

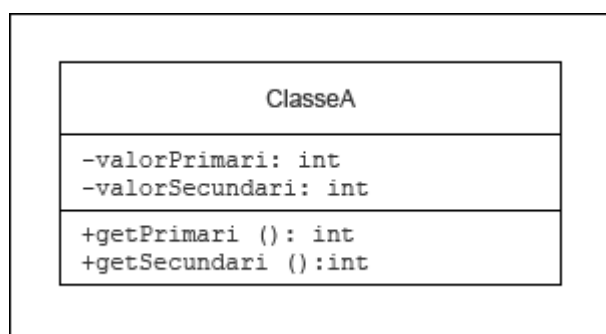
EXERCICIS DE PROGRAMACIÓ ORIENTADA A OBJECTES

Instanciar classes

L'objectiu d'aquesta activitat és veure com instanciar classes fetes per vosaltres.

- 1- Creeu una classe anomenada **ClasseA** d'acord a la següent especificació. Els dos mètodes **get...** serveixen per mostrar els valors emmagatzemats a cada atribut.

Figura



Afegiu tres constructors:

public ClasseA(), que assigna els valors 5 i 10 als dos atributs, respectivament.

public ClasseA(int vp), que assigna “vp” a “valorPrimari” i el valor 10 a “valorSecundari”.

public ClasseA(int vp, int vs), que assigna “vp” i “vs” als dos atributs, respectivament.

Proveu que heu fet la classe correctament afegint el següent mètode main al codi i executant la classe.

```
public static final void main (String[] args)
{
    ClasseA a = new ClasseA();
    ClasseA b = new ClasseA(20);
    ClasseA c = new ClasseA(20, 40);
    System.out.println("L'objecte _a:_ conté: " + a.getPrimari() + ", " + a.getSecundari());
    System.out.println("L'objecte _b:_ conté: " + b.getPrimari() + ", " + b.getSecundari());
    System.out.println("L'objecte _c:_ conté: " + c.getPrimari() + ", " + c.getSecundari());
}
```

```
package javaapplication10;
public class JavaApplication10 {
    public static void main(String[] args) {
        ClasseA a = new ClasseA();
        ClasseA b = new ClasseA(20);
        ClasseA c = new ClasseA(20, 40);
        System.out.println("L'objecte _a:_ conté: " + a.getPrimari() + ", " + a.getSecundari());
        System.out.println("L'objecte _b:_ conté: " + b.getPrimari() + ", " + b.getSecundari());
        System.out.println("L'objecte _c:_ conté: " + c.getPrimari() + ", " + c.getSecundari());
    }
}

package javaapplication10;
public class ClasseA {
    private int valorPrimari;
    private int valorSecundari;

    public ClasseA(){
        this.valorPrimari = 5;
        this.valorSecundari = 10;
    }

    public ClasseA(int vp){
        this.valorPrimari = vp;
        this.valorSecundari = 10;
    }

    public ClasseA(int vp, int vs){
        this.valorPrimari = vp;
        this.valorSecundari = vs;
    }

    public int getPrimari() {
        return valorPrimari;
    }

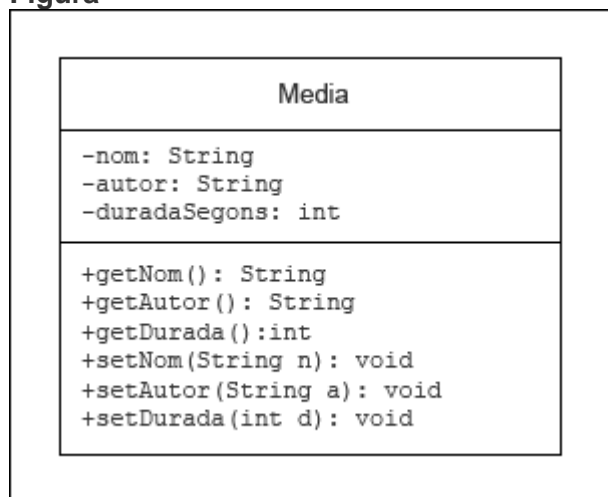
    public int getSecundari() {
        return valorSecundari;
    }
}
```

Codificació de classes

L'objectiu d'aquesta activitat és practicar la codificació d'una classe senzilla.

- 2- Donada la següent especificació, genereu el codi font d'aquesta classe. Els mètodes **set...** i **get...** serveixen per accedir i modificar, respectivament, els valors emmagatzemats als atributs. Podeu generar els constructors que vulgueu.

