**Exercici A1 – Punt**

Crea un programa amb una classe anomenada Punt que representarà un punt de dues dimensions

en un pla. Sols contindrá dos atributs sencers anomenats x e y (coordenades).

En el main de la classe principal instància 3 objectes Punt amb les coordenades (5,0), (10,10) i (-3,

7). Mostra per pantalla les seues coordenades (utilitza un println per a cada punt). Modifica totes

les coordenades (prova diferents operadors com = + - += \*=...) i torna a imprimir-les per pantalla.

public class Punt\_main {

public static void main(String[] args) {

Punt punto1 = new Punt(5,0);

Punt punto2 = new Punt(10,10);

Punt punto3 = new Punt(-3,7);

System.***out***.println("El punto 1: " + punto1.getX()+ " , " + punto1.getY());

System.***out***.println("El punto 2: " + punto2.getX()+ " , " + punto2.getY());

System.***out***.println("El punto 3: " + punto3.getX()+ " , " + punto3.getY());

punto1.x +=3;

punto1.y -=3;

punto2.x+=3;

punto2.y -=3;

punto3.x +=3;

punto3.y -=3;

System.***out***.println("El punto 1: " + punto1.x + " , " + punto1.y);

System.***out***.println("El punto 2: " + punto2.x+ " , " + punto2.y);

System.***out***.println("El punto 3: " + punto3.x+ " , " + punto3.y);

public class Punt {

int x,y;

public Punt(int x, int y) {

this.x = x;

this.y = y;

}

public int getX() {

return x;

}

public int getY() {

return y;

}

public void setX(int x) {

this.x = x;

}

public void setY(int y) {

this.y = y;

}

}

**Exercici C1 – Punt**

Modifica els atributs de Punt perquè siguen private. Fixa't que des del main ja no et deixarà utilitzar ni modificar els atributs x e y dels objectes.

Afegirem els getteres: int getX() i int getY() que retornaran els valors de x e i respectivament. És una forma indirecta de llegir els seus valors.

Afegirem també els setters: void setX(int x) i void setY(int y) que copiaran el valor passat com a paràmetre als atributs de la classe.

Tant getters com a setters han de ser public.

Corregeix el main per a utilitzar els getters i setters. Prova a instanciar diversos objectes, mostrar els seus valors per pantalla, modificar-los, etc.

public class Punt {

private int x,y;

public Punt(int x, int y) {

this.x = x;

this.y = y;

}

public int getX() {

return x;

}

public int getY() {

return y;

}

public void setX(int x) {

this.x = x;

}

public void setY(int y) {

this.y = y;

}

}

public class Punt\_main {

public static void main(String[] args) {

Punt punto1 = new Punt(5,0);

Punt punto2 = new Punt(10,10);

Punt punto3 = new Punt(-3,7);

System.***out***.println("El punto 1: " + punto1.getX()+ " , " + punto1.getY());

System.***out***.println("El punto 2: " + punto2.getX()+ " , " + punto2.getY());

System.***out***.println("El punto 3: " + punto3.getX()+ " , " + punto3.getY());

punto1.setX(punto1.getX() + 3);

punto1.setY(punto1.getY() + 9);

punto2.setX(punto2.getX() + 3);

punto2.setY(punto2.getY() + 9);

punto3.setX(punto3.getX() + 3);

punto3.setY(punto3.getY() + 9);

System.***out***.println("El punto 1: " + punto1.getX()+ " , " + punto1.getY());

System.***out***.println("El punto 2: " + punto2.getX()+ " , " + punto2.getY());

System.***out***.println("El punto 3: " + punto3.getX()+ " , " + punto3.getY());

}

}

**Exercici D1 – Punt**

Afig a la classe Punt els següents mètodes públics:

● public void imprimeix() // Imprimeix per pantalla les coordenades. Exemple: “(7, -5)”

● public void setXY(int x, int y) // Modifica totes dues coordenades. És com un setter doble.

● public void desplaça(int dx, int dy) // Desplaça el punt la quantitat (dx,dy) indicada.

Exemple: Si el punt (1,1) es desplaça (2,5) llavors estarà en (3,6).

● public int distancia(Punt p) // Calcula i retorna la distància entre el propi objecte (this) i un altre objecte (Punt p) que es passa com a paràmetre: distància entre dues coordenades.

Prova d'utilitzar aquests mètodes des del main per a comprovar el seu funcionament.

public class Punt {

private int x,y;

public Punt(int x, int y) {

this.x = x;

this.y = y;

}

public int getX() {

return x;

}

public int getY() {

return y;

}

public void setX(int x) {

this.x = x;

}

public void setY(int y) {

this.y = y;

}

public void imprimeix() {

System.***out***.println( + this.x + " , " + this.y );

}

public void setXY(int a, int b) {

this.x = a;

this.y = b;

}

public void desplaça (int dx, int dy) {

this.x += dx;

this.y += dy;

}

public double distancia( Punt p) {

return Math.*sqrt*(Math.*pow*(this.x - p.x, 2) + Math.*pow*(this.y - p.y, 2));

}

}

public class Punt\_main {

public static void main(String[] args) {

Punt punto1 = new Punt(5,0);

Punt punto2 = new Punt(10,10);

Punt punto3 = new Punt(-3,7);

punto1.imprimeix();

punto2.imprimeix();

punto3.imprimeix();

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

punto1.setXY(punto1.getX()+3, punto1.getY() - 8);

punto2.setXY(punto2.getX()/3, punto2.getY() \* 8);

punto3.setXY(punto3.getX()+3, punto3.getY() % 8);

punto1.imprimeix();

punto2.imprimeix();

punto3.imprimeix();

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

punto1.desplaça(2, 3);

punto2.desplaça(2, 3);

punto3.desplaça(2, 3);

punto1.imprimeix();

punto2.imprimeix();

punto3.imprimeix();

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

System.***out***.println("La distancia entre el punto1 y el punto2 es: " + punto1.distancia(punto2));

System.***out***.println("La distancia entre el punto2 y el punto3 es: " + punto2.distancia(punto3));

System.***out***.println("La distancia entre el punto1 y el punto3 es: " + punto1.distancia(punto3));

}

}

**Exercici E1– Punt**

Necessitem un mètode que ens permeta crear un objecte Punt amb coordenades aleatòries.

Aquesta funcionalitat no depén de cap objecte concret per la qual cosa serà estàtica. Haurà de

crear un nou Punt (utilitza el constructor) amb x e i entre -100 i 100, i després retornar-lo (amb

return).

● public static Punt creaPuntoAleatorio()

Prova’l en el main per a comprovar que funciona. Crea diversos punts aleatoris amb

Punt.creaPuntAleatori() i imprimeix el seu valor per pantalla.

public static Punt creaPuntoAleatorio() {

Random r = new Random();

Punt p = new Punt(-100 + r.nextInt(200), -100 + r.nextInt(200));

return p;

public class Punt\_main {

public static void main(String[] args) {

Punt punt\_aleat1 = Punt.*creaPuntoAleatorio*();

Punt punt\_aleat2 = Punt.*creaPuntoAleatorio*();

punt\_aleat1.imprimeix();

punt\_aleat2.imprimeix();

punt\_aleat1.setXY(punt\_aleat1.getX()+3, punt\_aleat1.getY()\*3);

punt\_aleat2.setXY(punt\_aleat2.getX()-3, punt\_aleat2.getY()/3);

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

punt\_aleat1.imprimeix();

punt\_aleat2.imprimeix();

//Creamos un array de puntos aleatorios

Punt puntos[] = new Punt[20];

for (int i=0;i<20; i++) {

puntos[i] = Punt.*creaPuntoAleatorio*();

System.***out***.println(" Punt aleatori " + i + ": ");

puntos[i].imprimeix();

}

**Exercici A2 – Persona**

Crea un programa amb una classe anomenada Persona que representarà les dades principals d'una

persona: dni, nom, cognoms i edat.

