

Familia Profesional <b>Informática y Telecomunicaciones</b>		Nombre del Ciclo Formativo <b>Título de Técnico Superior en Desarrollo de Aplicaciones Web</b>		
Centro Educativo <b>IES Campanillas</b> (sede PTA)		Módulo Profesional <b>Programación</b> Código: 0485 N.º de créditos ECTS: 14	Profesor <b>Juan Antonio Jiménez Morales</b>	
Curso lectivo: <b>2019/2020</b>	Grupo: <b>1º DAW</b>	Trimestre: <b>Segº – Ejercicios ampliación POO</b>	Fecha: <b>27/03/2020</b>	Pág. <b>1/3</b>

## EJERCICIOS

1. Cree una clase `Complejo` que permita trabajar con números complejos. Cada número complejo dispondrá de dos atributos de tipo `double`: la parte real y la parte imaginaria. Incluya los siguientes métodos:
  - a) constructor por defecto: creará un número complejo con partes real e imaginaria iguales a 0 (cero)
  - b) constructor parametrizado: creará un número complejo con partes real e imaginaria según los valores indicados como argumento
  - c) `parteReal`: devolverá la parte real del número complejo
  - d) `parteImaginaria`: devolverá la parte imaginaria del número complejo
  - e) `suma`: recibirá como argumento un número complejo, y devolverá como resultado la suma compleja del número complejo que recibe el mensaje y del número complejo que se le ha pasado como argumento
  - f) `resta`: recibirá como argumento un número complejo, y devolverá como resultado la resta compleja del número complejo que recibe el mensaje y del número complejo que se le ha pasado como argumento
  - g) `multiplicacion`:
    - i. recibirá como argumento un número complejo, y devolverá como resultado la multiplicación compleja del número complejo que recibe el mensaje y del número complejo que se le ha pasado como argumento
    - ii. recibirá como argumento un número real, y devolverá como resultado la multiplicación del número complejo que recibe el mensaje y el número real que se le ha pasado como argumento
  - h) `división`: recibirá como argumento un número complejo, y devolverá como resultado la división compleja del número complejo que recibe el mensaje y del número complejo que se le ha pasado como argumento
  - i) `toString`: devolverá una cadena de caracteres, representando al número complejo de la siguiente forma: (<parte\_real>,<parte\_imaginaria>). Es decir, la cadena de caracteres contendrá, entre paréntesis y separados por una coma, los valores real e imaginario del número complejo que reciba el mensaje.
  - j) `equals`: se le pasará como argumento un número complejo, y devolverá `true` si el número complejo que recibe el mensaje es igual al número complejo que se ha pasado como argumento, y `false` en caso contrario.

Debe tener en cuenta lo siguiente:

- i. Al sumar dos números complejos, (a,b) y (c,d), el resultado que se obtiene es: (a+c,b+d)
- ii. Al restar dos números complejos, (a,b) y (c,d), el resultado que se obtiene es: (a-c,b-d)
- iii. Al multiplicar dos números complejos, (a,b) y (c,d), el resultado que se obtiene es: (a\*c-b\*d,a\*d-b\*c)
- iv. Al multiplicar un número complejo (a,b) por un número real, c, el resultado que se obtiene es: (a\*c.b\*c)
- v. Al dividir dos números complejos, (a,b) y (c,d), el resultado que se obtiene es:  

$$\left( \frac{a*c + b*d}{c^2 + d^2}, \frac{b*c - a*d}{c^2 + d^2} \right)$$
- vi. Dos números complejos son iguales si las partes reales son iguales entre sí y las partes imaginarias también son iguales entre sí.

Una vez haya creado la clase, cree un programa que le permita mostrar el funcionamiento de dicha clase.  
 Sugerencia avanzada: crear una calculadora.

Familia Profesional <b>Informática y Telecomunicaciones</b>		Nombre del Ciclo Formativo <b>Título de Técnico Superior en Desarrollo de Aplicaciones Web</b>		
Centro Educativo <b>IES Campanillas</b> (sede PTA)		Módulo Profesional <b>Programación</b> Código: <b>0485</b> N.º de créditos ECTS: <b>14</b>	Profesor <b>Juan Antonio Jiménez Morales</b>	
Curso lectivo: <b>2019/2020</b>	Grupo: <b>1º DAW</b>	Trimestre: <b>Segº – Ejercicios ampliación POO</b>	Fecha: <b>27/03/2020</b>	Pág. <b>2/3</b>

