



PRACTICA 12C

Facultad de ciencias de la computación
Ing. En ciencias de la computación



IRVYN XICALE CABRERA
201963582

Introducción

Esta practica nos ayuda a comprender el lenguaje java además de dar inicio a los principios de la programación orientada a objetos.

Preguntas

1.-Definir los términos de excepciones: try, catch, finally, throw, throws.

Try : intentar ejecutar el código que este dentro del try y en caso de que obtenga un error entonces actuara el catch

Catch : atrapa el error obtenido generalmente en una variable llamada e (el nombre de la variable puede cambiar)

Finally : son líneas de código que se necesitan tanto si se produce la excepción o no, el finally no es obligatorio

Throws: un método marcado con throws debe tratar o atrapar la excepción o declarar otro throws indicando que propagará la excepción mediante la sentencia throw para lanzar la excepción de manera explícita

2.-¿Cuál es la sentencia que se utiliza para re-lanzar una excepción?

Throw new Exception

3. ¿Cuál es la salida del siguiente código de excepciones?

```
import java.io.*;

public class exceptionHandle{

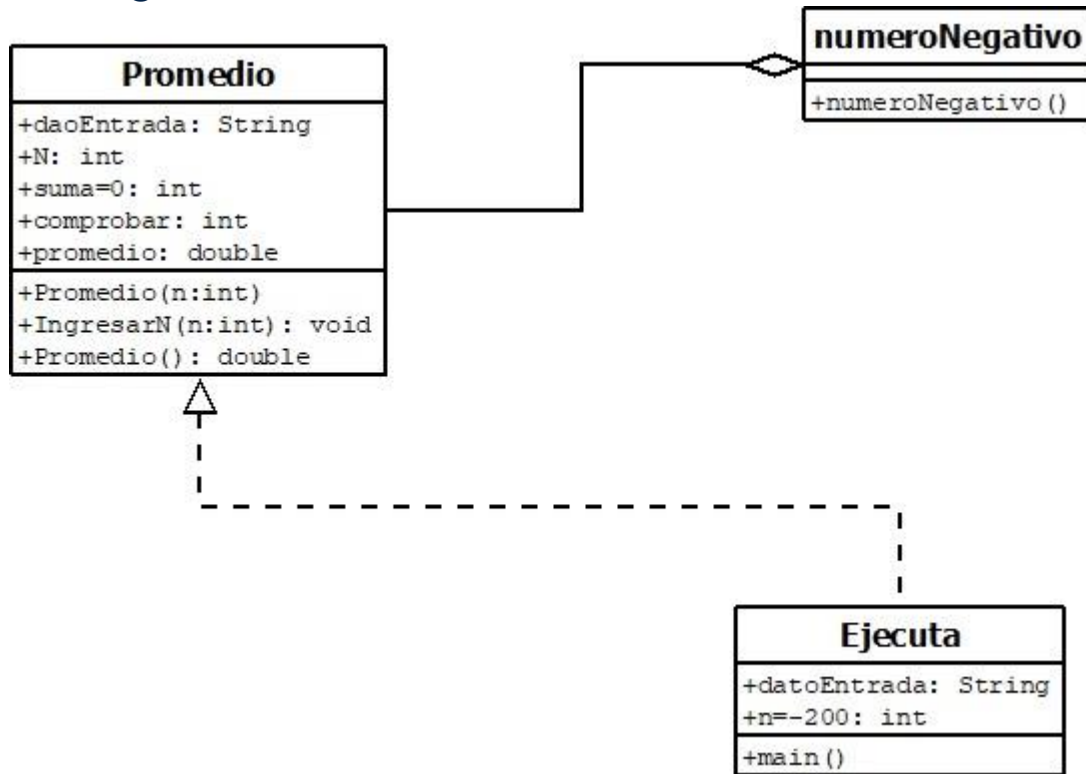
    public static void main(String[] args) throws Exception{
```

```
try{  
    int a,b;  
    BufferedReader in = new BufferedReader(new InputStreamReader(System.in));  
    a = Integer.parseInt(in.readLine());  
    b = Integer.parseInt(in.readLine());  
} catch(NumberFormatException ex){  
    System.out.println(ex.getMessage() + " no es un valor numérico"); System.exit(0); }  
}  
}
```