En el main de la classe principal instància dos objectes de la classe Persona. Després, demana per

teclat les dades de totes dues persones (guarda’ls en els objectes). Finalment, imprimeix dos

missatges per pantalla (un per objecte) amb un missatge de l'estil “Ferrandis Lujan Garcia amb DNI

… és / no és major d'edat”.

public class Persona {

int edat;

String dni;

String nom, cognom;

}

public class Persona\_main {

public static void main(String[] args) {

Persona persona1 = new Persona();

Persona persona2 = new Persona();

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

System.***out***.println("Introduce el nombre de la primera persona: ");

persona1.nom = entrada.next();

System.***out***.println("Introduce el apellido de la primera persona: ");

persona1.cognom = entrada.next();

System.***out***.println("Introduce la edad de la primera persona: ");

persona1.edat = entrada.nextInt();

System.***out***.println("Itroduce el dni de la primera persona: ");

persona1.dni = entrada.next();

if (persona1.edat<18) {

System.***out***.println(persona1.nom + " " + persona1.cognom + " amb DNI " + persona1.dni + " es menor de edad." );

}else {

System.***out***.println(persona1.nom + " " + persona1.cognom + " amb DNI " + persona1.dni + " es mayor de edad. ");

}

entrada.close();

}

**Exercici B2 – Persona**

Afig a Persona el constructor de baix i corregeix el main per a utilitzarlo:

*public Persona(String dni, String nom, String cognoms, int edat) {*

*this.dni = dni;*

*this.nom = nom;*

*this.cognoms = cognoms;*

*this.edat = edat;*

*}*

public class Persona {

String dni;

String nom;

String cognoms;

int edat;

public Persona(String dni, String nom, String cognoms, int edat) {

this.dni = dni;

this.nom = nom;

this.cognoms = cognoms;

this.edat = edat;

}

public class ProgramaPersona {

public static void main(String[] args) {

Persona persona1 = new Persona("18999548P", "Josep", "Serrà Ferrandis", 25);

Persona persona2 = new Persona("20222444L", "Laia", "Barallat Sanxis", 17);

String cadena1 = persona1.nom + " " + persona1.cognoms + " amb DNI " + persona1.dni;

String cadena2 = persona2.nom + " " + persona2.cognoms + " amb DNI " + persona2.dni;

if (persona1.edat >= 18) {

cadena1 += " és major d'edat";

} else {

cadena1 += " no és major d'edat";

}

if (persona2.edat >= 18) {

cadena2 += " és major d'edat";

} else {

cadena2 += " no és major d'edat";

}

System.***out***.println(cadena1);

System.***out***.println(cadena2);

}

**Exercici C2 – Persona**

Aplica l’encapsulament bàsic a la classe Persona: Declara tots els seus atributs com private i crea

tots els getters i setters necessaris (un get i un set per atribut).

Corregeix el main per a utilitzar els getters i setters. Prova a instanciar diversos objectes, mostrar

els seus valors per pantalla, modificar-los, etc.

public class Persona {

private String dni;

private String nom;

private String cognoms;

private int edat;

public Persona(String dni, String nom, String cognoms, int edat) {

this.dni = dni;

this.nom = nom;

this.cognoms = cognoms;

this.edat = edat;

}

public String getDni() {

return dni;

}

public String getNom() {

return nom;

}

public String getCognoms() {

return cognoms;

}

public int getEdat() {

return edat;

}

public void setDni(String dni) {

this.dni = dni;

}

public void setNom(String nom) {

this.nom = nom;

}

public void setCognoms(String cognoms) {

this.cognoms = cognoms;

}

public void setEdat(int edat) {

this.edat = edat;

}

public class ProgramaPersona {

public static void main(String[] args) {

Persona persona1 = new Persona("18999548P", "Josep", "Serrà Ferrandis", 25);

Persona persona2 = new Persona("20222444L", "Laia", "Barallat Sanxis", 17);

String cadena1 = persona1.getNom() + " " + persona1.getCognoms() + " amb DNI " + persona1.getDni();

String cadena2 = persona2.getNom() + " " + persona2.getCognoms() + " amb DNI " + persona2.getDni();

if (persona1.getEdat() >= 18) {

cadena1 += " és major de edat";

} else {

cadena1 += " no és major de edat";

}

if (persona2.getEdat() >= 18) {

cadena2 += " és major de edat";

} else {

cadena2 += " no és major de edat";

}

System.***out***.println(cadena1);

System.***out***.println(cadena2);

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

persona1.setNom("Joan");

persona1.setEdat(11);

persona2.setNom("Carme");

persona2.setEdat(33);

cadena1 = persona1.getNom() + " " + persona1.getCognoms() + " amb DNI " + persona1.getDni();

cadena2 = persona2.getNom() + " " + persona2.getCognoms() + " amb DNI " + persona2.getDni();

if (persona1.getEdat() >= 18) {

cadena1 += " és major de edat";

} else {

cadena1 += " no és major de edat";

}

if (persona2.getEdat() >= 18) {

cadena2 += " és major de edat";

} else {

cadena2 += " no és major de edat";

}

System.***out***.println(cadena1);

System.***out***.println(cadena2);

}

**Exercici D2 – Persona**

Afig a la classe Persona els següents mètodes públics:

● public void imprimeix() // Imprimeix la informació de l'objecte: “DNI:… Nom:… etc.”

● public boolean esMajorEdat() // Retorna true si és major d'edat (false si no).

● public boolean esJubilat() // Retorna true si té 65 anys o més (false si no).

● public int diferenciaEdat(Persona p) // Retorna la diferència d'edat entre ‘this’ i p.

public class Persona {

private String dni;

private String nom;

private String cognoms;

private int edat;

public Persona(String dni, String nom, String cognoms, int edat) {

this.dni = dni;

this.nom = nom;

this.cognoms = cognoms;

this.edat = edat;

}

public String getDni() {

return dni;

}

public String getNom() {

return nom;

}

public String getCognoms() {

return cognoms;

}

public int getEdat() {

return edat;

}

public void setDni(String dni) {

this.dni = dni;

}

public void setNom(String nom) {

this.nom = nom;

}

public void setCognoms(String cognoms) {

this.cognoms = cognoms;

}

public void setEdat(int edat) {

this.edat = edat;

}

// Rertorna si és menor d'edat

public boolean esMenor() {

return this.edat < 18;

}

// Retorna si està jubilat

public boolean esJubilat() {

return this.edat >= 65;

}

//Devuelve la diferencia de edat entre este objeto y el recibido

public int diferenciaEdat(Persona p) {

return this.edat - p.edat;

}

public void mostra() {

System.***out***.println("DNI: " + this.dni);

System.***out***.println("Nom: " + this.nom);

System.***out***.println("Cognoms: " + this.cognoms);

System.***out***.println("Edat: " + this.edat);

}

public class ProgramaPersona {

public static void main(String[] args) {

// Instanciem persones

Persona persona1 = new Persona("18999548P", "Josep", "Serrà Ferrandis", 25);

Persona persona2 = new Persona("20222444L", "Laia", "Barallat Sanxís", 17);

// Mostrem p1 y p2

persona1.mostra();

persona2.mostra();

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

// Realitzem alguns canvis

persona1.setNom("Joan");

persona1.setEdat(11);

persona2.setNom("Carme");

persona2.setEdat(33);

// Mostrem p1 y p2

persona1.mostra();

persona2.mostra();

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

// Mostrem diferència d'edat

System.***out***.println("La diferència d'edat és " + persona1.diferenciaEdat(persona2));

}

**Exercici E2 – Persona**

**El DNI d'una persona no pot variar**. Afig el modificador final a l'atribut dni i assegura't que es

guarde el seu valor en el constructor. Lleva el mètode setDNI(…) que de totes maneres ja no es

podrà utilitzar perquè Java no et deixarà modificar l'atribut dni.

