser valores positivos ou negativos
- Ex: 0.12, -45.17, 3.14159, 0.000001

Tipos de dados

■ Não Numéricos

□ Caracteres (**char**, **String**)

- Representadas por um (**char**) ou uma sequência de caracteres (**String**) (letras, números ou caracteres especiais)
- Textos em geral
- Geralmente são delimitadas por aspas (“ ”)
- Ex: “F”, “Rua”, “Computação”, “20/10/1998”

□ Lógico (**boolean**)

- Representadas por apenas 2 valores:
 - Verdadeiro (true)
 - Falso (false)

Declaração de variáveis

- Os tipos de dados mais utilizados no Java são:

Tipo	Valores que podem assumir	Tamanho em bytes
byte	de -128 até 127	1
short	de -32.768 até 32.767	2
int	de -2.147.483.648 a 2.147.483.647	4
<u>long</u>	de -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807	8
float	de -3.4×10^{38} a 3.4×10^{38}	4
<u>double</u>	de -1.7×10^{308} a 1.7×10^{308}	8
char	armazena apenas um caracter	1
<u>String</u>	armazena um conjunto de caracteres	cada caracter = 1 byte
boolean	true ou false	1 bit

Identificadores

- Todas as variáveis ou constantes são identificadas por um nome que chamamos de **IDENTIFICADOR**
- Exemplos:
 - salario
 - idade
 - j
 - xpto
 - nomeVariavel

Identificadores

- Regras para formação de identificadores:
 - Devem começar OBRIGATORIAMENTE com uma LETRA ou com os caracteres sublinhado (_) ou cifrão (\$);
 - Só podem conter LETRAS, NÚMEROS, caracteres sublinhado (_) ou cifrão (\$);
 - **Não podem conter** ESPAÇOS ou caracteres especiais (@, ?, !, -, caracteres acentuados, c com cedilha, etc.);
 - O Java é *case sensitive*, ou seja, distingue maiúsculas de minúsculas;
 - **Nomes válidos:** `i, j, FRUTA, salario, $a12, i23ER21, um_nome_qualquer, nomePai, A$1`
 - **Nomes inválidos:** `1, 25salario, nome-fruta, 5JOSE, nome do pai, média`

Identificadores

- ❑ O identificador deve passar ao programador (ou quem está lendo o programa/algoritmo) a origem do dado que a variável ou constante vai armazenar, ou seja, não devemos criar nomes que não tenham nada a ver com o conteúdo da variável ou constante;
- ❑ É padrão no Java começar com letra minúscula um identificador de variável:
 - **Nomes inapropriados:** `i, j, a12, i23ER21, xyz, xibobo, ricardo`
 - **Nomes apropriados:** `idade, salario, nomeFruta, contador, endereco_correspondencia`

Exemplo 8

(identificadores **inapropriados**)

```
import java.util.Scanner;

public class X {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        long a, b;
        a = teclado.nextLong();
        b = 1;
        for (int c=2; c<=a; c++)
            b = b * c;
        System.out.println(b);
        teclado.close;
    }
}
```

Exemplo 9

(identificadores **apropriados**)

```
import java.util.Scanner;

public class CalculaFatorial {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        long numero, fatorial;
        System.out.println("Informe um número: ");
        numero = teclado.nextLong();
        fatorial = 1;
        for (int contador=2; contador<= numero; contador++)
            fatorial = fatorial * contador;
        System.out.println("Fatorial = "+fatorial);
        teclado.close;
    }
}
```

Variáveis

(abreviação de dados variáveis)

■ O que é uma Variável?

- ❑ Em termos lógicos é o registro formal, dentro do programa, dos dados a serem utilizados pelo próprio programa;
- ❑ Em termos físicos é uma posição na memória do computador reservada pelo programa, durante a sua execução, para armazenar dados a serem utilizados pelo próprio programa;
- ❑ Estes dados podem ter sido gerados pelo próprio programa (comandos de repetição, comando de atribuição, etc.) ou inseridos pelo usuário quando da execução de um comando de entrada/leitura de dados.

