

CONCEITOS BÁSICOS DE ORIENTAÇÃO A OBJETOS

PROF. ME. HÉLIO
ESPERIDIÃO



CLASSES E OBJETOS PARA PROGRAMAÇÃO ORIENTADA A OBJETOS

Classes são estruturas das linguagens de POO criadas para conter os dados que devem ser representados e as operações que devem ser efetuadas com esses dados para determinado modelo .



Repare como essa definição associa classe e modelo, passando pela ideia de representação de dados e operações em ambos.

ESBOÇO

01

Quando esboçamos nossos primeiros modelos, não foi mencionada a existência de padrão de representação naqueles objetos, embora ele estivesse presente.

02

A escrita de uma classe, contudo, requer critério.

03

O que vem a seguir serve como introdução às regras de escrita de classes na linguagem Java.

NOME DA CLASSE

01

início com letra
maiúscula,

02

sem acento e sem
espaço entre palavras.

NOME DE MÉTODOS E ATRIBUTOS

01

início com letra
minúscula;

02

acento
permitido,
porém pouco
recomendado;

03

sem espaço
entre palavras.



COMO CRIAR UMA CLASSE

- O procedimento de criação de uma classe segue as regras determinadas pela linguagem Java, incluindo:
- Uma classe é sempre declarada com a palavra reservada `class`, seguida do nome da classe.
- O corpo da classe é delimitado por `{` e `}` (leia-se “abre chave” e “fecha chave”).
- A cada “abre chave” deve corresponder um “fecha chave”.



INSTÂNCIAS E ATRIBUTOS PARA PROGRAMAÇÃO ORIENTADA A OBJETOS

- Um objeto (ou instância) é como chamamos a materialização da classe, que assim poderá ser usada para representar dados e executar operações. Fazendo uma analogia, uma classe poderia ser considerada a planta de um prédio, que descreve o prédio, mas não corresponde fisicamente a ele.
- Os prédios construídos de acordo com aquela planta seriam as instâncias

VARIÁVEIS EM JAVA

- A linguagem de programação Java é estaticamente tipada (statically-typed), o que significa que todas as variáveis devem primeiro ser declaradas para que depois possam ser usadas. Basta declarar o tipo de dado, seguido do nome do campo

```
public class POOI {  
    public static void main(String[] args) {  
        int gear = 1;  
    }  
}
```


DECLARAR VARIÁVEL

```
package poo;
public class P00 {

    public static void main(String[] args) {
        int x=10;
        String nome="Hélio";
    }
}
```

Tipo do dado

Nome da variável

Valor do dado é opcional



DADOS PRIMITIVOS

- Oito tipos de dados primitivos suportados pelo Java são byte, short, int, long, float, double, boolean e char

EXEMPLO CLASSE COM ATRIBUTOS

```
public class RegistroAcademicoSimples {  
    String nomeDoAluno; //cadeia de caracteres que deverá conter o nome do aluno  
    int númeroDeMatrícula; // representará o RA do aluno.  
    boolean éBolsista; //assume true ou false.  
    short anoDeMatrícula; //representará o ano da matrícula.  
}
```



INICIALIZAÇÃO DE REFERÊNCIAS

NomeDaClasse nome daReferência = new
NomeDaClasse.

MÉTODO MAIN

- O método main deve conter, obrigatoriamente, os modificadores `public static`, nesta ordem, e deve retornar `void` e receber como argumento um array de instâncias `String`
- Um programa em Java inicia-se pelo método `main()`, e sua declaração é `public static void main (String [] args)`

ENTENDENDO MELHOR

01

public: torna o método visível de qualquer outra classe

02

static: ainda será mais bem estudado. Dispensa a criação de uma instância da sua classe para que possamos criá-lo

03

void: indica que main() não retorna nada.

04

String [] args: argumentos passados para classe executável via linha de comando.

ARQUIVO FONTE

01

Em Java, cada classe é colocada em um arquivo fonte.

