PROGRAMACION ORIENTADA A OBJETOS

REPORTE EJERCICIO 18

Jose Enrique Gudiño Gomez

1169740

DESCRIPCIOSN DEL PROBLEMA.

Realizar las siguientes actividades:

1. Codifica las clases que se presentan en el diagrama UML.

2. Utilice la imagen como guía para hacer elaborar una aplicación para registrar un televisor

inteligente, no se va a crear la parte gráfica, solo se toma en cuenta los datos que se

deben registrar.

3. En la aplicación todos los datos serán String.

Restricciones:

1. Todos los campos son obligatorios por lo que será un error dejar en blanco alguno de ellos.

2. Para el correo electrónico deberá seguir el formato usuario@servidor.dominio, el servidor

puede tener tres niveles como máximo subcarpeta.carpeta.servidor y el dominio serán 3

letras, no cumplir con alguna de estas condiciones o tener espacios en blanco

representan errores, en su construcción.

3. Tanto nombre como apellido deben empezar con letra.

4. El número de teléfono se deberá escribir en el formato (999)999-9999.

5. El código postal será de 5 números, el cero no es válido como primer número.

FICHA.

package modelo;

@author jabad

public class Ficha {

private String nombre;

private int edad;

private String direccion;

private String telefono;

private String eMail;

public Ficha(String nombre, int edad, String direccion, String telefono, String eMail) {

this.nombre = nombre;

this.edad = edad;

this.direccion = direccion;

this.telefono = telefono;

this.eMail = eMail;

}

public String getNombre() {

return nombre;

}

public void setNombre(String nombre) {

this.nombre = nombre;

}

public int getEdad() {

return edad;

}

public void setEdad(int edad) {

this.edad = edad;

}

public String getDireccion() {

return direccion;

}

public void setDireccion(String direccion) {

this.direccion = direccion;

}

public String getTelefono() {

return telefono;

}

public void setTelefono(String telefono) {

this.telefono = telefono;

}

public String geteMail() {

return eMail;

}

public void seteMail(String eMail) {

this.eMail = eMail;

}

@Override

public String toString() {

if (direccion.equals("")) {

return "Nombre : " + nombre

+ "\nEdad : " + edad

+ "\nTelefono : " + telefono

+ "\nEmail : " + eMail;

} else {

return "Nombre : " + nombre

+ "\nEdad : " + edad

+ "\nDireccion: " + direccion

+ "\nTelefono : " + telefono

+ "\nEmail : " + eMail;

}

}

}

MOTODOS.

package controlador;

import java.util.ArrayList;

import modelo.Ficha;

import util.MiExcepcion;

@author jabad

public class Metodos {

ArrayList<Ficha> lista = new ArrayList<>();

public void add(String nombre, int edad, String direccion, String telefono, String eMail) {

lista.add(new Ficha(nombre, edad, direccion, telefono, eMail));

}

public String valida\_nombre(String nombre) throws MiExcepcion {

if (nombre.equals("")) {

throw new MiExcepcion("Campo obligatorio");

}

return nombre;

}

public int valida\_edad(String edad) throws MiExcepcion {

if (edad.equals("")) {

throw new MiExcepcion("Campo obligatorio");

}

try {

int edadNumerica = Integer.parseInt(edad);

if (edadNumerica <= 0) {

throw new MiExcepcion("Debe ser entero positivo");

}

return edadNumerica;

} catch (NumberFormatException ex) {

throw new MiExcepcion("Dato entero");

}

}

public String valida\_telefono(String telefono) throws MiExcepcion {

String numeros = "0123456789";

if (telefono.equals("")) {

throw new MiExcepcion("Campo obligatorio");

}

int len = telefono.length();

if (len != 14) {

throw new MiExcepcion("Formato: (000)000‐00‐00");

}

boolean wrongFormat = false;

for (int x = 0; x < 14; x++) {

switch (x) {

case 0:

wrongFormat = (telefono.charAt(x) != '(');

break;

case 4:

wrongFormat = (telefono.charAt(x) != ')');

break;

case 8:

case 11:

wrongFormat = (telefono.charAt(x) != '‐');

break;

default:

wrongFormat = !(numeros.contains(Character.toString(telefono.charAt(x))));

}

if (wrongFormat) {

throw new MiExcepcion("Formato: (000)000‐00‐00");

}

}

return telefono;

}

public String valida\_eMail(String eMail) throws MiExcepcion {

if (eMail.equals("")) {

throw new MiExcepcion("Campo obligatorio");

}

if (!eMail.contains("@")) {

throw new MiExcepcion("Formato: usuario@dominio.extension");

}

String[] parts = eMail.split("@");

if (parts.length != 2) {

throw new MiExcepcion("Formato: usuario@dominio.extension");

}

String[] server = parts[1].split("\\.");

if ((server.length < 2 || server.length > 3) || (server[1].length() != 3)

|| (server.length == 3 && server[2].length() != 2)) {

throw new MiExcepcion("Formato: usuario@dominio.extension");

}

return eMail;

}

public void show(){

for(Ficha f: lista){

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

System.out.println(f);

}

}

}

REGISTRÓ.

package vista;

import controlador.Metodos;

import java.util.InputMismatchException;

import java.util.Scanner;

import util.MiExcepcion;

@author jabad

public class Registro {

static Metodos control;

static Scanner inN;

static Scanner inS;

public static void main(String[] args) {

control = new Metodos();

inN = new Scanner(System.in);

inS = new Scanner(System.in);

do {

switch (menu()) {

case 1:

alta();

break;

case 2:

mostrar();

break;

case 3:

System.exit(0);

break;

default:

System.out.println("Opcion no valida");

break;

}

} while (true);

}

static int menu() {

System.out.println("\*\*\* Agenda \*\*\*");

System.out.println("1. Guardar");

System.out.println("2. Mostrar");

System.out.println("3. Terminar");

System.out.print(">> ");

do {

try {

int op = inN.nextInt();

return op;

} catch (InputMismatchException ex) {

System.out.println("Introduce un numero entero");

inN = new Scanner(System.in);

}

} while (true);

}

static void alta() {

String nombre;

int edad;

String direccion;

String telefono;

String eMail;

System.out.println("captura de datos (\*obligatorio)");

do {

System.out.print("\*Nombre: ");

nombre = inS.nextLine();

try {

nombre = control.valida\_nombre(nombre);

break;

} catch (MiExcepcion miEx) {

System.out.println(miEx.getMessage());

}

} while (true);

do {

System.out.print("\*Edad: ");

String ed = inS.nextLine();

try {

edad = control.valida\_edad(ed);

break;

} catch (MiExcepcion miEx) {

System.out.println(miEx.getMessage());

}

} while (true);

System.out.print("Direccion: ");

direccion = inS.nextLine();

do {

System.out.print("\*Telefono: ");

telefono = inS.nextLine();

try {

telefono = control.valida\_telefono(telefono);

break;

} catch (MiExcepcion miEx) {

System.out.println(miEx.getMessage());

}

} while (true);

do {

System.out.print("\*Email: ");

eMail = inS.nextLine();

try {

eMail = control.valida\_eMail(eMail);

break;

} catch (MiExcepcion miEx) {

System.out.println(miEx.getMessage());

}

} while (true);

control.add(nombre, edad, direccion, telefono, eMail);

}

static void mostrar() {

System.out.println();

control.show();

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

System.out.println();

}

}

MI EXCEPCION.

package util;

@author jabad

public class MiExcepcion extends Exception{

public MiExcepcion() {

}

public MiExcepcion(String mensaje){

super(mensaje);

}

}