Utilizando os comandos

import java.util.Scanner;  
  
public class App {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);

Quando eu utilizo esse tipo de código, eu posso colocar as informações dentro do teste “no meu main”

Isso é como se o meu usuário estivesse utilizando o meu programa

Alt + j = faz com que eu possa usar vários cursores

Representar veículos

(Navio, submarino, trem, caminhão, carro, bicicleta, avião)

Navio: lotação máxima, combustível

Submarino

Lotação máxima, combustível, profundidade máxima

Trem

Lotação máxima, combustível, numero de rodas

Caminhão

Lotação máxima, combustível, numero de rodas

Carro

Lotação máxima, combustível, numero de rodas

Bicicleta

Aviao

Lotação máxima, combustível, numero de rodas, altitude maxima

Característica dos veículos

(**Nome, velocidade máxima, terreno (terra,água,ar) - PAI**

Profundidade máxima, lotação máxima, **combustível**, altitude máxima, número de rodas

ToString ele coloca automaticamente o que ele vai imprimir para mim

“new Terreno(“aéreo”)) por exemplo eu restrinjo para que meu susuário não altere, uma vez que veículos aéreos o terreno é sempre aéreo.

Para o conceito de poliformismo

Não podemos dizer que será criado um navio e ele é um submarino.. são coisas diferentes.

Então na hora de criar classes, elas precisam fazer sentido.

No poliformismo, eu “escolho” perder propriedades do meu “PAI” e manter o que tem na classe filha.

No exemplo do navio, ele é um veiculo, então vou colocar na hora de imprimir ele

imprimirVeiculo(navio) – vai mostrar as informações que contém na classe navio..

Eu não posso ter

MODIFICADORES DE ACESSO , enums, collections e data e hora – aula tarde – NETO

MODIFICADORES DE ACESSO

O private é só para métodos e atributos – nunca usar para classe

Default é não informar nada – visibilidade no mesmo pacote

Final - eu não posso usar a classe como herança, para classes, métodos ou atributos não forem modificados;

Abstract – usada para classes de estrutura.

Static – ele modifica toda a estrutura

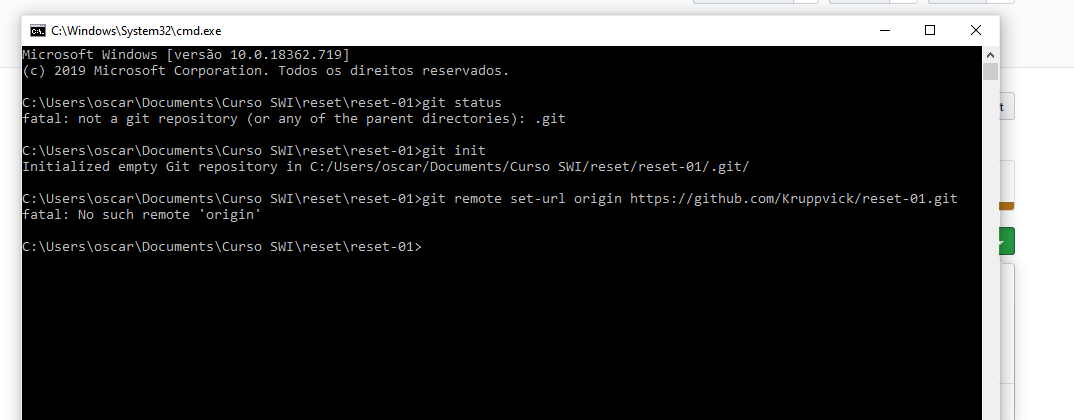
Pacotes

Getter – método de acesso

Uso dele no caso de códigos privados

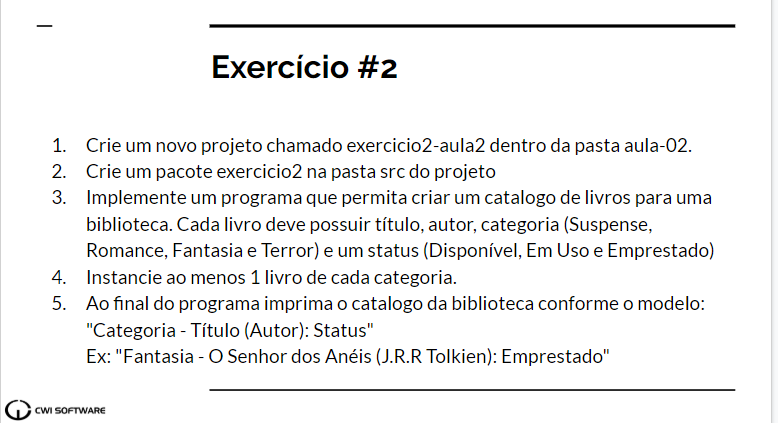
Setter – método de criar valor

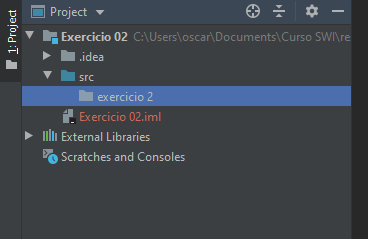
Repositorio



Enum

Enum é um tipo de classe – mas tem caracter exclusivas dela





Main Ctrl + espaço

Alinhar Ctrl + alt + l

Outra forma de fazer a impressão poderia ser a seguinte:

Public void imprimir (){

System.out.println(categoria.getDescricao

Coleções –

List <Aluno> alunos = new ArrayList <> ();

List <Aluno> --- Interface

new ArrayList <> (); --- Implementacao

alunos.add (“Alexandre”)