**La majoria d'edat als 18 anys és un valor comú a totes les persones i no pot variar**. Crea un nou

atribut anomenat majoriaEdad que siga static i final. Hauràs d'inicialitzar-lo a 18 en la declaració.

Utilitza-ho en el mètode que comprova si una persona és major d'edat.

Crea un mètode **static boolean validarDNI(String dni)** que retorne true si dni és vàlid (té 8

números i una lletra). Si no, retornarà false. Utilitza’l en el constructor per a comprovar el dni (si no

és vàlid, mostra un missatge d'error i no guardes els valors).

Realitza algunes proves en el main per a comprovar el funcionament dels canvis realitzats. També

pots utilitzar Persona.validarDNI(…) per exemple per a comprovar si uns DNI introduïts per teclat

són vàlids o no (sense necessitat de crear cap objecte).

public class Persona {

private String dni;

private String nom;

private String cognoms;

private int edat;

public Persona(String dni, String nom, String cognoms, int edat) {

this.dni = dni;

this.nom = nom;

this.cognoms = cognoms;

this.edat = edat;

}

public String getDni() {

return dni;

}

public String getNom() {

return nom;

}

public String getCognoms() {

return cognoms;

}

public int getEdat() {

return edat;

}

public void setDni(String dni) {

this.dni = dni;

}

public void setNom(String nom) {

this.nom = nom;

}

public void setCognoms(String cognoms) {

this.cognoms = cognoms;

}

public void setEdat(int edat) {

this.edat = edat;

}

// Rertorna si és menor d'edat

public boolean esMenor() {

return this.edat < 18;

}

// Retorna si està jubilat

public boolean esJubilat() {

return this.edat >= 65;

}

//Devuelve la diferencia de edat entre este objeto y el recibido

public int diferenciaEdat(Persona p) {

return this.edat - p.edat;

}

public void mostra() {

System.***out***.println("DNI: " + this.dni);

System.***out***.println("Nom: " + this.nom);

System.***out***.println("Cognoms: " + this.cognoms);

System.***out***.println("Edat: " + this.edat);

}

// Funció que rep un DNI i retorna si és vàlid o no. COMPTE!! no comprova que la lletra siga correcta!

public static boolean validarDNI(String valor) {

// Expresió regular que indica 8 dígits i al final alguna de les lletres que se permeten

// rangs A-H J-N P-T V-Z (recordeu, algunes no se permeten)

String dniRegexp = "\\d{8}[A-HJ-NP-TV-Z]";

// Retorna true si se compleix la expresió regular

return Pattern.*matches*(dniRegexp, valor);

}

public class ProgramaPersona {

public static void main(String[] args) {

// Instanciem persones

Persona persona1 = new Persona("18999548P", "Josep", "Serrà Ferrandis", 25);

Persona persona2 = new Persona("20222444L", "Laia", "Barallat Sanxís", 17);

// Mostrem p1 y p2

persona1.mostra();

persona2.mostra();

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

// Realitzem alguns canvis

persona1.setNom("Joan");

persona1.setEdat(11);

persona2.setNom("Carme");

persona2.setEdat(33);

// Mostrem p1 y p2

persona1.mostra();

persona2.mostra();

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

// Mostrem diferència d'edat

System.***out***.println("La diferència d'edat és " + persona1.diferenciaEdat(persona2));

//Validem alguns DNIs

System.***out***.println("DNI 12345678Z => " + ((Persona.validarDNI("12345678Z")) ? "Valido" : "No vàlid") );

System.***out***.println("DNI 1234567Z => " + ((Persona.validarDNI("1234567Z")) ? "Valido" : "No vàlid") );

System.***out***.println("DNI 123456789 => " + ((Persona.validarDNI("123456789")) ? "Valido" : "No vàlid") );

}

**Exercici A3 – Rectangle**

Crea un programa amb una classe anomenada Rectangle que representarà un rectangle mitjançant

dues coordenades (x1,y1) i (x2,y2) en un pla, per la qual cosa la classe haurà de tindre quatre

atributs sencers: **x1**, **y1**, **x2**, **y2**.

En el main de la classe principal instància 2 objectes Rectangle en (0,0)(5,5) i (7,9)(2,3). Mostra per

pantalla les seues coordenades, perímetres (suma de costats) i àrees (ample x alt). Modifica totes

les coordenades com consideres i torna a imprimir coordenades, perímetres i àrees.

public class Rectangle {

int x1, y1, x2, y2;

}

public class ProgramaRectangle {

public static void main(String[] args) {

Rectangle rec1 = new Rectangle();

Rectangle rec2 = new Rectangle();

rec1.x1 = 0;

rec1.y1 = 0;

rec1.x2 = 5;

rec1.y2 = 5;

rec2.x1 = 7;

rec2.y1 = 9;

rec2.x2 = 2;

rec2.y2 = 3;

System.***out***.println("Coordenades del rectangle 1 (" + rec1.x1 + "," + rec1.y1 + ") i (" + rec1.x2 + "," + rec1.y2 + ")");

System.***out***.println("Coordenades del rectangle 2 (" + rec2.x1 + "," + rec2.y1 + ") i (" + rec2.x2 + "," + rec2.y2 + ")");

System.***out***.println("El perímetre del rectangle 1 és: " + *perimetre*(rec1));

System.***out***.println("El perímetre del rectangle 2 és: " + *perimetre*(rec2));

System.***out***.println("L'àrea del rectangle 1 és: " + *area*(rec1));

System.***out***.println("L'àrea del rectangle 2 és: " + *area*(rec2));

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

rec1.x1 = 5;

rec1.y1 = 5;

rec1.x2 = 15;

rec1.y2 = 15;

rec2.x1 = 17;

rec2.y1 = 19;

rec2.x2 = 22;

rec2.y2 = 24;

System.***out***.println("Coordenades del rectangle 1 (" + rec1.x1 + "," + rec1.y1 + ") i (" + rec1.x2 + "," + rec1.y2 + ")");

System.***out***.println("Coordenades del rectangle 2 (" + rec2.x1 + "," + rec2.y1 + ") i (" + rec2.x2 + "," + rec2.y2 + ")");

System.***out***.println("El perímetre del rectangle 1 és: " + *perimetre*(rec1));

System.***out***.println("El perímetre del rectangle 2 és: " + *perimetre*(rec2));

System.***out***.println("L'àrea del rectangle 1 és: " + *area*(rec1));

System.***out***.println("L'àrea del rectangle 2 és: " + *area*(rec2));

}

public static double perimetre(Rectangle rect) {

int costat1 = Math.*abs*(rect.x1 - rect.x2);

int costat2 = Math.*abs*(rect.y1 - rect.y2);

return (costat1 + costat2) \* 2;

}

public static double area(Rectangle rect) {

int costat1 = Math.*abs*(rect.x1 - rect.x2);

int costat2 = Math.*abs*(rect.y1 - rect.y2);

return costat1 \* costat2;

}

**Exercici B3 – Rectangle**

En el nostre programari necessitem assegurar-nos que la coordenada (x1,y1) represente la cantonada inferior esquerra i la (x2,y2) la superior dreta del rectangle, com en el dibuix.

Afig a Rectangle un constructor amb els 4 paràmetres. Inclou un if que comprove els valors **(\*)**. Si són vàlids guardarà els paràmetres en l'objecte.

Si no ho són mostrarà un missatge de l'estil “ERROR al instanciar Rectangle...” utilitzant System.err.println(…). No podrem evitar que s’instancie l'objecte però almenys avisarem per pantalla.