Imprime en caso de ingresar una cadena = "cadena" no es un valor numérico

En caso de ser un numero lo guarda en las variables

5. Diagramas ULM



Código

NumeroNegativo.java

```
package ejer5;
```

```
public class numeroNegativo extends Exception{  
    public numeroNegativo(){  
        super("el numero es negativo");  
    }  
}
```

PROMEDIO.JAVA

```
package ejer5;  
import java.io.*;  
  
public class Promedio {  
    InputStreamReader entrada = new InputStreamReader (System.in);  
    BufferedReader flujoEntrada = new BufferedReader(entrada);  
    String datoEntrada;  
  
    int[] N;  
    int suma=0,comprobar;  
    double promedio;  
  
    public Promedio (int n){
```

```
    N = new int[n];  
}  
  
public void ingresarN(int n ){  
    for(int i=0;i<n;i++){  
        System.out.println("ingresa el numero " + (i+1));  
        do{  
            try{  
                datoEntrada = flujoEntrada.readLine();  
                comprobar= Integer.parseInt(datoEntrada);  
            }catch(Exception e){  
                System.out.println("Error " + e.getMessage());  
            }  
            try{  
                if(comprobar<0){  
                    throw new numeroNegativo();  
                }  
            } catch (numeroNegativo e){  
                System.out.println("Error " + e.getMessage());  
            }  
        }  
    }  
}
```

```

        }while(comprobar<0);

        N[i]= Integer.parseInt(datoEntrada);

        suma=suma+N[i];

    }

}

public double promedio(){

    promedio=suma/N.length;

    return promedio;

}

}

```

EJECUTA.JAVA

```

package ejer5;
import java.io.*;

public class Ejecuta {

    public static void main (String[] args){

        InputStreamReader entrada = new InputStreamReader (System.in);

        BufferedReader flujoEntrada = new BufferedReader(entrada);
    }
}

```

```
String datoEntrada;
int n=-200;

System.out.println("ingresa N:");

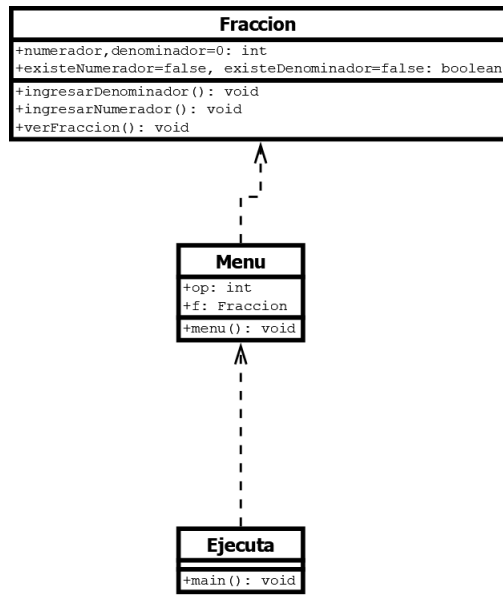
do{
    try {
        datoEntrada = flujoEntrada.readLine();
        n = Integer.parseInt(datoEntrada);
    }catch(Exception e){
        System.out.println("Error " + e.getMessage());
    }
    try{
        if(n<0){
            throw new numeroNegativo();
        }
    }catch (numeroNegativo e){
        System.out.println("Error " + e.getMessage());
    }
}while(n<0);
```



```
Promedio promedio = new Promedio(n);  
promedio.ingresarN(n);  
System.out.println("el promedio es " + promedio.promedio());  
  
}  
  
}
```

```
ingresa N:  
-200  
Error el numero es negativo  
3  
ingresa el numero 1  
-500  
Error el numero es negativo  
10  
ingresa el numero 2  
8  
ingresa el numero 3  
9  
el promedio es 9.0  
PS C:\Users\irvyn\OneDrive\Pictures\pagina web\excepciones> |
```

