|  |  |
| --- | --- |
| TP04 Composición de clases  Dr. Carlos Hugo García Capulín | Descripción breve  Universidad de Guanajuato  JOSE ADRIAN RODRIGUEZ GONZALEZ  Programación orientada objetos 2  Fecha de entrega: 28/03/2025 23:59 |

Contenido

[Introducción: 1](#_Toc194068234)

[Desarrollo. 1](#_Toc194068235)

[Resultados: 10](#_Toc194068236)

[Conclusión: 10](#_Toc194068237)

# Introducción:

En esta ocasión se pone en carga la composición de clases y el como las clases, y si bien en C++ se pueden sobrecargar los operadores, en Java, no se puede hacer esto. Para el manejo de la composición de clases, lo realizaremos con la multiplicación de matrices complejas en Java.

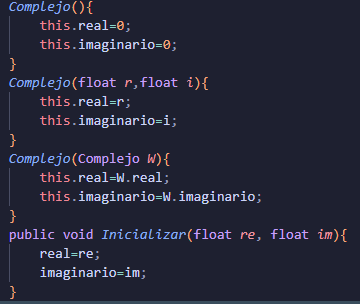
# Desarrollo.

Para esta ocasión, se pide que la matriz, capture el número complejo, que se imprima en pantalla la matriz, y que se pueda realizar el producto matricial de números complejos, y para ello, vamos a emplear una clase previamente desarrollada llamada Complejo

Texto

Descripción generada automáticamente La clase complejo utiliza el paquete de Scanner y dentro de la clase se tienen dos atributos que son privadas, el número real e imaginario. Esta clase demuestra la sobrecarga de funciones, es decir, que cuando le pasas una distinta cantidad de parámetros a una función, va a interpretarla como una función distinta, y este caso se demuestra en el constructor de clase, que se crearon 3 *Complejo. En la ilustración 3* se muestra que el primer constructor, no se le pasa ningún valor, ya que a real e imaginario se los da como 0. Después tenemos al constructor que construye el número complejo, y puedes pasarle ya sea los dos números en flotante, o le puedes pasar otro objeto de la clase complejo y este va a *copiar*, los atributos que actualmente tenga.

Ilustración 1 Clase Complejo

**Después, existe una función denominada inicializar se encargar de hacer el proceso similar a los constructores, esta función será utilizada en la clase donde esté la matriz compleja.

Esta clase posee las 4 operaciones básicas de un número complejo, además de que tenemos una función para poder capturar los datos y mostrar en pantalla.

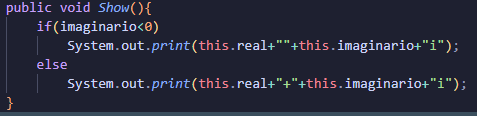
**La función show, solamente pregunta si el valor es negativo el del complejo, solamente para tener la consideración del signo negativo o positivo.

Ilustración 2 Inicializadores

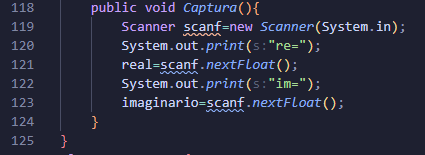
**

Ilustración 3 función captura

Ilustración 4 Función show

Y captura utiliza el objeto scanner para estar capturando la parte real e imaginaria.

Como se observo en la primera ilustración, tenemos distintas presentaciones de suma o como sumar un número complejo. Matemáticamente se define:

La consideración por tomar en cuenta aquí en la clase es si queremos pasarle los parámetros, un número complejo por los dos números reales, un objeto del número complejo o los dos números complejos. Aquello que tienen como valor de retorno void, es porque estas funciones se guarda el resultado de la operación en el lugar u objeto donde se instancian.

En cambio, aquella que devuelve un complejo, permite la creación de otro objeto del tipo complejo que le pueda pasar el resultado de la suma de los números complejos.

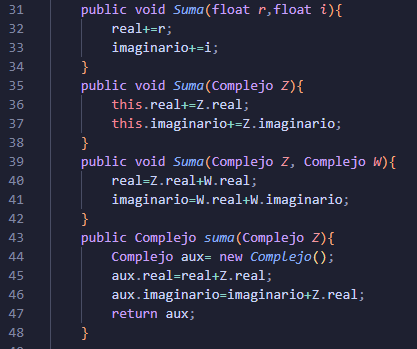
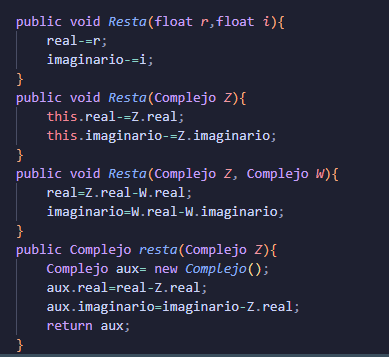


Ilustración 5 Operaciones, funciones de suma

La situación es demasiado parecida cuando se realiza una resta, ya que también tanto le podemos pasar el número a restar como si fueran dos flotantes, un objeto complejo o los dos objetos complejos. Además, podemos tanto guardar los valores en el objeto que llama a estas funciones, o podemos guardar en otro objeto complejo.