Corregeix el main per a utilitzar aquest constructor. Hauria de mostrar un missatge d'error.

**(\*) Pista:** És suficient amb un if ( (condició) && (condició) )

public class Rectangle {

int x1, y1, x2, y2;

public Rectangle(int x1, int y1, int x2, int y2) {

// Comprovem si és un rectangle vàlid

if ((x1 < x2) && (y1 < y2)) {

this.x1 = x1;

this.y1 = y1;

this.x2 = x2;

this.y2 = y2;

} else {

System.***err***.println("ERROR al instanciar el Rectangle (" + x1 + "," + y1 + "),(" + x2 + "," + y2 + ")");

}

public class ProgramaRectangle {

public static void main(String[] args) {

Rectangle rec1 = new Rectangle(0, 0, 5, 5);

Rectangle rec2 = new Rectangle(7, 9, 2, 3);

System.***out***.println("Coordenades del rectangle 1 (" + rec1.x1 + "," + rec1.y1 + ") i (" + rec1.x2 + "," + rec1.y2 + ")");

System.***out***.println("Coordenades del rectangle 2 (" + rec2.x1 + "," + rec2.y1 + ") i (" + rec2.x2 + "," + rec2.y2 + ")");

System.***out***.println("El perímetre del rectangle 1 és: " + *perimetre*(rec1));

System.***out***.println("El perímetre del rectangle 2 és: " + *perimetre*(rec2));

System.***out***.println("L'àrea del rectangle 1 és: " + *area*(rec1));

System.***out***.println("L'àrea del rectangle 2 és: " + *area*(rec2));

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

}

public static double perimetre(Rectangle rect) {

int costat1 = Math.*abs*(rect.x1 - rect.x2);

int costat2 = Math.*abs*(rect.y1 - rect.y2);

return (costat1 + costat2) \* 2;

}

public static double area(Rectangle rect) {

int costat1 = Math.*abs*(rect.x1 - rect.x2);

int costat2 = Math.*abs*(rect.y1 - rect.y2);

return costat1 \* costat2;

**Exercici C3 – Rectangle**

Aplica l’encapsulament bàsic a la classe Rectangle: Declara tots els seus atributs com private i crea tots els getters i setters necessaris (un get i un set per atribut).

Recordes la condició explicada en B3? Tindràs que programar els setters de manera que comproven el valor passat com a argument abans de guardar-lo en l'objecte. Si no fora correcte es mostrarà un missatge d'error (i NO es guardarà el valor).

Corregeix el main per a utilitzar els getters i setters. Prova a instanciar diversos objectes, mostrar els seus valors, modificar-los, etc. Prova diversos valors erronis per a comprovar si funciona.

public class Rectangle {

private int x1;

private int y1;

private int x2;

private int y2;

public Rectangle(int x1, int y1, int x2, int y2) {

// Comprovem si és un rectangle vàlid

if ((x1 < x2) && (y1 < y2)) {

this.x1 = x1;

this.y1 = y1;

this.x2 = x2;

this.y2 = y2;

} else {

System.***err***.println("ERROR al instanciar el Rectangle (" + x1 + "," + y1 + "),(" + x2 + "," + y2 + ")");

}

}

public int getX1() {

return x1;

}

public int getY1() {

return y1;

}

public int getX2() {

return x2;

}

public int getY2() {

return y2;

}

public void setX1(int x1) {

if (x1 > this.x2) {

System.***err***.println("ERROR el valor de x1 no pot ser major que el valor de x2");

} else {

this.x1 = x1;

}

}

public void setY1(int y1) {

if (y1 > this.y2) {

System.***err***.println("ERROR el valor de y1 no puede ser mayor que el valor de y2");

} else {

this.y1 = y1;

}

}

public void setX2(int x2) {

if (x2 < this.x1) {

System.***err***.println("ERROR el valor de x2 no puede ser menor que el valor de x1");

} else {

this.x2 = x2;

}

}

public void setY2(int y2) {

if (y2 < this.y1) {

System.***err***.println("ERROR el valor de y2 no puede ser menor que el valor de y1");

} else {

this.y2 = y2;

}

public class ProgramaRectangle {

public static void main(String[] args) {

Rectangle rec1 = new Rectangle(0, 0, 5, 5);

Rectangle rec2 = new Rectangle(7, 9, 2, 3);

System.***out***.println("Coordenades del rectangle 1 (" + rec1.getX1() + "," + rec1.getY1() + ") i (" + rec1.getX2() + "," + rec1.getY2() + ")");

System.***out***.println("Coordenades del rectangle 2 (" + rec2.getX1() + "," + rec2.getY1() + ") i (" + rec2.getX2() + "," + rec2.getY2() + ")");

System.***out***.println("El perímetre del rectangle 1 és: " + *perimetre*(rec1));

System.***out***.println("El perímetre del rectangle 2 és: " + *perimetre*(rec2));

System.***out***.println("L'àrea del rectangle 1 és: " + *area*(rec1));

System.***out***.println("L'àrea del rectangle 2 és: " + *area*(rec2));

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

rec1.setX1(10);

rec1.setX2(3);

rec2.setY1(12);

rec2.setY2(5);

System.***out***.println("Coordenades del rectangle 1 (" + rec1.getX1() + "," + rec1.getY1() + ") i (" + rec1.getX2() + "," + rec1.getY2() + ")");

System.***out***.println("Coordenades del rectangle 2 (" + rec2.getX1() + "," + rec2.getY1() + ") i (" + rec2.getX2() + "," + rec2.getY2() + ")");

System.***out***.println("El perímetre del rectangle 1 és: " + *perimetre*(rec1));

System.***out***.println("El perímetre del rectangle 2 és: " + *perimetre*(rec2));

System.***out***.println("L'àrea del rectangle 1 és: " + *area*(rec1));

System.***out***.println("L'àrea del rectangle 2 és: " + *area*(rec2));

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

}

public static double perimetre(Rectangle rect) {

int lado1 = Math.*abs*(rect.getX1() - rect.getX2());

int lado2 = Math.*abs*(rect.getY1() - rect.getY2());

return (lado1 + lado2) \* 2;

}

public static double area(Rectangle rect) {

int lado1 = Math.*abs*(rect.getX1() - rect.getX2());

int lado2 = Math.*abs*(rect.getY1() - rect.getY2());

return lado1 \* lado2;

}

**Exercici D3 – Rectangle**

Afig a la classe Rectngle mètodes públics amb les següents funcionalitats:

● Mètode per a imprimir la informació del rectangle per pantalla.

● Mètodes setters dobles i quàdruples: **setX1Y1**, **set X2Y2** i **setAll(…)**.

● Mètodes **getPerimetre** i **getArea** que calculen i retornen el perímetre i àrea de l'objecte.