6. Diagramas ULM



Código

FRACCION.JAVA

```
import java.io.*;
public class Fraccion {
    int numerador,denominador=0;
    boolean existeNumerador=false, existeDenominador=false;

    // lectura
    InputStreamReader entrada=new InputStreamReader (System.in);
    BufferedReader flujoEntrada=new BufferedReader(entrada);
    String datoEntrada;
```

```

public void ingresarDenominador(){
    System.out.println("Ingresa el denominador");
    do{
        try{
            datoEntrada = flujoEntrada.readLine();
            denominador=Integer.parseInt(datoEntrada);
            existeDenominador=true;
        }catch(Exception e){
            System.out.println("Error " + e.getMessage());
        }
        try{
            if (denominador == 0){
                throw new DenominadorCeroExcepcion();
            }
        } catch (DenominadorCeroExcepcion e){
            System.out.println("Error " + e.getMessage());
            e.printStackTrace();
        } finally{
            System.out.println("Intentalo otra vez");
        }
    }while(denominador==0);
}

public void ingresarNumerador(){
    try{
        System.out.println("Ingresar el Numerador");
        datoEntrada= flujoEntrada.readLine();
        numerador= Integer.parseInt(datoEntrada);
        existeNumerador=true;
    }catch(Exception e){
        System.out.println("ERROR " + e.getMessage());
    }
}

```

```

    }
}

public void verFraccion(){
    if(existeNumerador==true && existeDenominador == true){
        System.out.println(numerador + " / " + denominador);
    }
}
}

```

MENU.JAVA

```

import java.io.*;
public class Menu {
    int op;
    Fraccion f = new Fraccion();

    InputStreamReader entrada = new InputStreamReader(System.in);
    BufferedReader flujoEntrada = new BufferedReader(entrada);
    String datoEntrada;

    public void menu(){
        try {
            do{
                System.out.println("\tMenu\n0.-salir\n1.-Ingresar numerador\n2.-Ingresar denominador\n3.-
ver fraccion");
                datoEntrada = flujoEntrada.readLine();
                op = Integer.parseInt(datoEntrada);
                switch(op){
                    case 0:

```

```

        System.out.println("adios");
        break;
        case 1:
            f.ingresarNumerador();
            break;
        case 2:
            f.ingresarDenominador();
            break;
        case 3:
            f.verFraccion();
            break;
        default: System.out.println("opcion invalida");
    }

    }while(op!=0);
}catch (IOException e) {
    System.out.println("Error " + e.getMessage());
}
}
}

```

EJECUTA.JAVA

```

public class Ejecuta {

    public static void main(String[] args){
        Menu m = new Menu();
    }
}

```

```
        m.menu();  
    }  
}
```

```
Menu  
0.-salir  
1.-Ingresar numerador  
2.-Ingresar denominador  
3.-ver fraccion  
1  
Ingresar el Numerador  
0  
Menu  
0.-salir  
1.-Ingresar numerador  
2.-Ingresar denominador  
3.-ver fraccion  
2  
Ingresa el denominador  
0  
Error el denominador es 0  
DenominadorCeroExcepcion: el denominador es 0  
    at Fraccion.ingresarDenominador(Fraccion.java:24)  
    at Menu.menu(Menu.java:24)  
    at Ejecuta.main(Ejecuta.java:6)  
Intentalo otra vez  
0  
Error el denominador es 0  
DenominadorCeroExcepcion: el denominador es 0  
    at Fraccion.ingresarDenominador(Fraccion.java:24)  
    at Menu.menu(Menu.java:24)  
    at Ejecuta.main(Ejecuta.java:6)  
Intentalo otra vez  
5  
Intentalo otra vez  
Menu  
0.-salir  
1.-Ingresar numerador  
2.-Ingresar denominador  
3.-ver fraccion  
3
```

8. ¿Cuál es la diferencia entre try y catch?

Try intenta correr el código y el catch captura la excepción

9. ¿Qué pasa si en un try no existe un catch?

No se puede compilar el código

10. ¿Cuál sería el resultado de un bloque try catch sin bloque finally?

Todo ejecutaría normal porque el finally es opcional

11. ¿Explique las diferencias entre el uso de excepciones para controlar errores y el uso de condicionales "if" ¿

Con el "if" tenemos la posibilidad de validar el dato, aunque podemos combinar las excepciones con el "if"

12. Dado el siguiente código Java

```
int LimiteInferior; // más sentencias...
```

```
try {
```

```
    System.out.println(" Ingresa el bloque try. ");
```

```
    if (LimiteInferior < 100)
```



```
throw new Exception("Violación del límite inferior.");  
System.out.println("Salir del bloque try.");  
} catch (Exception e) {  
System.out.println("Excepción: " + e.getMessage());  
}  
System.out.println("Después del bloque catch");
```

¿Cuál es la salida Sí?

