

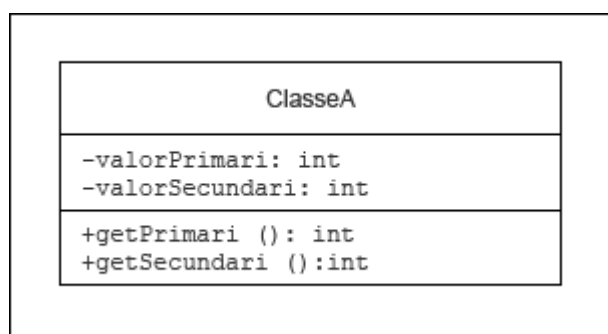
## EXERCICIS DE PROGRAMACIÓ ORIENTADA A OBJECTES

### 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 exercisi1;
```

```
public class Exercisi1 {
```

```
    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.getValorPrimari() + ", " +
a.getValorSecundari());
        System.out.println("L'objecte _b:_ conté: " + b.getValorPrimari() + ", " +
b.getValorSecundari());
        System.out.println("L'objecte _c:_ conté: " + c.getValorPrimari() + ", " +
c.getValorSecundari());

    }
```

```
}
```

```
package exercisi1;
```

```
public class ClasseA {
```

```
    private int valorPrimari;
    private int valorSecundari;
```

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

```
    public ClasseA(int valorPrimari) {
        this.valorPrimari = valorPrimari;
    }
```

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

```

public int getValorPrimari() {
    return valorPrimari;
}

public int getValorSecundari() {
    return valorSecundari;
}
}

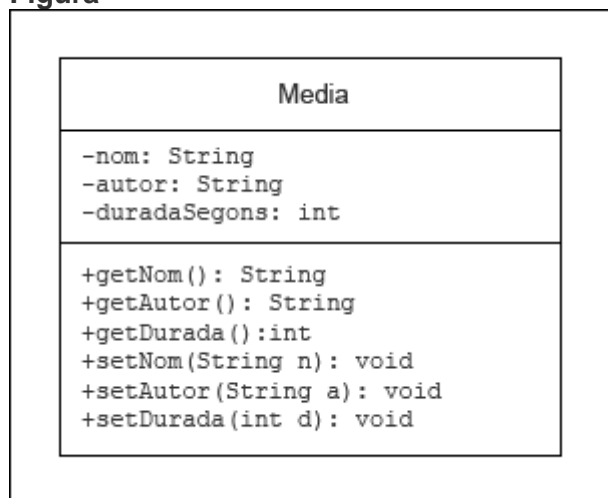
```

## 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 exercisi2;
```

```
public class Exercisi2 {
```

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

```
        Media Johnny_sins_y_la_secretaria_nueva = new Media();
```

```
        Johnny_sins_y_la_secretaria_nueva.setNom("Mariana");
```

```
        Johnny_sins_y_la_secretaria_nueva.setAutor("abel");
```

```
        Johnny_sins_y_la_secretaria_nueva.setDuradaSegons(100);
```

```
        System.out.println("El nom es : "+ Johnny_sins_y_la_secretaria_nueva.getNom() );
```

```
        System.out.println("El autor es : "+ Johnny_sins_y_la_secretaria_nueva.getAutor() );
```

```
        System.out.println("La durada es : "+  
        Johnny_sins_y_la_secretaria_nueva.getDuradaSegons() );
```

```
    }
```

```
}
```

```
package exercisi2;
```

```
public class Media {
```

```
private String Nom;
```

```
private String Autor;
```

```
private int DuradaSegons;
```

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

```
    return Nom;
```

```
}
```

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

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

```
}
```

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

```
    return Autor;
```

```
}
```

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

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

```
}
```

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

```
    return DuradaSegons;
```

```
}
```