La resta realiza de manera muy similar, ya que se debe de tener en cuenta, que la resta es solamente la multiplicación de la identidad del número complejo con la inversa aditiva.



Y como se puede observar en la ilustración 6, las operaciones se dan dependiendo a que parámetros le pasemos y que devolvemos en las funciones.

También tenemos las funciones de multiplicación, que se definen como

Ilustración 6 Operaciones, funciones de resta

Las operaciones de las multiplicaciones se pueden ilustrar en la ilustración 7

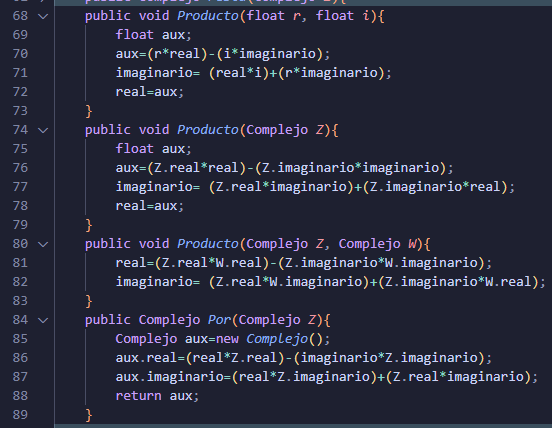
Como observamos matemáticamente, debemos definir un lado para poder calcular la parte real e imaginar, y para ello, podemos tener 3 funciones que almacenen el resultado en el objeto que las llamo, ya sea pasándole los dos números, real e imaginario, el complejo, o los dos valores complejos. O por su parte, podemos crear un objeto complejo pasándole a la función un solo valor complejo.

Ilustración 7, cuatro funciones de producto

La división también tiene su forma o manera de realizarse, ya que

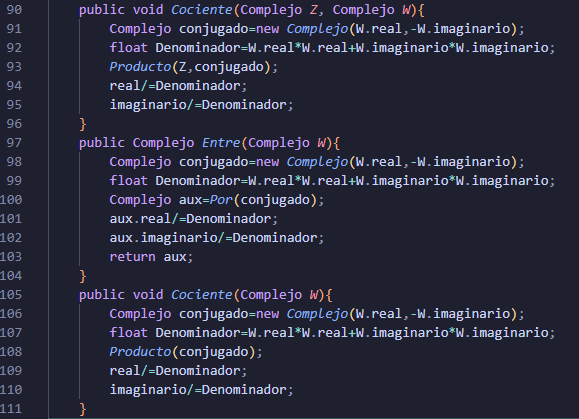
Teniendo esto en cuenta, podemos pasarle 3 funciones, puedes pasarle dos parámetros, dos objetos del tipo complejo, y que se guarden los cálculos dentro de la función que lo declaro, (así como también le podemos pasar un número complejo solamente). O por otra parte podemos crear un objeto del tipo complejo y pasarle solo un objeto complejo.

Ilustración 8. Cociente

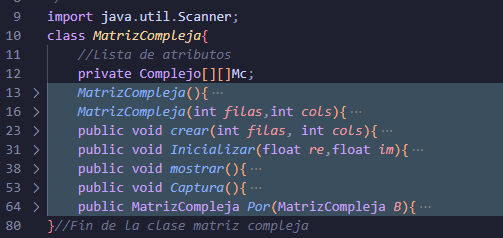
Todas estas funciones serán útiles para la clase de matriz.

Ilustración 9 Matriz compleja

Cuando compilamos el código junto con el otro que se hará con respecto a la matriz compleja, se genera un archivo del tipo class, del cual podremos utilizar en la clase matriz compleja. Tenemos que la clase tiene un atributo Mc privado del tipo complejo. Por el hecho de haber es compilado el primer archivo y que esté dentro del mismo directorio, generándose como un archivo class, java toma estos class y los toma como si en el mismo archivo hubiera estado importado. Entonces, sin problemas, todas las funciones que se habían declarado en Complejo, las podrá utilizar dentro de matriz compleja si y solo si usamos un objeto del tipo complejo.

La matriz compleja tiene dos constructores  
Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Ilustración 10 Constructores

Texto

Descripción generada automáticamenteEl primer constructor, solo hace que la referencia exista y sea de nulos, el segundo, pide el numero de filas y columnas que deseas utilizar para tu matriz. En base a eso, solo inicializa los números complejos en 0 dentro de cada uno de los índices o espacios dentro de la matriz. (Recordando que el constructor *Complejo (),* sin parámetros, los números r e i, son 0).

Ilustración 11 Inicializar

Texto

Descripción generada automáticamenteEn dado caso, la función de crear hace el mismo proceso que el constructor, y podemos reemplazar la acción del segundo constructor, a manera de abstraer la creación de una matriz compleja.

También se creó una función llamada *Inicializar,* en la que puedes inicializar lo valores de la matriz con un valor especifico, (ilustración 11).

