**Ejercicio 1**

Crea una clase que defina los siguientes métodos estáticos:

• isInt(String n), que recibe una cadena en el parámetro ny retorna el valor booleano truesi la cadena representa un número de tipo int, o el valor booleano falseen caso contrario.

• isDouble(String n), que recibe una cadena en el parámetro ny retorna el valor booleano truesi la cadena representa un número de tipo double, o el valor booleano falseen caso contrario.

Da una solución basada en captura de excepciones y pon a prueba ambos métodos en el método main.

**package** unidad6;

**public** **class** Excep1 {

**public** **static** **boolean** isInt(String n) {

**int** numero;

**try** {

numero = Integer.*parseInt*(n);

}

**catch** (NumberFormatException ex) {

System.***out***.println("No es un número entero");

**return** **false**;

}

System.***out***.println("Es un número entero");

**return** **true**;

}

**public** **static** **boolean** isDouble(String n) {

**double** numero;

**try** {

numero = Double.*parseDouble*(n);

}

**catch** (NumberFormatException ex) {

System.***out***.println(n + " no es un número con decimales");

**return** **false**;

}

System.***out***.println(n + " es un número con decimales");

**return** **true**;

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

String a = "ab";

String b = "3";

String c = "2.4";

System.***out***.println(*isInt*(a));

System.***out***.println(*isInt*(b));

System.***out***.println(*isInt*(c));

System.***out***.println(*isDouble*(a));

System.***out***.println(*isDouble*(b));

System.***out***.println(*isDouble*(c));

}

}

**Ejercicio 2**

Crea una clase se defina un método estático para resolver una ecuación de segundo grado 𝑎𝑥2 + 𝑏𝑥 + 𝑐 = 0 según las especificaciones siguientes:

* • Declara tres parámetros formales de tipo doublepara recibir el valor de los coeficientes.
* • Retorna las soluciónes −𝑏±ξ𝑏2−4𝑎𝑐 2𝑎
* • Lanza la excepción IllegalArgumentExeptionsi el coeficiente de mayor grado es igual a 0;
* • Lanza la excepción ArithmeticExceptionsi no tiene solución (el discriminante es negativo).

Añade un método mainpara poner a prueba el método anterior pidiendo por teclado los coeficientes a, by cde una ecuación de segundo grado y mostrando en la salida estándar uno de los resultados siguientes:

* • Las soluciones de la ecuación.
* • El mensaje “*no es una ecuación de segundo grado*” si el coeficiente atiene el valor cero.
* • El mensaje “*la ecuación no tiene una solución real*” si la ecuación no tiene solución.

**package** unidad6;

**public** **class** Excep2 {

// 𝑎𝑥2 + 𝑏𝑥 + 𝑐 = 0

**private** **double** n1, n2, n3;

**public** **static** **void** ecuacion(**double** a, **double** b, **double** c) {

**double** resultado, discriminante;

**try** {

*mayor*(a,b);

}

**catch** (Exception e) {

e.printStackTrace();

System.***err***.println("Illegal argument exception");

}

**try** {

discriminante = (Math.*pow*(b, 2)-(4\*a\*c));

*discriminante*(discriminante);

}

**catch** (ArithmeticException e) {

System.***err***.println("No tiene solución");

}

resultado = (-b+(Math.*sqrt*(Math.*pow*(b, 2)-(4\*a\*c))))/(2\*a);

System.***out***.println(resultado);

}

**public** **static** **void** mayor(**double** a, **double** b) **throws** Exception{

**if** ((a>b && a==0) || (b>a && b==0))

**throw** **new** IllegalArgumentException("Coeficiente de mayor grado es igual a 0");

}

**public** **static** **void** discriminante(**double** n) {

**if** (n<0)

**throw** **new** ArithmeticException("El discriminante es negativo");

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**double** a, b, c;

a=3;

b=-17;

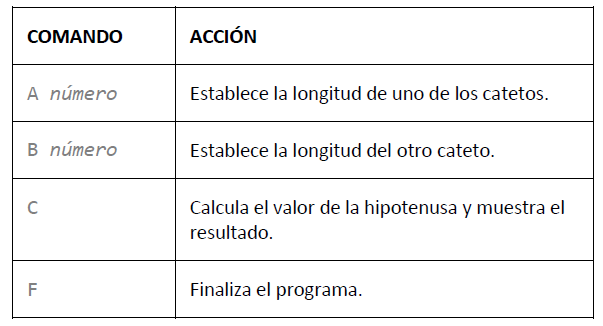
c=20;

*ecuacion*(a,b,c);

}

**Ejercicio 3**

Crea un programa que calcule la hipotenusa de un triángulo rectángulo a partir de la longitud de los catetos. El programa funcionará leyendo los comandos siguientes de la entrada estándar:



No se mostrará en la salida estándar ningún mensaje para indicar la entrada de datos salvo un indicador de línea de comando, por ejemplo, el carácter ‘>’seguido de un espacio en blanco.