2. Cree una clase `Racional` que permita trabajar con números racionales (fracciones). Cada número racional dispondrá de dos atributos de tipo entero: el numerador y el denominador. Incluya los siguientes métodos:
- constructor por defecto: creará un número racional con numerador igual a 0 y denominador igual a 1.
  - constructor parametrizado: creará un número racional con numerador y denominador según los valores indicados como argumento
  - `numerador`: devolverá el valor del numerador de la fracción
  - `denominador`: devolverá el valor del denominador de la fracción
  - `suma`: recibirá como argumento un número racional, y devolverá como resultado la suma fraccionaria del número racional que recibe el mensaje y del número racional que se le ha pasado como argumento
  - `resta`: recibirá como argumento un número racional, y devolverá como resultado la resta fraccionaria del número racional que recibe el mensaje y del número racional que se le ha pasado como argumento
  - `multiplicacion`:
    - recibirá como argumento un número racional, y devolverá como resultado la multiplicación fraccionaria del número racional que recibe el mensaje y del número racional que se le ha pasado como argumento
    - recibirá como argumento un número real, y devolverá como resultado la multiplicación del número racional que recibe el mensaje y el número real que se le ha pasado como argumento
  - `división`: recibirá como argumento un número racional, y devolverá como resultado la división fraccionaria del número racional que recibe el mensaje y del número racional que se le ha pasado como argumento
  - `irreducible`: devolverá el número racional equivalente e irreducible al que recibe el mensaje
  - `toString`: devolverá una cadena de caracteres, representando al número racional de la siguiente forma: `<numerador>/<denominador>`. Es decir, la cadena de caracteres contendrá, separados por una barra “del 7”, los valores del numerador y denominador del número racional que reciba el mensaje.
  - `equals`: se le pasará como argumento un número racional, y devolverá `true` si el número racional que recibe el mensaje es igual al número racional que se ha pasado como argumento, y `false` en caso contrario.

Debe tener en cuenta lo siguiente:

- Al sumar dos números racionales,  $a/b$  y  $c/d$ , el resultado que se obtiene es:  $(a*d+b*c)/(b*d)$
- Al restar dos números racionales,  $a/b$  y  $c/d$ , el resultado que se obtiene es:  $(a*d-b*c)/(b*d)$
- Al multiplicar dos números racionales,  $a/b$  y  $c/d$ , el resultado que se obtiene es:  $(a*c)/(b*d)$
- Al multiplicar un número racional  $a/b$  por un número real,  $c$ , el resultado que se obtiene es:  $(a*c)/b$
- Al dividir dos números racionales,  $a/b$  y  $c/d$ , el resultado que se obtiene es:  $(a*c)/(d*b)$
- Dos números racionales son iguales si son fracciones equivalentes. Es decir, si la fracción irreducible que se obtiene de cada uno de ellos es la misma. Dos fracciones irreducibles son iguales si son iguales los numeradores y son iguales los denominadores
- Una fracción es irreducible si numerador y denominador no tienen divisores comunes. En caso de que tuviera divisores comunes, se pueden dividir numerador y denominador por dichos divisores comunes, obteniéndose la fracción irreducible.
- Tras la realización de un cálculo, es muy recomendable optar por devolver la fracción irreducible equivalente.

Una vez haya creado la clase, cree un programa que le permita mostrar el funcionamiento de dicha clase. Sugerencia avanzada: crear una calculadora.

Familia Profesional <b>Informática y Telecomunicaciones</b>		Nombre del Ciclo Formativo <b>Título de Técnico Superior en Desarrollo de Aplicaciones Web</b>		
Centro Educativo <b>IES Campanillas</b> (sede PTA)		Módulo Profesional <b>Programación</b> Código: <b>0485</b> N.º de créditos ECTS: <b>14</b>	Profesor <b>Juan Antonio Jiménez Morales</b>	
Curso lectivo: <b>2019/2020</b>	Grupo: <b>1º DAW</b>	Trimestre: <b>Segº – Ejercicios ampliación POO</b>	Fecha: <b>27/03/2020</b>	Pág. <b>3/3</b>

3. Desarrolla una clase `Cafetera` con atributos:

- a) `capacidadMaxima` (la cantidad máxima de café que puede contener la cafetera)
- b) `cantidadActual` (la cantidad actual de café que hay en la cafetera).

Implementa, al menos, los siguientes métodos:

- a) Constructor predeterminado: establece la capacidad máxima en 1000 (c.c.) y la actual en cero (cafetera vacía).
- b) Constructor con la capacidad máxima de la cafetera: inicializa la cantidad actual de café igual a la capacidad máxima.
- c) Constructor con la capacidad máxima y la cantidad actual: Si la cantidad actual es mayor que la capacidad máxima de la cafetera, la ajustará al máximo.
- d) `maximaCap`: devuelve la capacidad máxima de la cafetera.
- e) `cantidad`: devuelve la cantidad actual de café que hay en la cafetera
- f) `llenarCafetera`: pues eso, hace que la cantidad actual sea igual a la capacidad.
- g) `servirTaza`: simula la acción de servir una taza con la capacidad que se le pase como argumento. Si la cantidad actual de café "no alcanza" para llenar la taza, se sirve lo que quede.
- h) `vaciarCafetera`: pone la cantidad de café actual en cero.
- i) `agregarCafe`: añade a la cafetera la cantidad de café indicada como argumento, teniendo en cuenta que "se derramará" lo que no quepa.

Una vez haya implementado la clase, haga un programa de ejemplo para ilustrar el funcionamiento de esta clase.