Texto

Descripción generada automáticamenteTambién se creó una función llamada *Mostrar,* que permite imprimir en pantalla los números complejos escritos en una matriz compleja. Este proceso solamente recorrer la matriz de filas y columnas e imprime con la función “show”, que es una función propia de la clase Complejo. Más, sin embargo, debe de verificar que la matriz no sea nula. La función de captura, que se muestra en la ilustración 13, se encarga de pedirle al usuario, dadas las filas y columnas de la matriz, previamente creada, recorriendo la matriz inicializada con anterioridad y con la función Captura, que es una función propia de la clase Complejo, permite el número complejo que estará dentro de nuestra matriz.

Ilustración 12 Función mostrar

Finalmente, la función de multiplicar está definida como *Por*, en ella le pasamos un solo parámetro que es el objeto de tipo MatrizCompleja. Es decir, para multiplicar una matriz necesitas otra matriz.

Ilustración 13 Función Captura

Texto

Descripción generada automáticamenteEste concepto es demasiado similar en cuestión de una matriz de números reales tal que:

Ilustración 14 Multiplicación de matriz compleja

Para la multiplicación de matrices de números complejos se realiza el mismo proceso que con la matriz de números reales, solamente que se debe de tener en cuenta el cómo se multiplica un número complejo. Primeramente, el objeto que llame a la función de multiplicación será la que multiplicará con matriz pasada como parámetro. De tal forma que en la línea 65 creamos un nuevo objeto del tipo *matrizCompleja*. E instanciamos una variable auxiliar, que nos ayudará a la hora de hacer las multiplicaciones. Antes de realizar la multiplicación, se deben de verificar las longitudes de las matrices, que si la cantidad de columnas es la misma que la cantidad de filas de la segunda matriz. Si es así, se opera, si no, solamente devuelve la misma matriz inicializada en 0. Cuando se realiza el proceso de multiplicación, necesitamos dos iterados, el primero es el de las columnas de la matriz *A,* luego son las filas de la matriz B.Y un último iterador que será el pivote que se irá desplazando para hacer cada producto. En esta ocasión se utiliza la función Producto (Complejo, Complejo W). Esto porque quiero que en auxiliar se guarde el resultado. Teniendo el valor ya almacenado en Aux, ahora sumaremos el valor aux con lo que tenemos en la posición ,iterativamente lo haremos hasta terminar la longitud de las columnas de A.

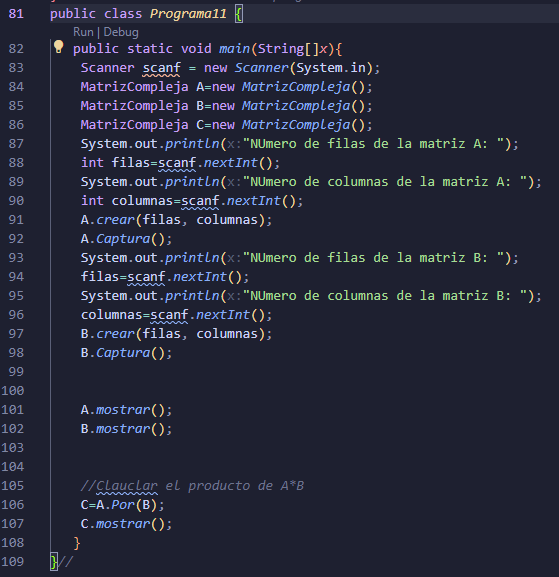
Y en la clase principal, a su vez que en la función principal lo que realiza es instanciar o llamar el paquete de Scanner, que será el que pedirá los datos al usuario. Primeramente, necesitamos pedirle al usuario que nos de la cantidad de filas y columnas. Instanciamos 3 matrices, y creamos la matriz tanto en A, como en B, a su vez de que capturarán los valores de cada uno de los elementos de la matriz. Después se llama a la función *mostrar*, para poder desplegar en pantalla los valores ingresados tanto en A, como en B.

Ilustración 15 Main

Finalmente se realiza la matriz y se imprime.

# Resultados:

Texto

Descripción generada automáticamenteSe pide primero la cantidad de filas y columnas en la matriz A, y se capturan los números complejos d el a matriz A. Después, hacemos lo mismo, pero en la matriz B. Para esta ocasión, ambas matrices tienen 2 columnas y 2 filas, por lo que, al multiplicarse, se juntan en una matriz de 2x2.

Ilustración 16 Resultado en la consola

La matriz A tiene los siguientes valores

Al multiplicar las matrices debemos de obtener la siguiente matriz.

Podemos verificarlo en el sitio de Symbolab, para realizar las matrices complejas  
Texto, Carta

Descripción generada automáticamente

Ilustración 17 Demostración con software externo

Como se puede observar, la multiplicación fue correcta, todos los elementos hicieron las multiplicaciones y sumas, además de que se almacenan dentro de los elementos en la matriz.

# Conclusión:

Con esta práctica creo una matriz compuesta, un término que se suele confundir con la herencia ya que como tal, la herencia consiste en la extensión de una clase madre. Sin embargo, las clases compuestas son un tipo de clases que dentro de ellas tienen otras clases,(como en este caso la clase complejo), reduciendo en gran medida el código que se hubiera tenido que implementar dentro de la matriz Java