Instituto Politécnico Nacional
Unidad Profesional
Interdisciplinaria De Ingeniería
Campus Zacatecas

Ingeniería en Sistemas Computacionales

Práctica 4: sobrecarga de funciones

Alumna: Vanessa Melenciano Llamas

Boleta: 2020670081

Profesora: Monreal Mendoza Sandra Mireya

Materia: Programación Orientada a Objetos

Grupo: 2CM2

Fecha: 27 de octubre de 2020

Índice

Introducción

Sobrecarga se refiere a la práctica de cargar una función con más de un significado. Básicamente, el término expresa que se cargan uno o más identificadores de función sobre un identificador previo.

La sobrecarga no es un concepto nuevo en los lenguajes de programación, por ejemplo el operador = está sobrecargado en muchos lenguajes de alto nivel y se utilizan en instrucciones de asignación y en expresiones condicionales.

La sobrecarga otorga flexibilidad, permite a las personas utilizar código con menos esfuerzo, ya que extiende operaciones que son conceptualmente similares en naturaleza.

El objetivo de la sobrecarga es reducir el número de identificadores distintos para una misma acción, pero con matices que la diferencian. La sobrecarga se puede realizar tanto en métodos generales, como en constructores.

Una clase puede tener varios constructores, que se diferencian por el tipo y número de sus argumentos (sobrecargados).

La sobrecarga de métodos hace que un mismo nombre pueda representar distintos métodos con distinto tipo y número de parámetros, manejados dentro de la misma clase. En el ámbito de la POO, la sobrecarga de métodos se refiere a la posibilidad de tener dos o más métodos con el mismo nombre pero distinta funcionalidad. Es decir, dos o más métodos con el mismo nombre realizan acciones diferentes y el compilador usará una u otra dependiendo de los parámetros usados. Esto también se aplica a los constructores (de hecho, es la aplicación más habitual de la sobrecarga).

Se pueden diferenciar varios métodos sobrecargados a través de sus parámetros, ya sea por la cantidad, el tipo o el orden de los mismos.

Gracias a la sobrecarga de métodos, una clase puede tener distinto comportamiento dependiendo de cual método sobrecargado se use.

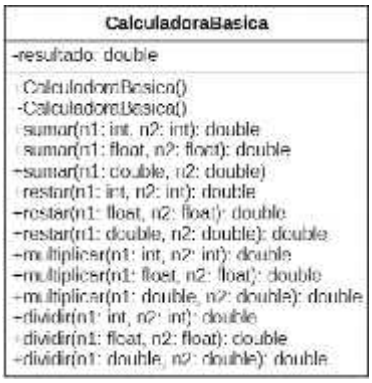
Objetivos

Implementar la sobrecarga de métodos para realizar operaciones aritméticas básicas y manejo de arreglos.

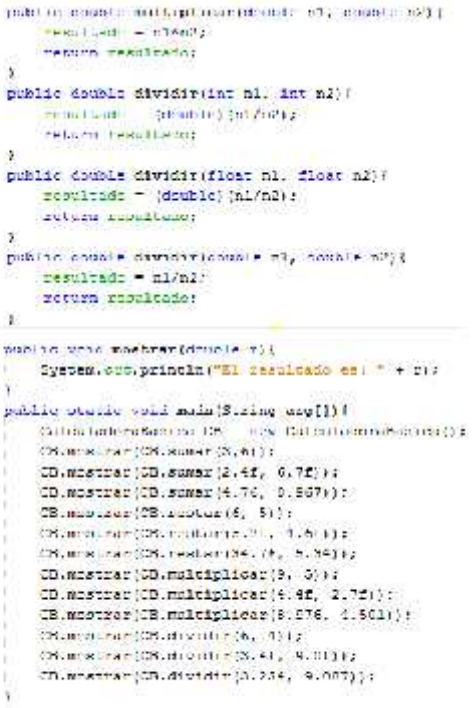
Desarrollo

Diseño (UML) y funcionamiento de la solución

Ejercicio 1



El ejercicio consiste en crear una especie de calculadora básica que realice sumas, restas, multiplicaciones y divisiones con números que pueden ser declarados como int, double, float, pero, en cualquier caso, la calculadora dará los resultados con un valor double.



La solución para este ejercicio es utilizando la sobrecarga de métodos, haciendo tres funciones para cada operación, una recibiendo valores int, otra float y al final double. Como atributo es el resultado, el cual se da valor en la función que se mande a llamar, para posteriormente imprimirlo en pantalla.