Después de cada ejecución sin error del comando C, el programa reiniciará la longitud de los catetos con el valor 0.

El programa se ha de desarrollar usando el manejo de excepciones para comprobar la validez de la entrada de datos y evitar que el programa finalice, además de informar al usuario del error cometido. Se entienden como entradas no válidas las siguientes:

* • Valores no numéricos donde se esperan valores numéricos.
* • Valores numéricos incorrectos.
* • Comandos distintos de A, B, Co F.

Prueba el funcionamiento del programa con la siguiente secuencia de comandos:

A 3B4CA 3B 0CA0B3CA XB XA3CB3CXQ

package unidad6;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Hipotenusa {

public static void main(String[] args) throws IOException {

// TODO Auto-generated method stub

int catetoA = 0;

int catetoB = 0;

double hipotenusa;

String teclado = "";

BufferedReader in = new BufferedReader(new InputStreamReader(System.in));

while (!(teclado.equals("F"))) {

System.out.println("> ");

teclado = in.readLine().toUpperCase();

String letra = teclado.substring(0,1);

//System.out.println(letra);

switch (letra) {

case "A":

try {

catetoA = Integer.parseInt(teclado.substring(1).trim());

if (catetoA<=0)

throw new IllegalArgumentException();

//System.out.println(catetoA);

} catch (NumberFormatException e) {

System.err.println("Por favor, indique un número válido");

} catch (IllegalArgumentException e){

System.err.println("Debe ser un número mayor que 0");

}

break;

case "B":

try {

catetoB = Integer.parseInt(teclado.substring(1).trim());

if (catetoB<=0)

throw new IllegalArgumentException();

//System.out.println(catetoB);

} catch (NumberFormatException e) {

System.err.println("Por favor, indique un número válido");

} catch (IllegalArgumentException e){

System.err.println("Debe ser un número mayor que 0");

}

break;

case "C":

try {

if (catetoA <= 0 || catetoB <= 0)

throw new Exception();

hipotenusa = Math.sqrt(Math.pow(catetoA, 2)+Math.pow(catetoB, 2));

hipotenusa = Math.round(hipotenusa\*100);

hipotenusa /= 100;

System.out.println("Hipotenusa: " + hipotenusa);

catetoA = 0;

catetoB = 0;

} catch (Exception e) {

System.err.println("Por favor, indique el valor de los catetos");

}

break;

default:

System.err.println("Por favor, introduzca un comando válido");

}

}

}

}

OTRA SOLUCION

**package** pruebas;

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.Scanner;

**public** **class** EjerExcep3 {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner input = **new** Scanner(System.***in***);

String teclado, comando;

Integer ca = **null**, cb = **null**;

System.***out***.println(">");

teclado = input.nextLine();

**while** (!teclado.equalsIgnoreCase("f")) {

comando = teclado.substring(0,1);

**switch** (comando.toUpperCase()) {

**case** "A":

**try** {

ca = Integer.*parseInt*(teclado.substring(1,teclado.length()).trim());

**if** (ca<=0)

**throw** **new** IllegalArgumentException();

System.***out***.println("Cateto A: " + ca);

} **catch** (NumberFormatException e) {

System.***err***.println("Error al introducir valor del cateto");

} **catch** (IllegalArgumentException e) {

System.***err***.println("El valor del cateto no puede ser 0 o inferior");

} **finally** {

**break**;

}

**case** "B":

**try** {

cb = Integer.*parseInt*(teclado.substring(1,teclado.length()).trim());

**if** (cb<=0)

**throw** **new** IllegalArgumentException();

System.***out***.println("Cateto B: " + cb);

} **catch** (NumberFormatException e) {

System.***err***.println("Error al introducir valor del cateto");

} **catch** (IllegalArgumentException e) {

System.***err***.println("El valor del cateto no puede ser 0 o inferior");

} **finally** {

**break**;

}

**case** "C":

**try** {

**if** (ca == **null** || cb == **null**) {

**throw** **new** NullPointerException();

}

System.***out***.println("Hipotenusa: " + *calcula*(ca,cb));

ca=**null**;

cb=**null**;

} **catch** (NullPointerException e) {

System.***err***.println("ERROR. Establezca el valor de los catetos. A: " + ca + ", B: " + cb);

} **catch** (Exception e) {

System.***err***.println(e);

} **finally** {

**break**;

}

**default**:

System.***err***.println("Introduzca un comando válido o \"F\" para salir");

}

System.***out***.println(">");

teclado = input.nextLine();

}

input.close();

}

**public** **static** **double** calcula(**int** ca, **int** cb) {

**double** h;

h = Math.*sqrt*((Math.*pow*(ca, 2) + Math.*pow*(cb, 2)));

h = Math.*round*(h\*100)/100;

**return** h;

}

}