Figura



Un cop fet, comproveu que funciona amb un programa de prova (un mètode **main**), instanciant la classe i accedint i assignant valors als seus atributs.

```
package exercici2;
```

```
import java.util.Scanner;
```

```
public class Exercici2 {
```

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

```
        Media media1 = new Media("Joan", "Pere", 3);
```

```
        Media media2 = new Media("Pau", "Eloi", 8);
```

```
        Scanner sc = new Scanner(System.in);
```

```
        System.out.println(media1.getNom()+media1.getAutor()+media1.getDuradaSegons());
```

```
        System.out.println(media2.getNom()+media2.getAutor()+media2.getDuradaSegons());
```

```
media1.setAutor("Aleix");  
  
media2.setNom("Biel");  
  
System.out.println(media1.getNom()+media1.getAutor()+media1.getDuradaSegons());  
System.out.println(media2.getNom()+media2.getAutor()+media2.getDuradaSegons());  
  
}
```

```
}
```

```
package exercici2;
```

```
public class Media {
```

```
    private String nom;
```

```
    private String autor;
```

```
    private int duradaSegons;
```

```
    public String getNom() {
```

```
        return nom;
```

```
    }
```

```
    public String getAutor() {
```

```
        return autor;
```

```
    }
```

```
    public int getDuradaSegons() {
```

```
        return duradaSegons;
```

```
}
```

```
public void setNom(String nom) {
```

```
    this.nom = nom;
```

```
}
```

```
public void setAutor(String autor) {
```

```
    this.autor = autor;
```

```
}
```

```
public void setDuradaSegons(int duradaSegons) {
```

```
    this.duradaSegons = duradaSegons;
```

```
}
```

```
public Media(String nom, String autor, int duradaSegons) {
```

```
    this.nom = nom;
```

```
    this.autor = autor;
```

```
    this.duradaSegons = duradaSegons;
```

```
}
```

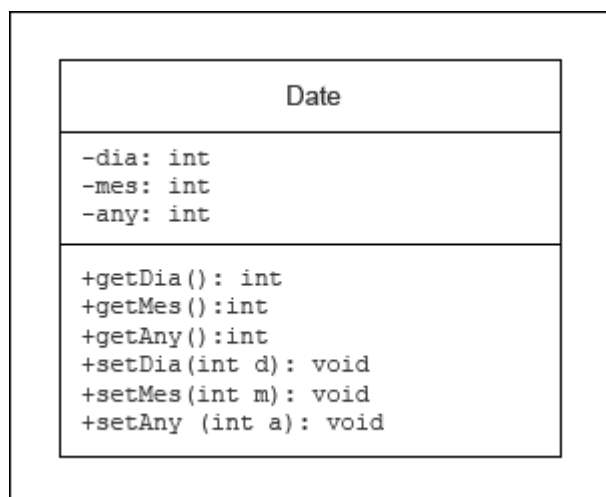
```
}
```

Modificació d'atributs

L'objectiu d'aquesta activitat és practicar la codificació d'una classe senzilla i entendre la utilitat dels mètodes `set...`

- 3- Donada la següent especificació, genereu el codi font d'aquesta classe. Els mètodes **set...** i **get...** serveixen per accedir i per modificar, respectivament, els valors emmagatzemats en els atributs. Podeu generar els constructors que vulgueu. Ara bé, per a aquest exercici, cal que els mètodes **set...** controlin que el valor que es vol assignar és correcte, i si no és el cas, no fer res (deixar el valor que hi havia inicialment). Mai no hi pot haver una data amb un dia-mes-any que no sigui coherent. S'han de controlar casos exagerats, com ara intentar assignar el valor de mes 25, o el dia 50, o com ara intentar desar el dia 31 quan el mes actual és el febrer. Recordeu també els anys de traspàs.

Figura



Un cop fet, comproveu que funciona amb un programa de prova (un mètode **main**), instanciant la classe i accedint i assignant valors als seus atributs.

```
package exercici3;
```

```
public class Exercici3 {
```