02

Esses arquivos representam partes de uma aplicação ou toda a aplicação

03

Arquivos fonte são gerados com a extensão .java e devem possuir o mesmo nome da classe que representam.

PRIMEIRA CLASSE JAVA

- A palavra reservada class deve ser escrita em letra minúscula

```
// A classe Vazia não possui campos nem métodos, mas mesmo assim
// pode ser usada para exemplificar as regras sintáticas básicas
// Java.
public class Vazia {
    /* Se houvesse campos ou métodos para a classe Vazia, eles
    | deveriam ser declarados aqui dentro. */
}
```

EXEMPLO DE CAMPOS OU ATRIBUTOS

```
public class RegistroAcademicoSimples {  
    String nomeDoAluno; //cadeia de caracteres que deverá conter o nome do aluno.  
    int númeroDeMatrícula; // representará o RA do aluno.  
    boolean éBolsista; //assume true ou false.  
    short anoDeMatrícula; //representará o ano da matrícula.  
}
```

ESTRUTURA DE UM PROGRAMA EM JAVA

- Como todo programa em Java vem a ser uma classe então toda a vez que formos escrever um programa Java devemos usar o comando **class** que declara uma classe.

```
package poo;

class Exemplo {

    public static void main(String args[]) {
        System.out.println("Alo Mundo!!!!");
    }

}
```

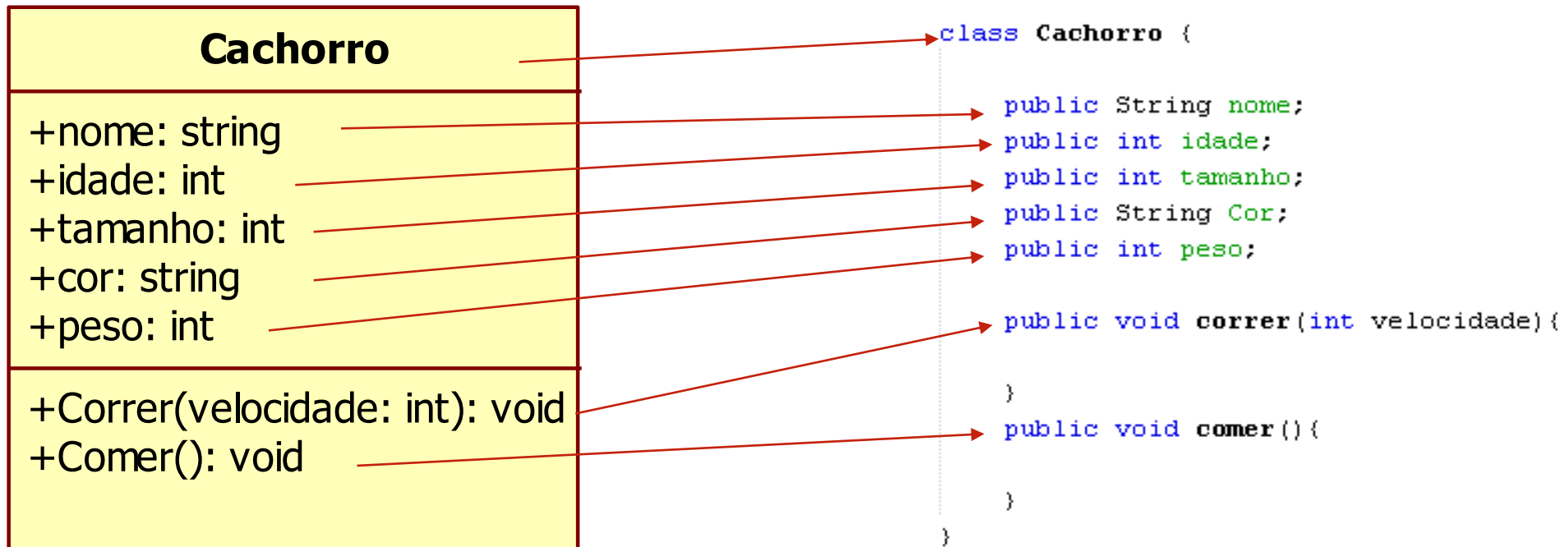
ENTENDENDO MELHOR

- **public** - Onde a palavra chave public significa que esse método será reconhecido por todas as outras classes e métodos. (objetos)
- **static** - Significa que esse método não será nunca instanciado.
- **void** - Significa que o método não irá retornar nenhum valor.