Prova d'utilitzar aquests mètodes des del main per a comprovar el seu funcionament.

public class Rectangle {

private int x1;

private int y1;

private int x2;

private int y2;

public Rectangle(int x1, int y1, int x2, int y2) {

// Comprobem si és un rectangle vàlid

if ((x1 < x2) && (y1 < y2)) {

this.x1 = x1;

this.y1 = y1;

this.x2 = x2;

this.y2 = y2;

} else {

System.***err***.println("ERROR al instanciar el Rectangle (" + x1 + "," + y1 + "),(" + x2 + "," + y2 + ")");

}

}

public int getX1() {

return x1;

}

public int getY1() {

return y1;

}

public int getX2() {

return x2;

}

public int getY2() {

return y2;

}

public void setX1(int x1) {

if (x1 > this.x2) {

System.***err***.println("ERROR el valor de x1 no pot ser major que el valor de x2");

} else {

this.x1 = x1;

}

}

public void setY1(int y1) {

if (y1 > this.y2) {

System.***err***.println("ERROR el valor de y1 no pot ser major que el valor de y2");

} else {

this.y1 = y1;

}

}

public void setX2(int x2) {

if (x2 < this.x1) {

System.***err***.println("ERROR el valor de x2 no pot ser menor que el valor de x1");

} else {

this.x2 = x2;

}

}

public void setY2(int y2) {

if (y2 < this.y1) {

System.***err***.println("ERROR el valor de y2 no pot ser menor que el valor de y1");

} else {

this.y2 = y2;

}

}

// Setter X1Y1

public void setX1Y1(int x1, int y1) {

this.setX1(x1);

this.setY1(y1);

}

// Setter X2Y2

public void setX2Y2(int x2, int y2) {

this.setX2(x2);

this.setY2(y2);

}

// Setter X1Y1 X2Y2 (tots)

public void setAll(int x1, int y1, int x2, int y2) {

this.setX1(x1);

this.setY1(y1);

this.setX2(x2);

this.setY2(y2);

}

// Obtindre el perímetre

public int getPerimetre() {;

return *abs*(this.x1 - this.x2) \* 2 + *abs*(this.y1 - this.y2) \* 2;

}

// Obtindre l'àrea

public int getArea() {

return *abs*(this.x1 - this.x2) \* *abs*(this.y1 - this.y2);

}

// Mostrar informació rectangle

public void mostra() {

System.***out***.println("Coordenades: (" + this.x1 + "," + this.y1 + ")-(" + this.x2 + "," + this.y2 + ")");

System.***out***.println("Perímetre: " + this.getPerimetre());

System.***out***.println("Àrea: " + this.getArea());

}

public class ProgramaRectangle {

public static void main(String[] args) {

// Instanciem i mostrem rectangles

Rectangle rec1 = new Rectangle(0, 0, 5, 5);

Rectangle rec2 = new Rectangle(7, 9, 2, 3);

rec1.mostra();

rec2.mostra();

// Canviem valors i mostrem

rec1.setX1(10);

rec1.setX2(3);

rec2.setY1(12);

rec2.setY2(5);

rec1.mostra();

rec2.mostra();

// Canviem valores i mostrem

rec1.setX1Y1(2, 4);

rec1.setX2Y2(10, 20);

rec2.setAll(3, 4, 5, 6);

rec1.mostra();

rec2.mostra();

}

**Exercici E3 – Rectangle**

Necessitem fer alguns canvis perquè totes les coordenades estiguen entre (0,0) i (100,100). Afig a

la classe Rectangle dos atributs anomenats min i max. Aquests valors són comuns a tots els

objectes i no poden variar. Pensa que modificaions necessites afegir a min i max.

Utilitza min i max en el constructor i en els setters per a comprovar els valors (com de costum, si no

són correctes mostra un missatge d'error i aplicacions els canvis).

També necessitem un mètode no constructor per a crear rectangles aleatoris. Implementa-ho.

Realitza proves en el main per a comprovar el seu funcionament.

public class Rectangle {

private int x1;

private int y1;

private int x2;

private int y2;

private final static int ***min*** = 0;

private final static int ***max*** = 100;

public Rectangle(int x1, int y1, int x2, int y2) {

// Comprobem si és un rectangle vàlid

if ((x1 < x2) && (y1 < y2) && (x1 >= Rectangle.***min*** && x1 <= Rectangle.***max***) &&

(x2 >= Rectangle.***min*** && x2 <= Rectangle.***max***) && (y1 >= Rectangle.***min*** &&

y1 <= Rectangle.***max***) && (y2 >= Rectangle.***min*** && y2 <= Rectangle.***max***)) {

this.x1 = x1;

this.y1 = y1;

this.x2 = x2;

this.y2 = y2;

} else {

System.***err***.println("ERROR al instanciar el Rectangle (" + x1 + "," + y1 + "),(" + x2 + "," + y2 + ")");

}

}

public int getX1() {

return x1;

}

public int getY1() {

return y1;

}

public int getX2() {

return x2;

}

public int getY2() {

return y2;

}

public void setX1(int x1) {

if (x1 > this.x2) {

System.***err***.println("ERROR el valor de x1 no pot ser major que el valor de x2");

} else if (x1 < Rectangle.***min*** || x1 > Rectangle.***max***) {

System.***err***.println("ERROR el valor ha d'estar entre " + this.***min*** + " i " + this.***max***);

} else {

this.x1 = x1;

}

}

public void setY1(int y1) {

if (y1 > this.y2) {

System.***err***.println("ERROR el valor de y1 no pot ser major que el valor de y2");

} else if (y1 < Rectangle.***min*** || y1 > Rectangle.***max***) {

System.***err***.println("ERROR el valor ha d'estar entre " + this.***min*** + " i " + this.***max***);

} else {

this.y1 = y1;

}

}

public void setX2(int x2) {

if (x2 < this.x1) {

System.***err***.println("ERROR el valor de x2 no pot ser menor que el valor de x1");

} else if (x2 < Rectangle.***min*** || x2 > Rectangle.***max***) {

System.***err***.println("ERROR el valor ha d'estar entre " + this.***min*** + " i " + this.***max***);

} else {

this.x2 = x2;

}

}

public void setY2(int y2) {

if (y2 < this.y1) {

System.***err***.println("ERROR el valor de y2 no pot ser menor que el valor de y1");

} else if (y2 < Rectangle.***min*** || y2 > Rectangle.***max***) {

System.***err***.println("ERROR el valor ha d'estar entre " + this.***min*** + " i " + this.***max***);

} else {

this.y2 = y2;

}

}

// Setter X1Y1

public void setX1Y1(int x1, int y1) {

this.setX1(x1);

this.setY1(y1);

}

// Setter X2Y2

public void setX2Y2(int x2, int y2) {

this.setX2(x2);

this.setY2(y2);

}

// Setter X1Y1 X2Y2 (tots)

public void setAll(int x1, int y1, int x2, int y2) {

this.setX1(x1);

this.setY1(y1);

this.setX2(x2);

this.setY2(y2);

}

// Obtindre el perímetre

public int getPerimetre() {;

return *abs*(this.x1 - this.x2) \* 2 + *abs*(this.y1 - this.y2) \* 2;

}

// Obtindre l'àrea

public int getArea() {

return *abs*(this.x1 - this.x2) \* *abs*(this.y1 - this.y2);

}

// Mostrar informació rectangle

public void mostra() {

System.***out***.println("Coordenades: (" + this.x1 + "," + this.y1 + ")-(" + this.x2 + "," + this.y2 + ")");

System.***out***.println("Perímetre: " + this.getPerimetre());

System.***out***.println("Àrea: " + this.getArea());

}

public static Rectangle generarRectangle() {

Random r = new Random();

int a, b, c, d;

a = r.nextInt(Rectangle.***max***);

b = r.nextInt(Rectangle.***max***);

c = *min*(100, a + r.nextInt(Rectangle.***max***));

d = *min*(100, b + r.nextInt(Rectangle.***max***));

Rectangle tmp = new Rectangle(a, b, c, d);

return tmp;

}

public class ProgramaRectangle {

public static void main(String[] args) {

// Instanciem i mostrem rectangles

Rectangle rec1 = new Rectangle(0, 0, 5, 5);

Rectangle rec2 = new Rectangle(7, 9, 2, 3);

rec1.mostra();

rec2.mostra();

// Canviem valors i mostrem

rec1.setX1(10);

rec1.setX2(3);

rec2.setY1(12);

rec2.setY2(5);

rec1.mostra();

rec2.mostra();

// Canviem valores i mostrem

rec1.setX1Y1(2, 4);

rec1.setX2Y2(10, 20);

rec2.setAll(3, 4, 5, 6);

rec1.mostra();

rec2.mostra();

rec2.setAll(300, 400, 500, 600);

System.***out***.println("Rectangle aleatori");

Rectangle rec3 = Rectangle.generarRectangle();

rec3.mostra();

**Exercici A4 – Artícle**

Crea un programa amb una classe anomenada Artícle amb els següents atributs: **nom**, **preu** (sense

IVA), **iva** (sempre serà 21) i **cuantsQueden** (representa quants queden en el magatzem).