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

```
        Date date = new Date(4, 5, 2004);
```

```
        Date date2 = new Date(5, 10, 9);
```

```
        Date date3 = new Date(1, 1, 2024);
```

```
        Date date4 = new Date(30, 2, 2020);
```

```
        Date date5 = new Date(31, 4, 2020);
```

```
        if (date.isValid()) {
```

```
            System.out.println(date.getDia() + "/" + date.getMes() + "/" + date.getAny());
```

```
        } else {
```

```
            System.out.println("Data incorrecta");
```

```
        }
```

```
        if (date2.isValid()) {
```

```
            System.out.println(date2.getDia() + "/" + date2.getMes() + "/" + date2.getAny());
```

```
        } else {
```

```
            System.out.println("Data incorrecta");
```

```
        }
```

```
        if (date3.isValid()) {
```

```
System.out.println(date3.getDia() + "/" + date3.getMes() + "/" + date3.getAny());

} else {

    System.out.println("Data incorrecta");

}

if (date4.isValid()) {

    System.out.println(date4.getDia() + "/" + date4.getMes() + "/" + date4.getAny());

} else {

    System.out.println("Data incorrecta");

}

if (date5.isValid()) {

    System.out.println(date5.getDia() + "/" + date5.getMes() + "/" + date5.getAny());

} else {

    System.out.println("Data incorrecta");

}

date5.setDia(1);

date5.setMes(1);

date5.setAny(1);

if (date5.isValid()) {

    System.out.println(date5.getDia() + "/" + date5.getMes() + "/" + date5.getAny());

} else {

    System.out.println("Data incorrecta");
```



```
}  
  
}  
  
}  
  
package exercici3;  
  
public class Date {  
    private int dia;  
    private int mes;  
    private int any;  
  
    public int getDia() {  
        return dia;  
    }  
  
    public int getMes() {  
        return mes;  
    }  
  
    public int getAny() {  
        return any;  
    }  
  
    public void setDia(int dia) {  
        this.dia = dia;  
    }  
  
    public void setMes(int mes) {  
        this.mes = mes;  
    }  
  
    public void setAny(int any) {  
        this.any = any;  
    }  
  
    public Date(int dia, int mes, int any) {  
        this.dia = dia;  
        this.mes = mes;  
        this.any = any;  
    }  
  
    public boolean isValid() {  
        if (mes < 1 || mes > 12) {  
            return false;  
        }  
    }  
}
```

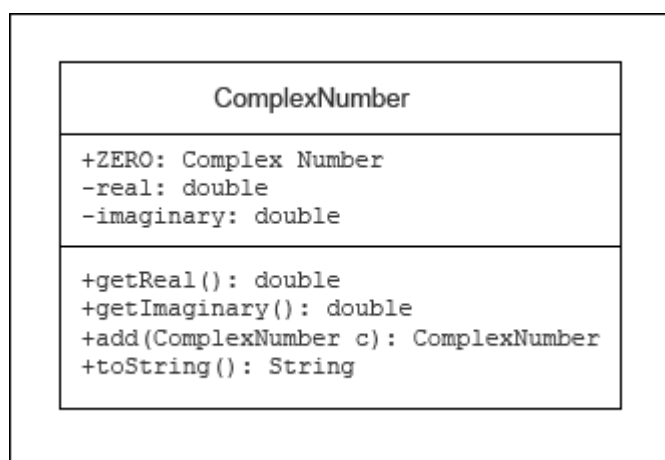
```
if (dia < 1 || dia > diesDelMes(mes)) {  
    return false;  
}  
  
return true;  
}  
  
private int diesDelMes(int month) {  
    switch (month) {  
        case 1: case 3: case 5: case 7: case 8: case 10: case 12:  
            return 31;  
        case 4: case 6: case 9: case 11:  
            return 30;  
        case 2:  
            return (anyDeTraspas()) ? 29 : 28;  
        default:  
            return -1;  
    }  
}  
  
private boolean anyDeTraspas() {  
    return (any % 4 == 0 && any % 100 != 0) || (any % 400 == 0);  
}  
}
```

Classes que usen altres classes

L'objectiu d'aquesta activitat és practicar la codificació d'una classe que, al mateix temps, usa classes al seu codi.

- 4- Donada la següent especificació, genereu el codi font d'aquesta classe. Podeu afegir-hi els constructors que vulgueu. La constant ZERO es refereix a una instància de **ComplexNumber** amb els valors "real" i "imaginary" ambdós a 0.

Figura



La descripció dels mètodes és:

get..., serveixen per accedir als valors dels dos atributs.

add(...), retorna un nou objecte ComplexNumber. El valor de cadascun dels seus dos atributs nous és la suma del mateix atribut al paràmetre "c" i al propi objecte que executa add (o sigui, "suma"). Per exemple, el valor "real" del nou objecte és la suma de l'atribut "real" de "c" i de l'objecte que executa add.

toString(), retorna una cadena de text on es mostren els valors dels atributs de l'objecte (per exemple, separats per una coma).

Un cop fet, comproveu que funciona amb un programa de prova (un mètode **main**), instanciant la classe diverses vegades i invocant els diferents mètodes. Ajudeu-vos del mètode toString per mostrar les dades per pantalla.

Aquest programa ja els ho dono fet i l'heu de còpia i enganxar i provar-lo.

