Lab exercise 1: Implementing a class
24292-Object Oriented Programming

Simón Gasión: 240126
Jordi Polo: 239980

1.Introduction

El programa debe de ser capaz de crear y mostrar una matriz con diferentes ciudades y la distancia entre ellas, donde cada ciudad es un punto en concreto con sus coordenadas “x” e “y” respectivamente.

Para implementar esta solución debemos de crear dos principales clases, la primera de ellas “GeometricPoint” la cual definirá las ciudades representadas por un punto cada una, está tomará como atributos las coordenadas “x” e “y” y un nombre “name”, el qual es usado para definir el nombre de la ciudad (punto). En cuanto a métodos, tendrá sus respectivos “getters” y “setters”, su método constructor y un método para calcular la distancia entre puntos y otro para imprimir los puntos (mostrar por pantalla).

Como segunda clase principal y no por ello menos importante sino esencial, definiremos la clase “DistanceMatrix” la cual creará la matriz con las diferentes ciudades y las respectivas distancias entre estas. Esta tendrá como atributos una LinkedList de ciudades de tipo “GeometricPoint” y una matriz de tipo matriz de reales “double matrix”. En cuanto a métodos, tendrá el método constructor y sus respectivos “getters”, donde podremos retornar el nombre de la ciudad, el número de ciudades y la distancia entre una y otra. Aparte de estos métodos también tendremos el método “addCity” con el cual podremos añadir una ciudad (punto) a la matriz y el método “createDistanceMatrix” con el cual crearemos la matriz.

Posteriormente hemos definido varias clases como “TestPoint” y “TestDistanceMatrix”, estas nos sirven únicamente para probar si las clases principales funcionan correctamente, implementando un “main” y probando los métodos de las clases en estos, llamando a estos métodos para diferentes ejemplos.

En el caso del ejercicio opcional implementado al final de la tarea, simplemente hemos implementado el codigo proporcionado por la guia de la practica. De esta manera se muestra por pantalla una pestaña aparte en la que deja añadir los propios puntos de tipo “GeometricPoint”, y refleja la propia matriz.

2.Description

La práctica como tal ha sido guiada por el documento correctamente y por tanto hemos seguido el camino específico de la práctica sin tener ningún tipo de debate entre maneras de hacer las clases. Simplemente hemos intentado hacerlas óptimas y entendibles. Por eso mismo no hemos tenido ninguna otra solución en mente que no sea la aplicada a la práctica.

Para esta hemos implementado varios conceptos de teoría, como la creación de clases en general, implementando los atributos de estas correctamente y los propios métodos, tanto los específicos para esta práctica como los teóricos y generales que se aplican a casi todas las clases, siendo estos los “getters”, “setters” y el propio método constructor de cada clase.

Aparte de la teoría más básica de java presentada en teoría, hemos utilizado conceptos como las LinkedList y matrices, concretamente para la clase “DistanceMatrix”, siendo en el caso de la LinkedList una lista de tipo “GeometricPoint” la clase la cual hemos definido anteriormente de manera sencilla sin ninguna complicación.

Claramente maniobrando los conceptos que traen las LinkedList y las matrices y con su complejidad, como en el caso de la LinkedList utilizar métodos como “add” para añadir un elemento, y en el caso de matrices el hecho simple pero más complejo que para la clase anterior de iterar en ella.

3.Conclusion

La solución del problema funciona perfectamente, habiendo sido probada tanto en la clase “TestPoint” como en la clase “TestDistanceMatrix” probando varios ejemplos de puntos y creando una matriz con estos. Mostrándonos que los métodos estaban bien implementados y no mostrando ningún problema a la hora de su ejecución.

Hemos tenido algún que otro pequeño problema y de fácil solución a la hora de programar, esto debido a que no estábamos familiarizados con el lenguaje de java y cosas como implementar el “main” o utilizar algún método propio de java, nos han dado algún pequeño inconveniente, haciendo que nos retrasaramos en buscar el fallo y la solución a este, que como ya he mencionado, soluciones rápidas y sencillas.

En cuánto al ejercicio opcional, hemos tenido un problema porque en nuestras funciones getCityName() y getDistance(), asumíamos que se esperaba que el índice 1 fuera el primer elemento en vez del 0 (implementado llamando a las funciones con index-1), pero el código de DisplayMatrix no lo hace, así que al intentar testear ésta parte del trabajo surgía un error al intentar llamar al elemento -1 de la lista.