- El límite inferior es 50

Se crea la excepción “Violación del límite inferior”

- El valor del límite inferior es 150

Salir del bloque try

¿Cuál es la salida del siguiente código?

```
public class Pregunta {  
public static void main(String[] args) {
```

```
try {  
return;  
} finally {  
System.out.println( "finally" );  
}  
}  
}
```

- a) finally
- b) Error de compilación
- c) No regresa nada
- d) Se lanza una excepción en tiempo de ejecución.

¿Cuál es la salida del siguiente código?

```
public class Pregunta {  
public static void main(String [] args) {  
try {  
maloMethod();  
System.out.print("A");  

```

```
} catch (RuntimeException ex) /* Line 10 */ {  
    System.out.print("B");  
} catch (Exception ex1) {  
    System.out.print("C");  
} finally {  
    System.out.print("D");  
} System.out.print("E");  
} public static void maloMethod() {  
    throw new RuntimeException();  
}  
}
```

A.BD

B.BCD

C.BDE

D.BCDE

Conclusión

Esta practica nos ayudo a comprender la programación orientada a objetos

Hoja de firmas

RÚBRICA HOJA DE EVALUACIÓN DE PRÁCTICAS

MATRICULA: 201963582 FECHA: 26/04/2021

NOMBRE: Xicale Cabrera Irvyn No. PRACTICA: 12c INDIVIDUAL (☒) COLABORATIVA (☐)

| CRITERIOS | Deficiente | Suficiente | Bueno | Excelente | Calificación Obtenida | |
|--|--|---|--|---|-----------------------|--------|
| | 1-5.9 | 6-7.9 | 8-9 | 9.1-10 | % | Puntos |
| CONOCIMIENTO TEÓRICO 20% | Conocimiento deficiente de los fundamentos teóricos de POO y no puede aplicarlos en el diseño de clases. | Conocimiento confuso de los fundamentos teóricos de POO y el diseño de las clases y relaciones es incompleto. | Conocimiento claro de los fundamentos teóricos POO pero requiere mejorar el modelado de las clases, sus métodos y sus relaciones entre clases. | Dominio del Conocimiento de los fundamentos teóricos POO y puede aplicarlos de forma completa en el modelado de las clases, métodos y todas las relaciones entre clase. | | |
| EJECUCIÓN DE LA PRÁCTICA 30% | No puede realizar la práctica ya que desconoce el entorno de trabajo y desarrollo de la práctica en lenguaje UML y Java. | Realiza la práctica de forma incompleta ya que desconoce el entorno de trabajo del lenguaje UML y Java | Realiza la práctica pero requiere mejorar en el manejo del entorno de trabajo del lenguaje de programación(sintaxis y semántica) | Realiza la práctica de forma correcta y completa, demuestra dominio del entorno de trabajo del lenguaje de programación (sintaxis y semántica). | | |

| | | | | | | |
|---|--|---|--|--|--|--|
| SOLUCIÓN DE LA PRÁCTICA 40% | No puede generar las soluciones o programas a los problemas planteados ya que no posee el dominio teórico y práctico del modelado y el lenguaje de programación. | Propone soluciones confusas o programas incompletos a los problemas planteados, ya que carece del dominio del modelado y lenguaje de programación | Genera soluciones con poca profundidad y los programas no están orientados de acuerdo a los problemas solicitados, por lo cual no tiene un dominio profundo de la temática y del Lenguaje. | Genera soluciones con profundidad y los programas son correctos de acuerdo a los problemas planteados, por lo cual demuestra un dominio de la temática y del Lenguaje de Programación. | | |
| ACTITUD DE APRENDER Y COLABORAR EN EQUIPO DE TRABAJO 10% | No posee una actitud proactiva para un aprendizaje autónomo y no le gusta participar y trabajar en equipo. | Posee una actitud propositiva para un aprendizaje autónomo, participa pero no le gusta trabajar en equipo. | Posee una actitud propositiva logrando un aprendizaje autónomo, colaborativo, le gusta trabajar en equipo pero requiere mejorar su participación y portaciones de forma profunda. | Posee una actitud proactiva logrando un aprendizaje autónomo, participa con propuestas concretas y profundas, le gusta trabajar en equipo y asume su responsabilidad dentro para lograr éxito del equipo | | |
| Total | | | | | | |

Firma del Alumn@

Vo. Bo. Docente