Codi: Exercicis	Data del Format: 02-09-2013	Pàgina 11 de 15
C/ Ferrer i Busquets, 17	25230-Mollerussa	E-mail: lasallemollerussa@lasalle.cat
Tel. 973 600 270	Fax 973 710 599	http://www.mollerussa.lasalle.cat

```
public class ComplexNumber
{
    public static final ComplexNumber ZERO = new ComplexNumber(0.0, 0.0);

    private double real;
    private double imaginary;

    public ComplexNumber(double real, double i) {
        this.real = real;
        this.imaginary = i;
    }

    public Double getReal() {
        return this.real;
    }

    public Double getImaginary()
    {
        return this.imaginary;
    }

    public ComplexNumber add(ComplexNumber c)
    {
        ComplexNumber resultat = new ComplexNumber(this.real + c.getReal(), this.imaginary+c.getImaginary());

        return resultat;
    }

    public String toString( ) {
        String resultat = this.real + " ";
        if (this.imaginary >= 0.0) {
            resultat += "+" + this.imaginary + "i";
        } else {
            resultat += this.imaginary + "i";
        }
        return resultat;
    }
}
```

```
public class ProvaComplexNumber
{
    public static void main (String[ ] args)
    {
        ComplexNumber c1 = new ComplexNumber(1.0,2.0);
        ComplexNumber c2 = new ComplexNumber(2.0,-3.0);

        ComplexNumber zero = ComplexNumber.ZERO;

        System.out.println("Hem creat el Nombre Complex c1:");
        System.out.println("\t Part real = " + c1.getReal());
        System.out.println("\t Part imaginària = " + c1.getImaginary());

        System.out.println("\nHem creat el Nombre Complex c2:");
        System.out.println("\t Part real = " + c2.getReal());
        System.out.println("\t Part imaginària = " + c2.getImaginary());

        System.out.println("\nOperació de suma (c1 + ZERO):");
        ComplexNumber c3 = c1.add(ZERO);
        System.out.println("\t Part real = " + c3.getReal());
        System.out.println("\t Part imaginària = " + c3.getImaginary());

        System.out.println("\nOperació de suma (c1 + c2):");
        c3 = c1.add(c2);
        System.out.println("\t Part real = " + c3.getReal());
        System.out.println("\t Part imaginària = " + c3.getImaginary());

        System.out.println("\nDarrera suma duta a terme: ");
        System.out.print("(" + c1 + ") + (" + c2 + ") = (" + c3 + ")");
    }
}
```

Després d'això heu d'ampliar aquest programa amb la resta i multiplicació de 2 número imaginaris de la mateixa forma que la suma.

Teoria de número imaginaris:

Clase Numero Complejo - Suma, Resta, Producto



El término número complejo describe la suma de un número real y un número imaginario (que es un múltiplo real de la unidad imaginaria, que se indica con la letra i).

En matemáticas, los números constituyen un cuerpo y, en general, se consideran como puntos del plano: el plano complejo. La propiedad más importante que caracteriza a los números complejos es el teorema fundamental del álgebra, que afirma que cualquier ecuación algebraica de grado n tiene exactamente n soluciones complejas.

Un número complejo se representa en forma binomial como:

$$z = a + bi$$

donde:

"a" es la parte real

"b" es la parte imaginaria

Suma.

La suma $z + w$ de los números complejos $z = a + bi$, $w = c + di$, es el número complejo

$$z + w = (a + bi) + (c + di) = (a + c) + (b + d)i$$

ejemplo:

$$(4 + 3i) + (5 + 9i) = (4 + 5) + (3 + 9)i = 9 + 12i$$

Codi: Exercicis	Data del Format: 02-09-2013	Pàgina 14 de 15
C/ Ferrer i Busquets, 17	25230-Mollerussa	E-mail: lasallemollerussa@lasalle.cat
Tel. 973 600 270	Fax 973 710 599	http://www.mollerussa.lasalle.cat

Resta.

La resta $z - w$ de los números complejos $z = a + b i$, $w = c + d i$, es la suma de z y del inverso aditivo de w

$$z - w = z + (-w) = (a + b i) + (-c - d i) = (a - c) + (b - d) i$$

ejemplo:

$$(9 - 5i) - (4 + 7i) = (9 - 4) + (+5 - 7)i = (5, -2)$$

Producto.

$$(a + b i) (c + d i) = (ac - bd) + (ad + bc)i$$

ejemplo

$$(2 + 3i) \text{ mult } (4 + 7i) = (2 \cdot 4 - 3 \cdot 7) + (2 \cdot 7 + 3 \cdot 4)i$$

$$= (8 - 21) + (14 + 12)i$$

$$= (8 - 21) + (14 + 12)i$$

$$= (-13, 26)$$