En el main de la classe principal instància un objecte de la classe article. Assigna-li valors a tots els

seus atributs (els que vulgues) i mostra per pantalla un missatge de l'estil “Pijama - Preu:10€ -

IVA:21% - PVP:12,1€” (el PVP és el preu de venda al públic, és a dir, el preu amb IVA). Després,

canvia el preu i torna a imprimir el missatge.

public class Article {

String nom;

double preu;

int iva;

int quantsEnQueden;

}

public class ProgramaArticle {

public static void main(String[] args) {

Article a1 = new Article();

a1.nom = "Camisa de quadres";

a1.preu = 20;

a1.iva = 21;

a1.quantsEnQueden = 5;

System.***out***.println(a1.nom + " - Preu: " + a1.preu + "€ - IVA: " + a1.iva + "% - PVP: " + (a1.preu + (a1.preu \* a1.iva / 100)) + "€");

a1.preu = 10;

System.***out***.println(a1.nom + " - Preu: " + a1.preu + "€ - IVA: " + a1.iva + "% - PVP: " + (a1.preu + (a1.preu \* a1.iva / 100)) + "€");

**Exercici B4 – Article**

Afig un constructor amb 4 paràmetres que assigne valors a nom, preu, iva i cuantsQueden. Aquest constructor haurà de mostrar un missatge d'error si algun dels valors nom, preu, iva o cuantsQueden no són vàlids. Què coindicions creus que podrien determinar si son vàlids o no?

Raona-ho i implementa el codi.

Corregeix el main i prova de crear diversos articles. Introdueix alguns amb valors incorrectes per a comprovar si avisa de l'error.

public class Article {

String nom;

double preu;

int iva;

int quantsEnQueden;

public Article(String nom, double preu, int iva, int quantsEnQueden) {

if (nom.equals("")) {

System.***err***.println("ERROR: El nom no pot estar buit");

} else if (preu <= 0) {

System.***err***.println("ERROR: El preu no pot ser menor o igual a cero");

} else if (iva != 21) {

System.***err***.println("ERROR: L'iva ha de ser del 21%");

} else if (quantsEnQueden < 0) {

System.***err***.println("ERROR: El stock no pot ser menor de cero");

} else {

this.nom = nom;

this.preu = preu;

this.iva = iva;

this.quantsEnQueden = quantsEnQueden;

}

public class ProgramaArticle {

public static void main(String[] args) {

Article a1 = new Article("Camisa de quadres", 20, 21, 5);

Article a2 = new Article("Pantaló de pana", 100, 10, -5);

System.***out***.println(a1.nom + " - Preu: " + a1.preu + "€ - IVA: " + a1.iva + "% - PVP: " + (a1.preu + (a1.preu \* a1.iva / 100)) + "€");

System.***out***.println(a2.nom + " - Preu: " + a2.preu + "€ - IVA: " + a2.iva + "% - PVP: " + (a2.preu + (a2.preu \* a2.iva / 100)) + "€");

**Exercici C4 – Article**

Aplica l’encapsulament bàsic a la classe Article: Declara tots els seus atributs com private i crea tots

els getters i setters necessaris (un get i un set per atribut).

Programa els setters perquè comproven els valors i els guarden en l'objecte sol si són correctes. En cas contrari mostra un missatge d'error.

public class Article {

private String nom;

private double preu;

private int iva;

private int quantsEnQueden;

public Article(String nom, double preu, int iva, int quantsEnQueden) {

if (nom.equals("")) {

System.***err***.println("ERROR: El nom no pot estar buit");

} else if (preu <= 0) {

System.***err***.println("ERROR: El preu no pot ser menor o igual a cero");

} else if (iva != 21) {

System.***err***.println("ERROR: L'iva ha de ser del 21%");

} else if (quantsEnQueden < 0) {

System.***err***.println("ERROR: El stock no pot ser menor de cero");

} else {

this.nom = nom;

this.preu = preu;

this.iva = iva;

this.quantsEnQueden = quantsEnQueden;

}

}

public String getNom() {

return nom;

}

public double getPreu() {

return preu;

}

public int getIva() {

return iva;

}

public int getQuantsEnQueden() {

return quantsEnQueden;

}

public void setNom(String nom) {

if (nom.equals("")) {

System.***err***.println("ERROR: El nom no pot estar buit");

} else {

this.nom = nom;

}

}

public void setPreu(double preu) {

if (preu <= 0) {

System.***err***.println("ERROR: El preu no pot ser menor o igual a cero");

} else {

this.preu = preu;

}

}

public void setIva(int iva) {

if (iva != 21) {

System.***err***.println("ERROR: L'iva deu ser del 21%");

} else {

this.iva = iva;

}

}

public void setQuantsEnQueden(int quantsEnQueden) {

if (quantsEnQueden < 0) {

System.***err***.println("ERROR: El stock no pot ser menor de cero");

} else {

this.quantsEnQueden = quantsEnQueden;

}

public class ProgramaArticle {

public static void main(String[] args) {

Article a1 = new Article("Camisa de quadres", 20, 21, 5);

Article a2 = new Article("Pantaló de pana", 100, 21, 10);

System.***out***.println(a1.getNom() + " - Preu: " + a1.getPreu() + "€ - IVA: " + a1.getIva() + "% - PVP: " + (a1.getPreu() + (a1.getPreu() \* a1.getIva() / 100)) + "€");

System.***out***.println(a2.getNom() + " - Preu: " + a2.getPreu() + "€ - IVA: " + a2.getIva() + "% - PVP: " + (a2.getPreu() + (a2.getPreu() \* a2.getIva() / 100)) + "€");

a1.setNom("Jaqueta de llana");

a1.setPreu(0);

a1.setPreu(30);

a2.setIva(10);

a2.setQuantsEnQueden(-5);

a2.setQuantsEnQueden(0);

System.***out***.println(a1.getNom() + " - Preu: " + a1.getPreu() + "€ - IVA: " + a1.getIva() + "% - PVP: " + (a1.getPreu() + (a1.getPreu() \* a1.getIva() / 100)) + "€");

System.***out***.println(a2.getNom() + " - Preu: " + a2.getPreu() + "€ - IVA: " + a2.getIva() + "% - PVP: " + (a2.getPreu() + (a2.getPreu() \* a2.getIva() / 100)) + "€");

**Exercici D4 – Article**

Afig a la classe Article mètodes públics amb les següents funcionalitats:

● Mètode per a imprimir la informació de l'article per pantalla.

● Mètode **getPVP** que retorne el preu de venda al públic (PVP) amb iva inclòs.

● Mètode **getPVPDescompte** que retorne el PVP amb un descompte passat com a argument.

● Mètode **vendre** que actualitza els atributs de l'objecte després de vendre una quantitat ‘x’

(si és possible). Retornarà true si ha sigut possible (false en cas contrari).