ENTENDENDO MELHOR

- **main** - Vem a ser o próprio método que é onde toda a aplicação escrita irá funcionar.
- **args[]** - Vem a ser um vetor criado pelo Java automaticamente com parâmetros passados pela linha de comandos esse vetor será sempre do tipo String.

CLASSE CACHORRO



INSTANCIAR MINHA CLASSE!

Cachorro
+nome: string +idade: int +tamanho: int +cor: string +peso: int
+Correr(velocidade: int): void +Comer(): void

```
class Cachorro {  
  
    public String nome;  
    public int idade;  
    public int tamanho;  
    public String Cor;  
    public int peso;  
  
    public void correr(int velocidade){  
  
    }  
    public void comer(){  
  
    }  
    public static void main (String args[]){  
  
        Cachorro c1 = new Cachorro();  
        c1.nome="Ted";  
  
        Cachorro c2 = new Cachorro();  
        c2.nome="Bob";  
  
        System.out.println(c1.nome);  
        System.out.println(c2.nome);  
  
    }  
}
```

CLASSE TRIANGULO

```
public class Triangulo {
    float lado1;
    float lado2, lado3;
    boolean éEquilátero(){
        boolean igualdade12, resultado;
        igualdade12 = (lado1 == lado2);
        boolean igualdade23;
        igualdade23 = (lado2 == lado3);
        if (igualdade12 && igualdade23)
            resultado = true;
        else
            resultado = false;
        return resultado;
    }
    float calculaPerímetro() {
        float resultado = lado1 + lado2 + lado3;
        return resultado;
    }
}
```

```
public class DemoDataSimples {