Declaração de variáveis

- Os tipos de dados mais utilizados:
 - ❑ **byte, short, int, long**: define variáveis do tipo inteiro, ou seja, sem casas decimais.
 - ❑ **float, double**: define variáveis do tipo real, ou seja, com casas decimais. (ex: `3.14`); >> Aspa simples
 - ❑ **char**: define variáveis do tipo caracter, ou seja, um caracter. (ex: `'A'`, `'9'`, `'@'`); >> Aspa simples
 - ❑ **String**: define variáveis do tipo caracter, ou seja, um ou uma cadeia de caracteres. (ex: `"ABC"`, `"Ricardo"`, `"Agosto/2006"`);
 - ❑ **boolean**: define variáveis do tipo lógico, ou seja, elas só tem dois valores: *true* ou *false*.

Como **String** é uma classe é **OBRIGATÓRIO** começar com letra maiúscula, os demais em minúsculo!

Exemplo 10

```
public class NomePrograma {  
    public static void main(String[] args) {  
        long idade, numero;  
        double altura, peso, salario;  
        String nomePai, rua, bairro, dataNasc;  
        boolean temPai, ehCasado, moraBH;  
    }  
}
```

Exemplo 11

(inicialização de variáveis)

```
public class NomePrograma {  
    public static void main(String[] args) {  
        int idade = 55, numero = 200;  
        float salario = 55400.35f;  
        // Quando é float tem que colocar o f no final  
        double taxaDolar = 5.26;  
        String dataNasc = "20/10/1963";  
        char letra = 'A';  
        // Quando é char tem que colocar aspa simples  
        boolean moraBH = true;  
    }  
}
```

Exemplo 12

(inicialização de variáveis)

```
public class NomePrograma {  
    public static void main(String[] args) {  
        final double PI = 3.1416;  
        // Coloque “final” antes do tipo para que a variável se  
        // transforme numa constante, ou seja, não pode ser  
        // alterada ao longo do programa  
        PI = 3.1416182;  
    }  
}
```

Exemplo 13

(inicialização de variáveis)

```
public class NomePrograma {  
    public static void main(String[] args) {  
        long areaBrasil = 8_516_000;  
        // Os sublinhados ajudam a tornar os números grandes  
        // mais legíveis.  
        // Os sublinhados não afetam o valor de uma variável.  
        areaBrasil = 8.516.000;  
    }  
}
```

Conversão de tipos de variáveis

De	Para	Método
int	String	String str = Integer.toString (numero);
<u>long</u>	String	String str = Long.toString (numero);
float	String	String str = Float.toString (numero);
<u>double</u>	String	String str = Double.toString (numero);
String	int	int numero = Integer.valueOf (str);
String	long	long numero = Long.valueOf (str);
String	float	float numero = Float.valueOf (str);
String	double	double numero = Double.valueOf (str);

Exercício 4

- Baixe o arquivo que está em “Exercícios de Fixação - Conteúdo 3” (planilha Excel) da pasta Conteúdos (Módulo I) da disciplina no Moodle;
- Faça os exercícios listados no arquivo (Partes 1 e 2);
- Poste (envie) o arquivo acima no mesmo local.

Exercício 5

- Baixe e execute os instaladores do Java e do IntelliJ no seu computador conforme as instruções dos slides anterior (FAZER EM CASA, SE AINDA NÃO O FEZ).
- Abra o IntelliJ:
 - Digite nele o programa Java do próximo slide;
 - Execute o programa;
 - Conserte os erros de sintaxe que por ventura apareçam devido a erros de digitação;
 - Entre com os dados para cada campo solicitado;
 - Verifique se o resultado está correto, não sendo, volte ao editor.

Exercício 5

```
import java.util.Scanner;

public class CalcIdade {
    public static void main(String[] args) {
        String nome;
        int anoNasc, anoAtual, idade;
        Scanner teclado = new Scanner(System.in);
        System.out.print("Digite seu nome: ");
        nome = teclado.nextLine();
        System.out.print("Digite o ano em que você nasceu: ");
        anoNasc = teclado.nextInt();
        System.out.print("Digite o ano em que estamos: ");
        anoAtual = teclado.nextInt();
        idade = anoAtual - anoNasc;
        System.out.println(nome+", você tem/terá "+idade+" anos em "+anoAtual);
        teclado.close();
    }
}
```


Fim

Prof. Ricardo Luiz de Freitas

ricardo.freitas@academico.domhelder.edu.br