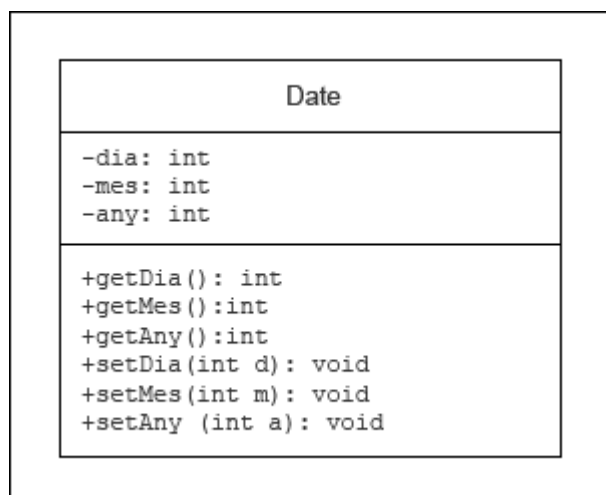
```
public void setDuradaSegons(int duradaSegons) {  
  
    this.DuradaSegons = duradaSegons;  
  
}  
  
}
```

## 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.

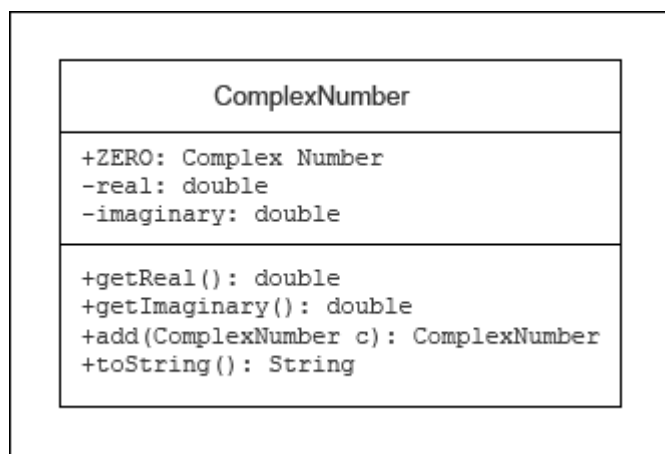
No el se fer

## 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.

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

    private double real;
    private double imaginary;

    public ComplexNumber(double r, double i) {
        this.real = r;
        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 res = c1.add(ComplexNumber.ZERO);
        System.out.println("\t Part real = " + res.getReal());
        System.out.println("\t Part imaginària = " + res.getImaginary());

        System.out.println("\nOperació de suma (C1 + C2):");
        res = c1.add(c2);
        System.out.println("\t Part real = " + res.getReal());
        System.out.println("\t Part imaginària = " + res.getImaginary());

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

```

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

package exercisi4;

```
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 res = c1.add(ComplexNumber.ZERO);
        System.out.println("\t Part real = " + res.getReal());
        System.out.println("\t Part imaginària = " + res.getImaginary());

        System.out.println("\nOperació de suma (C1 + C2):");
        res = c1.add(c2);
        System.out.println("\t Part real = " + res.getReal());
        System.out.println("\t Part imaginària = " + res.getImaginary());

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

        ComplexNumber res1 = c1.rest(ComplexNumber.ZERO);
        res1.rest(c2);

        System.out.println("\nDarrera resta duta a terme: ");
        System.out.print("(" + c1 + ") - (" + c2 + ") = (" + res1 + ")");
        System.out.println("\nDarrera multiplicacio duta a terme: ");

        ComplexNumber res2 = c1.multiply(ComplexNumber.ZERO);
        res2.multiply(c2);

        System.out.print("(" + c1 + ") * (" + c2 + ") = (" + res2 + ")");
    }
}
```

package exercisi4;

public class ComplexNumber {

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

    private double real;

    private double imaginary;

    public ComplexNumber(double r, double i) {

        this.real = r;

        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 ComplexNumber rest(ComplexNumber c) {

        ComplexNumber resultat = new ComplexNumber(this.real - c.getReal(), this.imaginary - c.getImaginary());

        return resultat;

    }

    public ComplexNumber multiply(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;
}
```

Teoria de número imaginari:

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  $z = a + b i$ ,  $w = c + d i$ , es el número complejo

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

ejemplo:

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

### 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:

$$(9 - 5i) - (4 + 7i) = (9 - 4) + (-5 - 7)i = (5, -12)$$

### Producto.

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

ejemplo

$$(2 + 3i) \cdot (4 + 7i) = (2 \cdot 4 - 3 \cdot 7) + (2 \cdot 7 + 3 \cdot 4)i$$

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

$$= (-13) + (26)i$$

$$= (-13, 26)$$