En el caso de las funciones que reciben valores diferentes a double, era necesario cambiarlas para evitar errores y que el resultado fuera double, así como la función retornara un doble.

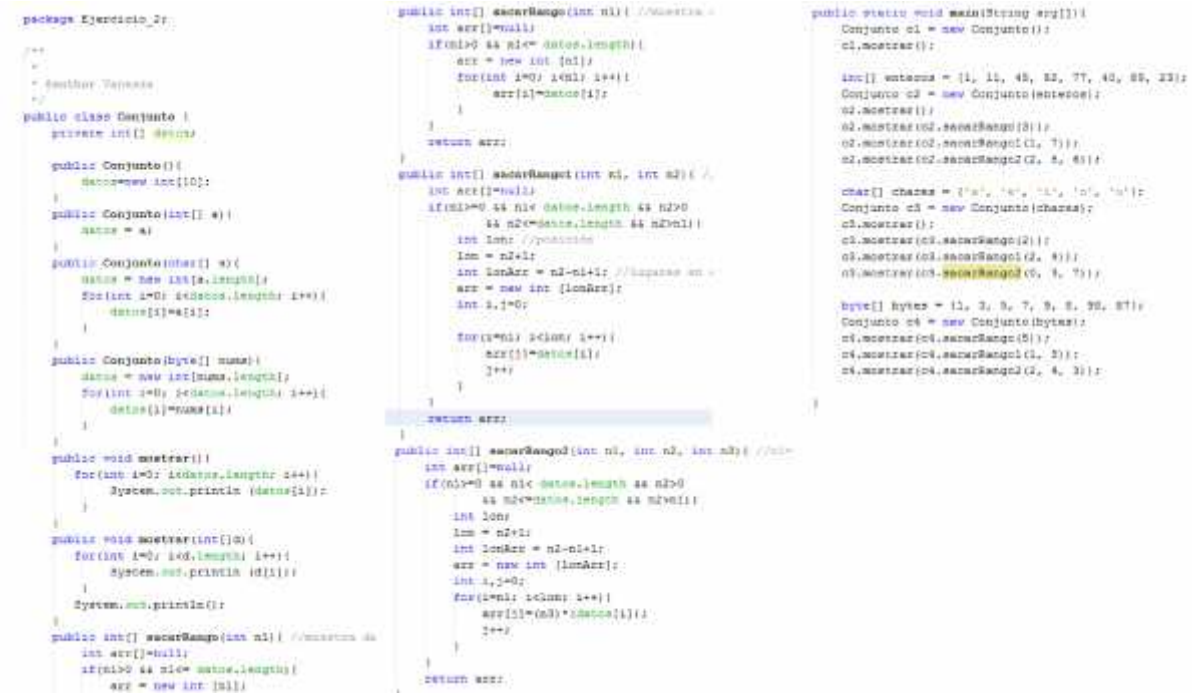
Ejercicio 2

Conjunto
-datos[]: int
+Conjunto() +Conjunto(a[]: int) +Conjunto(a[]: char) +Conjunto(nums[]: byte) ~Conjunto() +sacarRango(n: int): int[] +sacarRango(n1: int, n2: int): int[] +sacarRango(n1: int, n2: int, n3: int): int[] +mostrar(dat[]: int): void

Este ejercicio consiste en introducir una cadena, ya sea char, int o byte, y que imprima los caracteres de la cadena que se señalan en los valores mandados a cada función.

En el primer caso, se tiene que imprimir la cadena hasta la posición n, en el segundo caso, se imprime la cadena iniciando por n1 y terminando en n2.

En el último caso, se imprimen los números de las mismas posiciones que el anterior, la diferencia, es que se imprimirán multiplicados por n3.



Para la solución se utilizó la sobre carga de métodos, declarando tres que determinaran el rango, pero cada uno recibe diferente cantidad de valores.

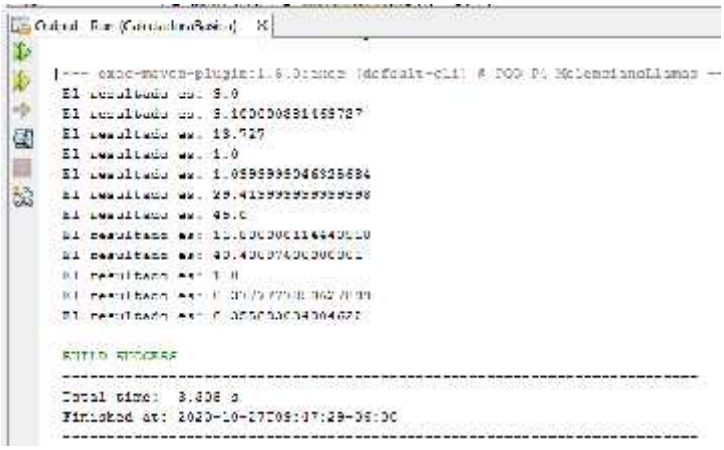
En el primero, solo imprime la cadena hasta n, y eso con ayuda de un for.