    /** * O método main permite a execução desta classe. Este método contém declarações de algumas instâncias da classe
    * DataSimples e demonstra como seus campos podem ser acessados diretamente, já que são públicos. */
    public static void main (String[] args) {
        // Criamos duas instâncias da classe DataSimples, usando new.
        // As instâncias serão associadas a duas referências, que permitirão
        // o acesso aos campos e métodos das instâncias.
        DataSimples hoje = new DataSimples();
        DataSimples independênciaDoBrasil = new DataSimples();
        byte umDia, umMês; short umAno;
        // e três variáveis para receberem o dia, mês e ano para as datas.
        umDia = 40;
        umMês = 1;
        umAno = 2017;
        hoje.inicializaDataSimples(umDia, umMês, umAno);
        // inicializa os campos da instância.
        hoje.mostraDataSimples(); //imprime 0/0/0. U1 - Fundamentos da orientação a objetos28
        // Inicializando "independênciaDoBrasil como uma data válida
        umDia = 7;
        umMês = 9;
        umAno = 1822;
        independênciaDoBrasil.inicializaDataSimples(umDia, umMês, umAno);
        independênciaDoBrasil.mostraDataSimples(); // imprime 7/9/1822.
        ...
        hoje.mostraDataSimples(); //imprime 0/3/2017

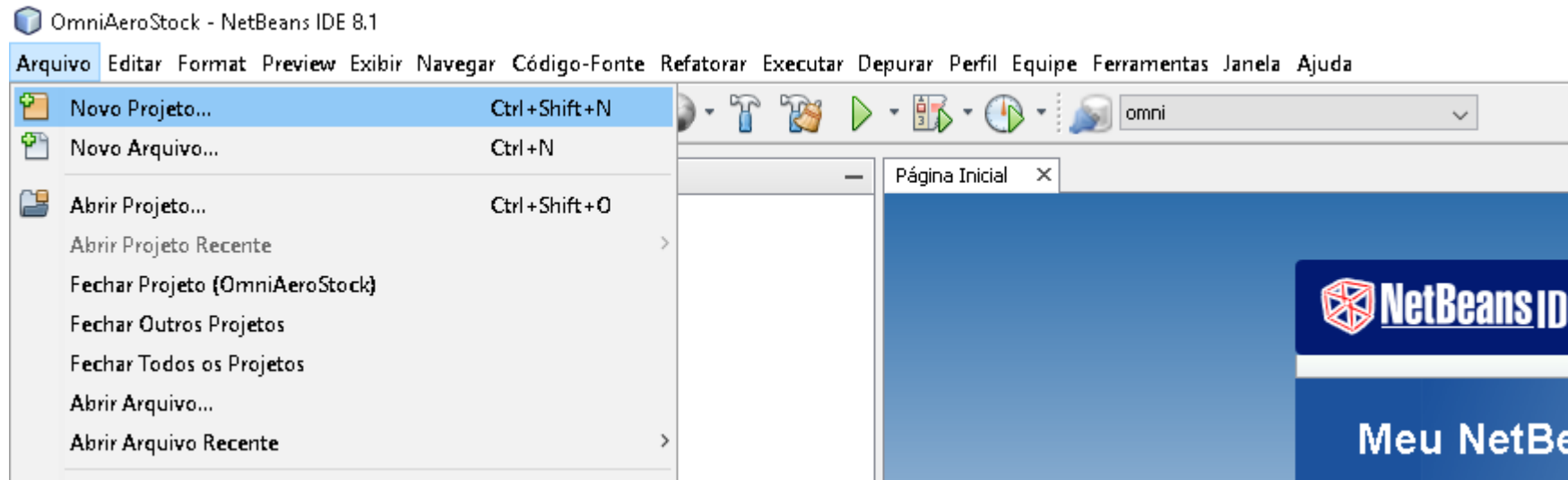
        independênciaDoBrasil.mês = 13;
        independênciaDoBrasil.mostraDataSimples(); // imprime 7/13/1822.    }
        //fim do método main()
    } // fim da classe DemoDataSimples.
```