● Mètode **emmagatzema** que actualitza els atributs de l'objete després d'emmagatzemar una

quantitat ‘x’ (si és possible). Retornarà true si ha sigut possible (false en cas contrari).

public class Article {

private String nom;

private double preu;

private int iva;

private int quantsEnQueden;

public Article(String nom, double preu, int iva, int quantsEnQueden) {

if (nom.equals("")) {

System.***err***.println("ERROR: El nom no pot estar buit");

} else if (preu <= 0) {

System.***err***.println("ERROR: El preu no pot ser menor o igual a cero");

} else if (iva != 21) {

System.***err***.println("ERROR: L'iva ha de ser el 21%");

} else if (quantsEnQueden < 0) {

System.***err***.println("ERROR: El stock no pot ser menor que cero");

} else {

this.nom = nom;

this.preu = preu;

this.iva = iva;

this.quantsEnQueden = quantsEnQueden;

}

}

public String getNom() {

return nom;

}

public double getPreu() {

return preu;

}

public int getIva() {

return iva;

}

public int getQuantsEnQueden() {

return quantsEnQueden;

}

public void setNom(String nom) {

if (nom.equals("")) {

System.***err***.println("ERROR: El nom no pot estar buit");

} else {

this.nom = nom;

}

}

public void setPreu(double preu) {

if (preu <= 0) {

System.***err***.println("ERROR: El preu no pot ser menor o igual a cero");

} else {

this.preu = preu;

}

}

public void setIva(int iva) {

if (iva != 21) {

System.***err***.println("ERROR: L'iva deu ser el 21%");

} else {

this.iva = iva;

}

}

public void setQuantsEnQueden(int quantsEnQueden) {

if (quantsEnQueden < 0) {

System.***err***.println("ERROR: El stock no pot ser menor que cero");

} else {

this.quantsEnQueden = quantsEnQueden;

}

}

// Preu amb iva inclós

public double getPVP() {

return this.preu + (this.preu / 100.0 \* this.iva);

}

// Preu amb descompte

public double getPVPDescompte(double descompte) {

double preu = this.getPVP();

return preu - (preu \* descompte / 100.0);

}

// Mètode vendre. Actualitza les quantitats si se pot vendre.

// Retorna true si la venda és correcta, false en cas contrari

public boolean vendre(int unitats) {

int quants = this.getQuantsEnQueden();

// Si no se pot, retorna false

if (unitats > quants) {

return false;

} // Si se pot, actualitza i retorna true

else {

this.setQuantsEnQueden(quants - unitats);

return true;

}

}

// Mètode vendre. Actualitza les quantitats si se pot emmagatzemar.

// Retorna true si l'emmagatzematge és possible

// False si no és possible (exemple, números negatius)

public boolean emmagatzemar(int unitats) {

int quants = this.getQuantsEnQueden();

if (quants + unitats > 0) {

this.setQuantsEnQueden(quants + unitats);

return true;

} else {

return false;

}

}

// Mostra les dades

public void mostra () {

System.***out***.println("Nom:" + this.getNom() + " IVA:" + this.getIva() + " Preu:" + this.getPreu() + " PVP:" + this.getPVP() + " Unitats:" + this.getQuantsEnQueden());

}

public class ProgramaArticle {

public static void main(String[] args) {

Article a1 = new Article("Camisa de quadres", 20, 21, 5);

Article a2 = new Article("Pantaló de pana", 100, 21, 10);

a1.mostra();

a2.mostra();

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

a1.setNom("Jaqueta de llana");

a1.setPreu(0);

a1.setPreu(30);

a2.setIva(10);

a2.setQuantsEnQueden(-5);

a2.setQuantsEnQueden(0);

a1.mostra();

a2.mostra();

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

if (a1.vendre(1000)) {

System.***out***.println("He venut mil!");

} else {

System.***out***.println("No he venut mil, se cancela la venda per falta de stock");

}

if (a1.vendre(2)) {

System.***out***.println("He venut 2!");

} else {

System.***out***.println("No he venut 2, se cancela la venda por falta de stock");

}

System.***out***.println("Vaig a emmagatzemar 3 més");

if (a1.emmagatzemar(3)) {

System.***out***.println("He emmagatzemat 3!");

} else {

System.***out***.println("No he emmagatzemat 3, no hi ha lloc");

}

a1.mostra();

a2.mostra();

}

**Exercici E4– Article**

A Espanya existeixen tres tipus d'IVA segons la mena de producte:

● L'IVA general (21%): per a la majoria de productes a la venda.

● L'IVA reduït (10%): hostaleria, transport, habitatge, etc.

● L'IVA super reduït (4%): aliments bàsics, llibres, medicaments, etc.

Aquests tres tipus d'IVA no poden variar i a cada article se li aplicarà un dels tres.

Raona quins canvis seria necessari realitzar a la classe Article i implementa'ls.

public class Article {

private String nom;

private double preu;

private int iva;

private int quantsEnQueden;

public static final int ***IVAGENERAL*** = 21;

public static final int ***IVAREDUIT*** = 10;

public static final int ***IVASUPERREDUIT*** = 4;

public Article(String nom, double preu, int iva, int quantsEnQueden) {

if (nom.equals("")) {

System.***err***.println("ERROR: El nom no pot estar buit");

} else if (preu <= 0) {

System.***err***.println("ERROR: El preu no pot ser menor o igual a cero");

} else if (iva != 21) {

System.***err***.println("ERROR: L'iva ha de ser el 21%");

} else if (quantsEnQueden < 0) {

System.***err***.println("ERROR: El stock no pot ser menor que cero");

} else {

this.nom = nom;

this.preu = preu;

this.iva = iva;

this.quantsEnQueden = quantsEnQueden;

}

}

public String getNom() {

return nom;

}

public double getPreu() {

return preu;

}

public int getIva() {

return iva;

}

public int getQuantsEnQueden() {

return quantsEnQueden;

}

public void setNom(String nom) {

if (nom.equals("")) {

System.***err***.println("ERROR: El nom no pot estar buit");

} else {

this.nom = nom;

}

}

public void setPreu(double preu) {

if (preu <= 0) {

System.***err***.println("ERROR: El preu no pot ser menor o igual a cero");

} else {

this.preu = preu;

}

}

public void setIva(int iva) {

if (iva != Article.***IVAGENERAL*** && iva != Article.***IVAREDUIT*** && iva != Article.***IVASUPERREDUIT***) {

System.***err***.println("ERROR: IVA no vàlid");

} else {

this.iva = iva;

}

}

public void setQuantsEnQueden(int quantsEnQueden) {

if (quantsEnQueden < 0) {

System.***err***.println("ERROR: El stock no pot ser menor que cero");

} else {

this.quantsEnQueden = quantsEnQueden;

}

}

// Preu amb iva inclós

public double getPVP() {

return this.preu + (this.preu / 100.0 \* this.iva);

}

// Preu amb descompte

public double getPVPDescompte(double descompte) {

double preu = this.getPVP();

return preu - (preu \* descompte / 100.0);

}

// Mètode vendre. Actualitza les quantitats si se pot vendre.

// Retorna true si la venda és correcta, false en cas contrari

public boolean vendre(int unitats) {

int quants = this.getQuantsEnQueden();

// Si no se pot, retorna false

if (unitats > quants) {

return false;

} // Si se pot, actualitza i retorna true

else {

this.setQuantsEnQueden(quants - unitats);

return true;

}

}

// Mètode vendre. Actualitza les quantitats si se pot emmagatzemar.