El segundo es casi igual, la diferencia es que recibe dos valores, uno de la posición de inicio, y otro de la final, y se implementa igualmanera con un for, marcando los limites del arreglo. Para esté caso se necesita la ayuda de otro contador, para manejar el arreglo de los datos guardados desde la posición señalada, y el arreglo que se va a imprimr empiece desde cero, para guardarlo correctamente

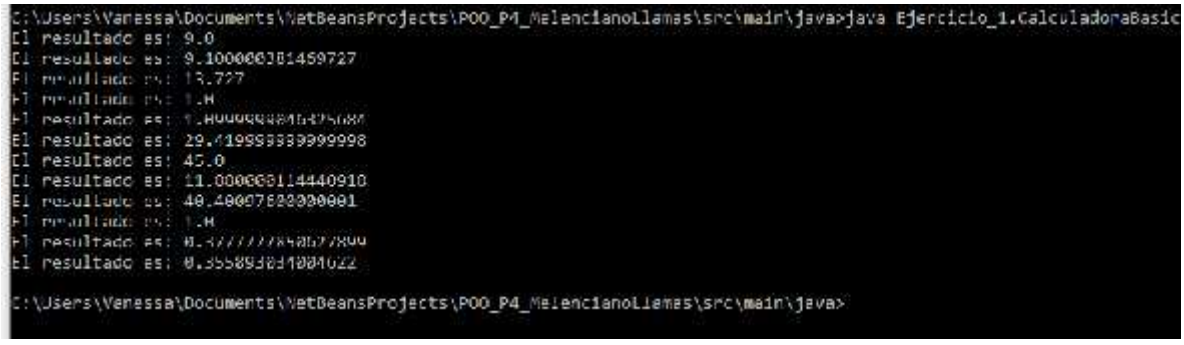
Funciones

Ejercicio 1

- Por el IDE

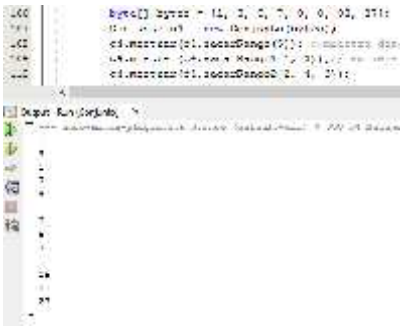


- Por consola

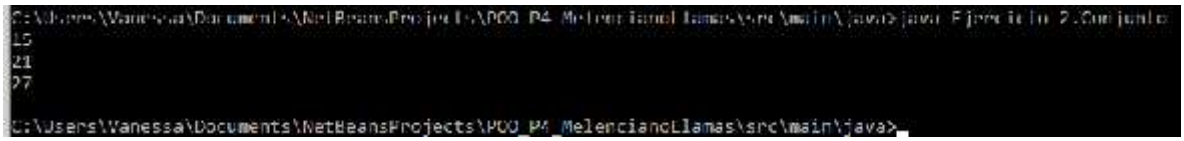


Ejercicio 2

- Por el IDE



- Por consola



Errores detectados

Al realizar la práctica, uno de los problemas presentados, fue el dominio de los métodos para su correcto funcionamiento. Y un poco en fallo de saber manejar la conversión entre los diferentes tipos de variables en java.

Posibles mejoras

Para mejorar el manejo de los temas de la programación orientada a objetos, es importante mantener una constante practica de estos, por ello para poder ir mejorando, realizar el trabajo es importante, y por ahora sería todo.

Conclusión

El concepto de sobrecarga de operadores, es una poderosa herramienta que permite a los programadores añadir funcionalidad y nuevos significados a los operadores primitivos.

El objetivo de la práctica se logró cumplir, al lograr comprender el concepto de sobrecarga de métodos en POO, utilizando las operaciones aritméticas básicas y los arreglos.

Referencia

Barnes, D., & Kolling, M. (2007). *Programación orientada a objetos con Java* (3ª edición ed.). Madrid, España: PEARSON EDUCACIÓN.

Martínes, G. (2009). PROGRAMACIÓN ORIENTADA A OBJETOS CON APRENDIZAJE ACTIVO. *Scientia Et Technica*, pp. 163-168.