

Tecnologías y paradigmas de programación. Actividad presencial del laboratorio 2. Grupos 1 y 4.

1. Cree una nueva solución con un proyecto de aplicación de consola. Encuentre una manera de agregar a esta solución el proyecto de laboratorio anterior "Geometría" donde se definió Point2d. Estos ejemplos le resultarán útiles:

utilidad.clases en encapsulación y parámetro.paso,

parámetros.opcionales.con nombre en sobrecarga. Implemente los siguientes métodos estáticos en el proyecto de aplicación de consola:

1. IncrementoDosDoble. Toma cuatro parámetros dobles, primero y segundo por referencia.
Modifica estos dos parámetros agregando el valor del tercer y cuarto parámetro respectivamente, devuelve void. Por ejemplo, esta llamada: IncrementTwoDouble(ref x,ref y,1.5,2.0) suma 1,5 a x y 2,0 a y.
2. Generar trayectoria. Toma un parámetro de longitud (int, valor predeterminado 10), comienza coordenadas xey (doble, valor predeterminado ambos 0,0), incremento xey (doble, valor predeterminado ambos 1,0). Devuelve una trayectoria, almacenada en un array Point2d con el número de elementos de longitud, donde todos los puntos excepto el primero se obtienen sumando al punto anterior de la trayectoria los incrementos indicados mediante la función IncrementTwoDouble previamente definida. Ver proyecto de matrices en encapsulación de solución.
3. ShowTrajectory, toma una matriz que representa una trayectoria como se describió anteriormente.
y un booleano opcional, verdadero si solo se van a mostrar los puntos del primer cuadrante ($x \geq 0$ y $y \geq 0$) falso en caso contrario. Muestra la representación textual de los puntos de trayectoria a la consola.
4. DistanceStartEnd, toma una matriz Point2d y, como referencias de salida, tres parámetros: distancia recorrida de la trayectoria (la suma de las distancias entre cada punto y el siguiente), puntos inicial y final. ¿Qué sucede con los valores del array de trayectoria si las propiedades devueltas de los puntos inicial y final se modifican después de la llamada a la función? Genere la trayectoria para verificarla.
5. Defina una matriz Point2d en Main que contenga una trayectoria, verifique si las funciones implementadas y los mecanismos de paso de argumentos funcionan como se esperaba.

2. Obtenga información sobre los métodos de extensión en la solución de sobrecarga. Extiende int con lo siguiente métodos (los ejemplos están escritos usando constantes por simplicidad):

1. Invertir, devuelve el int obtenido invirtiendo el orden de los dígitos de la persona que llama: 123456.Reverse() devuelve 654321 como int.
2. IsPrime, devuelve verdadero si la persona que llama es principal, falso en caso contrario: 13.IsPrime() devuelve verdadero, 14.IsPrime() devuelve falso como booleano.
3. mcd y Lcm, opcional: 24.Gcd(36) es 12, nuevamente un número entero.
4. Concatenar, opcional: 123.Concatenate(456) devuelve 123456 como int. Precaución cuando el segundo sea negativo, elija cualquier solución para este caso, suelte el signo o lo que sea.

5. Compruebe si todo funciona como se esperaba en un proyecto de aplicación de consola.

3. Utilice el código fuente proporcionado para crear un proyecto de prueba unitaria. Implemente la clase solicitada junto con los métodos y operadores para poder pasar las pruebas. Ver operadores en sobrecarga

Sube la actividad obligatoria antes del próximo Lab.