NetBeans

CRIAR PROJETOS

- O netbeans é baseado no gerenciamento de projetos.



CONFIGURAÇÃO DE PROJETO!

- Escolha a categoria **JAVA** e o projetos: “**Aplicação JA**”

Novo Projeto

Etapas

1. Escolher Projeto
2. ...

Escolher Projeto

Filtro:

Categorias:

- Java
- JavaFX
- Java Web
- Java EE
- HTML5/JavaScript
- Java ME Incorporado
- Java Card
- Maven
- PHP
- Groovy
- C/C++

Projetos:

- Aplicação Java
- Biblioteca de Classe Java
- Projeto Java com Códigos-fonte Existentes
- Projeto de Forma Livre Java

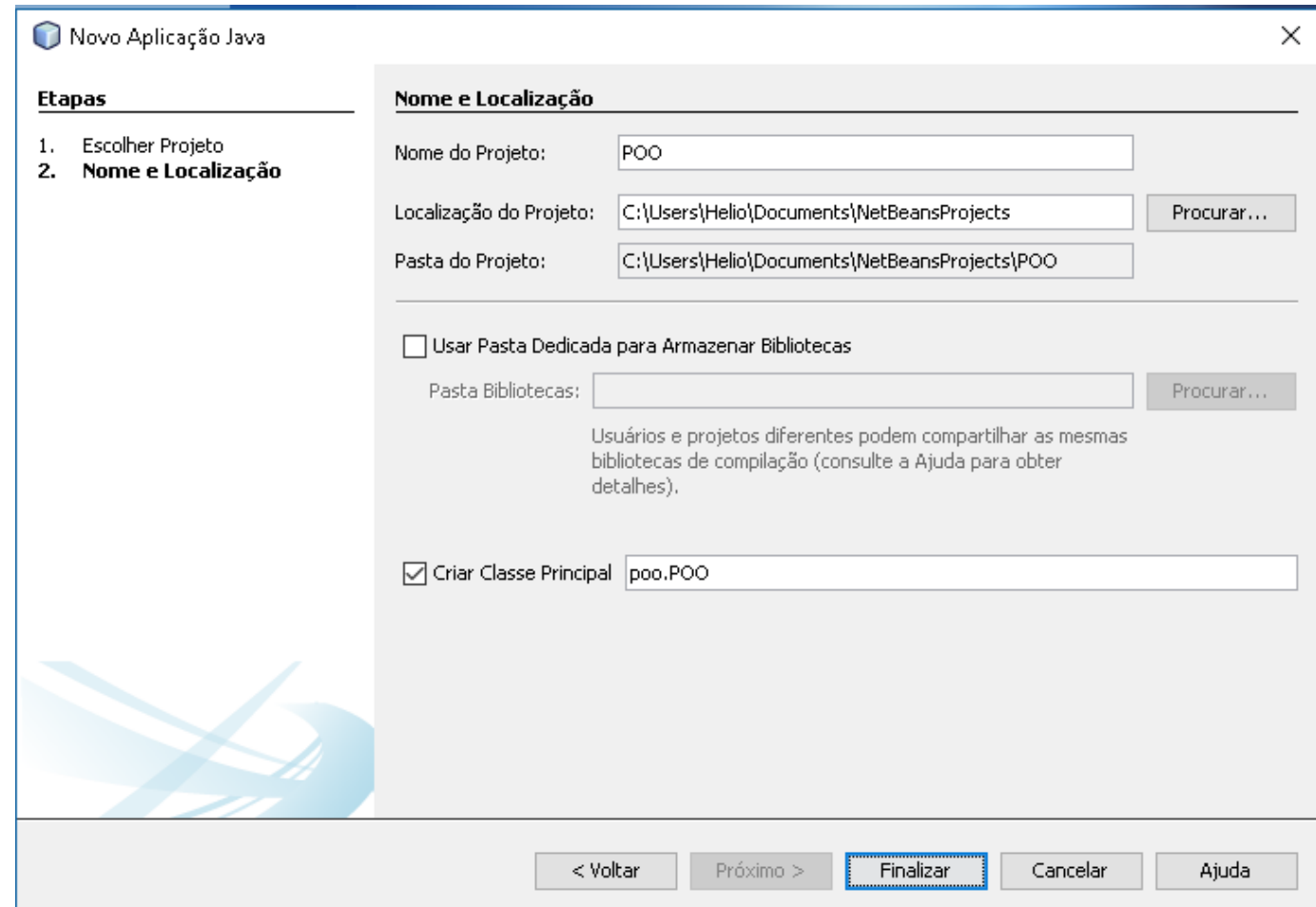
Descrição:

Cria uma nova aplicação Java SE em um projeto padrão do IDE. Você também pode gerar também uma classe principal neste projeto. Os projetos padrão utilizam **um script de construção Ant gerado pelo IDE** para construir, executar, e depurar seu projeto.

< Voltar **Próximo >** Finalizar Cancelar Ajuda

NOME E LOCALIZAÇÃO

- Defina o nome do projeto e a localização física no disco.



