public abstract class **Filtro** {

public abstract boolean eval(Noticia n);

}

public class **FiltroTitulo** extends Filtro {

private String titulo;

public FiltroTitulo(String tit) {

titulo = tit;

}

public boolean eval(Noticia n) {

return n.getTitulo().compareTo(titulo) == 0;

}

}

public class **FiltroCategoria** extends Filtro {

private String categoria;

public FiltroCategoria(String cat) {

categoria = cat;

}

public boolean eval(Noticia n) {

return n.getCategoria().compareTo(categoria) == 0;

}

}

public class **FiltroCantidad** extends Filtro {

private int cantidad;

public FiltroCantidad(int cantidad) {

this.cantidad = cantidad;

}

public boolean eval(Noticia n) {

return n.cantPals() < cantidad;

}

}

import java.util.Vector;

import java.util.Enumeration;

public class **FiltroContiene** extends Filtro {

private Vector palabras;

public FiltroContiene(Vector pals) {

palabras = pals;

}

public boolean eval(Noticia n) {

for (Enumeration e = palabras.elements(); e.hasMoreElements();)

if (!n.contiene((String) e.nextElement()))

return false;

return true;

}

}

public class **FiltroNot** extends Filtro {

private Filtro f;

public FiltroNot(Filtro f) {

this.f = f;

}

public boolean eval(Noticia n) {

return !f.eval(n);

}

}

public abstract class **FiltroCombinado** extends Filtro {

protected Filtro f1, f2;

public FiltroCombinado(Filtro f1, Filtro f2) {

this.f1 = f1;

this.f2 = f2;

}

}

public class **FiltroOr** extends FiltroCombinado {

public FiltroOr(Filtro f1, Filtro f2) {

super(f1, f2);

}

public boolean eval(Noticia n) {

return f1.eval(n) || f2.eval(n);

}

}

public class **FiltroAnd** extends FiltroCombinado {

public FiltroAnd(Filtro f1, Filtro f2) {

super(f1, f2);

}

public boolean eval(Noticia n) {

return f1.eval(n) && f2.eval(n);

}

}

public abstract class **ComponenteCuerpo** {

public abstract boolean contiene(String s);

public abstract int cantPalabras();

}

import java.util.Vector;

public class **Texto** extends ComponenteCuerpo {

Vector pals;

public Texto(Vector p) {

pals = p;

}

public Vector getPals() {

return pals;

}

public int cantPalabras() {

return pals.size();

}

public boolean contiene(String s) {

return pals.contains(s);

}

}

public class **Media** extends ComponenteCuerpo {

String nombreArchivo;

public Media(String s) {

nombreArchivo = s;

}

public int cantPalabras() {

return 0;

}

public boolean contiene(String s) {

return false;

}

}

import java.util.Vector;

import java.util.Enumeration;

public class **Combinacion** extends ComponenteCuerpo {

Vector componentes;

public Combinacion(Vector comp) {

componentes = comp;

}

public Vector getComponentes() {

return componentes;

}

public int cantPalabras() {

int total = 0;

for (Enumeration e = componentes.elements(); e.hasMoreElements();)

total+=((ComponenteCuerpo) e.nextElement()).cantPalabras();

return total;

}

public boolean contiene(String s) {

for (Enumeration e = componentes.elements(); e.hasMoreElements();)

if(((ComponenteCuerpo) e.nextElement()).contiene(s)) return true;

return false;

}

}

import java.util.Vector;

import java.util.Enumeration;

public class **Usuario** {

private String nombre;

private Vector preferencias;

private Aplicacion aplicacion;

public Usuario(String nom, Vector pref, Aplicacion apl) {

nombre = nom;

preferencias = pref;

aplicacion = apl;

}

public Vector getPreferencias() {

return preferencias;

}

public String getNombre() {

return nombre;

}

public Aplicacion getAplicacion() {

return aplicacion;

}

public void addPreferencia(Filtro f) {

preferencias.add(f);

}

public boolean interesa(Noticia n) {

for (Enumeration e = preferencias.elements(); e.hasMoreElements();)

if (n.satisface((Filtro) e.nextElement()))

return true;

return false;

}

}

public interface **Receptor** {

public void recibir(Noticia n);

}

import java.util.Vector;

public class **Aplicacion** implements Receptor {

private Vector recibidas;

public Aplicacion() {

recibidas = new Vector();

}

public Vector getRecibidas() {

return recibidas;

}

public void recibir(Noticia n) {

recibidas.add(n);

}

}

import java.util.Vector;

import java.util.Enumeration;

public class **Servidor** {

private String ip;

private Vector noticias;

private Vector usuarios;

public Servidor(String ip, Vector not, Vector us) {

this.ip = ip;

noticias = not;

usuarios = us;

}

public String getIp() { return ip;}

public void addUsuario(Usuario u) {

usuarios.add(u);

}

public void addNoticia(Noticia n) {

noticias.add(n);

for (Enumeration e = usuarios.elements(); e.hasMoreElements();) {

Usuario u = (Usuario) e.nextElement();

if (u.interesa(n))

u.getAplicacion().recibir(n);

}

}

public Vector buscar(Filtro f) {

Vector res = new Vector();

for (Enumeration e = noticias.elements(); e.hasMoreElements();) {

Noticia n = (Noticia) e.nextElement();

if (n.satisface(f))

res.add(n);

}

return res;

}

}

public class **Noticia** {

private ComponenteCuerpo cuerpo;

private String titulo;

private String categoria;

private String agencia;

public Noticia(String tit, String cat, String ag, ComponenteCuerpo c) {

cuerpo = c;

titulo = tit;

categoria = cat;

agencia = ag;

}

public boolean satisface(Filtro f) {

return f.eval(this);

}

public boolean contiene (String s) {

return cuerpo.contiene(s);

}

public int cantPals() {

return cuerpo.cantPalabras();

}

public String getTitulo() {

return titulo;

}

public String getCategoria() {

return categoria;

}

public String getAgencia() {

return agencia;

}

public ComponenteCuerpo getCuerpo() {

return cuerpo;

}

}