// Retorna true si l'emmagatzematge és possible

// False si no és possible (exemple, números negatius)

public boolean emmagatzemar(int unitats) {

int quants = this.getQuantsEnQueden();

if (quants + unitats > 0) {

this.setQuantsEnQueden(quants + unitats);

return true;

} else {

return false;

}

}

// Mostra les dades

public void mostra () {

System.***out***.println("Nom:" + this.getNom() + " IVA:" + this.getIva() + " Preu:" + this.getPreu() + " PVP:" + this.getPVP() + " Unitats:" + this.getQuantsEnQueden());

}

public class ProgramaArticle {

public static void main(String[] args) {

Article a1 = new Article("Camisa de quadres", 20, 21, 5);

Article a2 = new Article("Pantaló de pana", 100, 21, 10);

a1.mostra();

a2.mostra();

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

a1.setNom("Jaqueta de llana");

a1.setPreu(0);

a1.setPreu(30);

a2.setIva(10);

a2.setQuantsEnQueden(-5);

a2.setQuantsEnQueden(0);

a1.mostra();

a2.mostra();

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

if (a1.vendre(1000)) {

System.***out***.println("He venut mil!");

} else {

System.***out***.println("No he venut mil, se cancela la venda per falta de stock");

}

if (a1.vendre(2)) {

System.***out***.println("He venut 2!");

} else {

System.***out***.println("No he venut 2, se cancela la venda por falta de stock");

}

System.***out***.println("Vaig a emmagatzemar 3 més");

if (a1.emmagatzemar(3)) {

System.***out***.println("He emmagatzemat 3!");

} else {

System.***out***.println("No he emmagatzemat 3, no hi ha lloc");

}

a1.mostra();

a2.mostra();

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

a1.setIva(5);

a1.setIva(10);

a2.setIva(12);

a2.setIva(4);

a1.mostra();

a2.mostra();

}

**01. GESTIÓ D’UN APARCAMENT**

Es tracta de realitzar la gestió d’un aparcament fent ús de les estructures apreses en aquesta unitat.

Primer crearem la classe Aparcament. Aquesta classe contindrà dos atributs de classe per a controlar el total de places lliures i ocupades. Per altra banda, cada plaça tindrà un número assignat que no es podrà canviar, un estat (ocupada, lliure) i una matricula assignada en cas d’estar ocupada.

Per a l’estat utilitzarem un tipus de dada enumerat.

Pel que fa a les places d’aparcament, hi haurà 20 i les gestionarem mitjançant un array.

Quan arribe un cotxe demanarà accés i ens proporcionarà la matrícula. Si no hi ha places lliures no es permetrà l’accés i se li donarà un missatge d’avis. Si hi ha places lliures se li donarà accés indicant a quina plaça s’ha de dirigir (el criteri és lliure, la més baixa, random, etc.) i s’actualitzarà la plaça com ocupada i els contadors.

Quan un cotxe vullga eixir, es lliberarà la plaça i s’actualitzaran contadors.

En qualsevol moment podrem preguntar per l’estat d’aguna plaça o bé de tot el parking. En aquest cas s’haurà de fer ús del mètode toString.

public class Aparcament {

// Atributs de classe

private static int *placesLliures* = 0;

private static int *placesOcupades* = 0;

// Atributs d'instància

private final int numero;

private Estat estat;

private String matricula;

// Mètode constructor

public Aparcament() {

*placesLliures*++;

this.numero = *placesLliures*;

this.estat = Estat.LLIURE;

this.matricula=null;

}

// Mètodes getter

public Estat mostraEstat() {

return this.estat;

}

public int mostraNumero() {

return this.numero;

}

public static int mostraLliures() {

return Aparcament.*placesLliures*;

}

public static int mostraOcupades() {

return Aparcament.*placesOcupades*;

}

// Mètode per a mostrar l'esta d'una plaça d'aparcament

public String toString() {

String missatge;

if (this.estat==Estat.LLIURE) {

missatge= "La plaça d'aparcament número " + this.numero + " es troba " + this.estat;

} else {

missatge= "La plaça d'aparcament número " + this.numero + " es troba " + this.estat + " pel vehicle " + this.matricula;

}

return missatge;

}

public void processaEntrada (String matricula){

*placesOcupades*++;

*placesLliures*--;

this.estat = Estat.OCUPADA;

this.matricula=matricula;

}

public void processaEixida (){

*placesOcupades*--;

*placesLliures*++;

this.estat = Estat.LLIURE;

this.matricula=null;

}

public enum *Estat* {

***LLIURE***, ***OCUPADA***

}

public class ProgramaAparcament {

final static int ***PLACES***=20;

public static void main (String[] args){

// Definició de variables

int i, opcio=0;

boolean opcioValida;

Aparcament[] places = new Aparcament [***PLACES***];

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

// Inicialització de les places.

for (i = 0; i < ***PLACES***; i++){

places[i] = new Aparcament();

}

// Processar menù d'opcions

do {

opcioValida=false;

System.***out***.println ("\n========= MENÚ APARCAMENT ========");

System.***out***.println ("1.- Entra de vehicle");

System.***out***.println ("2.- Eixida de vehicle");

System.***out***.println ("3.- Estat d'una plaça");

System.***out***.println ("4.- Estat del parking");

System.***out***.println ("0.- Finalitzar");

// Validar opció triada

do {

System.***out***.print("Introdueix una opció: ");

if(menu.hasNextInt()) {

opcio=menu.nextInt();

if ((opcio >=0) && (opcio<=4)){

opcioValida=true;

} else {

System.***out***.println("L'opció no és correcta.");

}

}

menu.nextLine();

} while (!opcioValida);

// Executar opció

switch (opcio) {

case 1:

*entradaVehicle*(places);

break;

case 2:

*eixidaVehicle*(places);

break;

case 3:

*informacioPlaça*(places);

break;

case 4:

*informacioAparcament*(places);

break;

}

} while (opcio!=0);

System.***out***.println("Adeu!");

}

// Procediment que gestiona l'entrada d'un vehicle

public static void entradaVehicle(Aparcament[] places){

int i;

boolean trobat=false;

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

String matricula;

// Si l'aparcament està complet donem un avis

if (Aparcament.mostraLliures()<=0){

System.***out***.println ("Ho sentim molt, l'aparcament està complet.");

// Si hi ha places donem la primera lliure

} else {

System.***out***.println ("Introdueix la matrícula: ");

matricula = entrada.nextLine();

i=0;

while (i<***PLACES*** && !trobat){

if (places[i].mostraEstat()==Estat.LLIURE){

trobat=true;

places[i].processaEntrada(matricula);

System.***out***.println ("Vaja a la plaça número " + places[i].mostraNumero());

}

i++;

}

}

}

// Procediment que gestiona l'eixida d'un vehicle

public static void eixidaVehicle(Aparcament[] places){

int posicio;

boolean trobat=false;

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

// Validar el número de la plaça

do {

System.***out***.print("Introdueix el número de la plaça: ");

if(entrada.hasNextInt()) {

posicio=entrada.nextInt();

if ((posicio >=1) && (posicio<=***PLACES***)&& (places[posicio-1].mostraEstat() == Estat.OCUPADA)){

trobat=true;

places[posicio-1].processaEixida();

System.***out***.println ("Ja pot eixir de l'aparcament. Moltes gràcies.");

}

}

entrada.nextLine();

if (!trobat) {

System.***out***.println("La plaça indicada no és correcta.");

}

} while (!trobat);

}

// Procediment que mostra l'estat d'una plaça d'aparcament

public static void informacioPlaça(Aparcament [] places){

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

int posicio=0;

boolean opcioValida=false;

do {

System.***out***.print("\nIntrodueix el número de la plaça a consultar: ");

if(entrada.hasNextInt()) {

posicio=entrada.nextInt();

if ((posicio >=1) && (posicio<=***PLACES***)){

opcioValida=true;

}

}

entrada.nextLine();

if (!opcioValida){

System.***out***.println("El número de plaça triat no és correcte.");

}

} while (!opcioValida);

System.***out***.println (places[posicio-1]);

}

// Procediment que mostra l'estat de l'aparcament complet

public static void informacioAparcament(Aparcament [] places){

int i;

System.***out***.println ("\nESTAT ACTUAL DE L'APARCAMENT");

for (i = 0; i < ***PLACES***; i++){

System.***out***.println (places[i]);

}