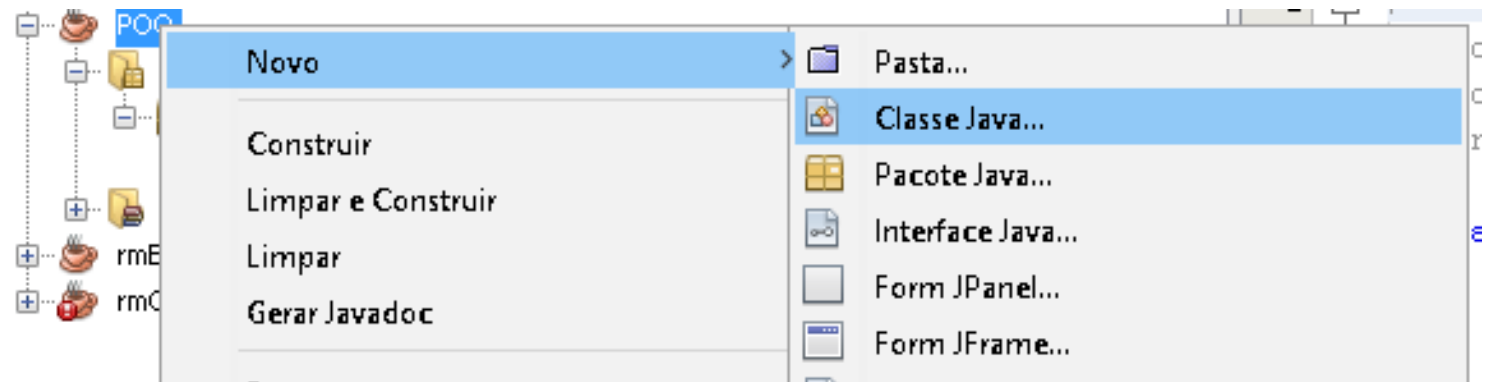
The screenshot shows the 'Novo Aplicação Java' dialog box in NetBeans. The window title is 'Novo Aplicação Java'. On the left, there is a 'Etapas' (Steps) section with two items: '1. Escolher Projeto' and '2. Nome e Localização'. The 'Nome e Localização' section is active and contains the following fields and options:

- Nome do Projeto:** A text box containing 'POO'.
- Localização do Projeto:** A text box containing 'C:\Users\Helio\Documents\NetBeansProjects' with a 'Procurar...' button to its right.
- Pasta do Projeto:** A text box containing 'C:\Users\Helio\Documents\NetBeansProjects\POO'.
- Usar Pasta Dedicada para Armazenar Bibliotecas**
 - Pasta Bibliotecas:** An empty text box with a 'Procurar...' button to its right.
 - Text below: 'Usuários e projetos diferentes podem compartilhar as mesmas bibliotecas de compilação (consulte a Ajuda para obter detalhes).'
- Criar Classe Principal**

At the bottom of the dialog, there are five buttons: '< Voltar', 'Próximo >', 'Finalizar' (highlighted with a blue border), 'Cancelar', and 'Ajuda'.

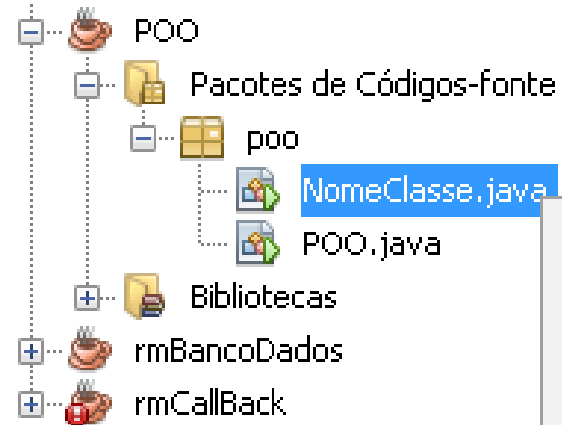
CRIAR AS PRÓPRIAS CLASSES

- Do lado esquerdo da interface do netbeans é mostrada a relação de projetos já criados anteriormente, procure o seu projeto e efetue um clique com o botão direito, sequencialmente escolha a opção **Novo** e **Classe Java**



COMO RODAR!

- Para executar sua classe clique com o botão direito sobre seu nome e escolha a opção executar arquivo.



Abrir	
Recortar	Ctrl+X
Copiar	Ctrl+C
Colar	Ctrl+V
Compilar Arquivo	F9
Executar Arquivo	Shift+F6

PRÁTICA

- Faça o diagrama de classes, implemente em JAVA e instancie as classes abaixo:
- - Aluno
 - Professor
 - Carro
 - Sapato
 - Cabeça
 - Olho

EXERCÍCIOS

- <https://www.draw.io/>