import java.util.ArrayList;

import java.util.Scanner;

**public class Main**

static Scanner ler = new Scanner(System.in);

public static void main(String args[]){

Classe c = new Classe();

System.out.println("Nome da Turma");

c.setNome(ler.nextLine()); c.setNumero(ler.nextInt());

*//Criar menu de opções*

while(true){

int escolha = menu();

if(escolha==1){

**- CLASSE PADRÃO**: public class NOME{

private TIPO\_VAR Atributo;*//instanciar atributo*

private CLASSE objeto;*//associar com outra classe (relação)*

private ArrayList<Classe> classe;*//declarar lista*

protected TIPO\_VAR Atributo;//*proteção intermediária (associar com package familia)*

private EClasse Atributo;//*instanciar classe Enum*

***//Construtores:***

public Classe() e public Classe(TIPO\_Var atributos,)

this.atributo = atributo;

classe = new ArrayList<>();

***//get/set padrão:***

public TIPO\_VAR getAtributo(){

return

public void setAtributo(TIPO\_VAR atributo){

//String: if(!atributo.equals("");

this.atributo = atributo;

***//get/set ArrayList:***

public ArrayList<Classe> getClasse(){

Return classe;

public void addClasse(Classe NOME\_Abreviado){

classe.add(NOME\_Abreviado);

***//get/set Classe:***

public Classe getClasse(){

return classe;

public void setClasse(Classe classe){

this.classe=classe;

***//get/set Enum:***

public EClasse getAtributo(){

return atributo;

public void setAtributo(EClasse atributo){

this.atributo=atributo;

**//Métodos:**

contas.size(): *representar o tamanho do ArrayList*

for (Conta c : contas): *foreach com objeto criado no void ArrayList*

if (c.getCliente().saldoContas(): *como atribuir atribuitos ao objeto (get)*

*objeto.get(0): posição do objeto Array (Lista)*

**- CLASSE ENUM**: *Lista com variáveis padrões*

public enum ENOME{

VALORES PADRÕES,...

**- CLASSE HERANÇA**: public class NOME extends CLASSE\_PAI

*//Construtor da herança usando SUPER (convoca do pai)*

public NOME(){

super(0,"",0);//*representação tipo\_var (int 0, string “”)*

**- CLASSE INTERFACE**: *Métodos obrigatórios*

public interface INOME{

public TIPO\_VAR NOME\_METODO()}

*//Sintaxe para atribuir interfase (inserir o metodo obrigatorio)*

public class NOME implements INOME

**- CLASSE ABSTRACT:** public abstract class NOME

*//Representa uma super-classe que implementa “modelo obrigatorio” a ser seguido pelas classes filhas*

Abstract TIPO\_VAR getAtibuto();

Classe filha:

public TIPO\_VAR getAtibuto(){

}