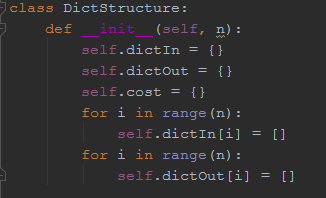
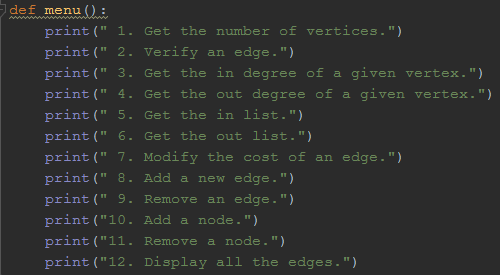
**Graphs: practical work no. 1**

**Toadere Andreas-Robert Python**

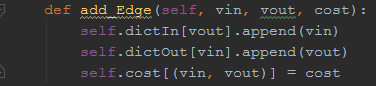
For this implementation I chose the Python programming language. The directed graph is represented by three dictionaires: dictIn, dictOut and cost. For dictIn and dictOut the key is an integer ( a vertex ) and the value is a list of inbound or outbound neighbours. For cost, the key is a pair of two vertices and the value is an integer ( the cost ):



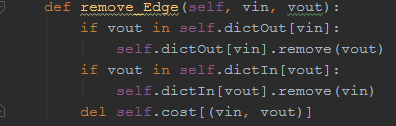
For the user-interface the program has a basic command menu, where each command is represented by a number. This is the list of commands:



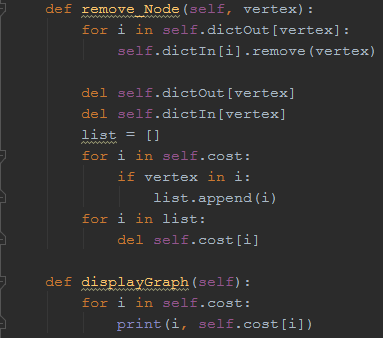
The following pictures will show the key functions of the implementations:



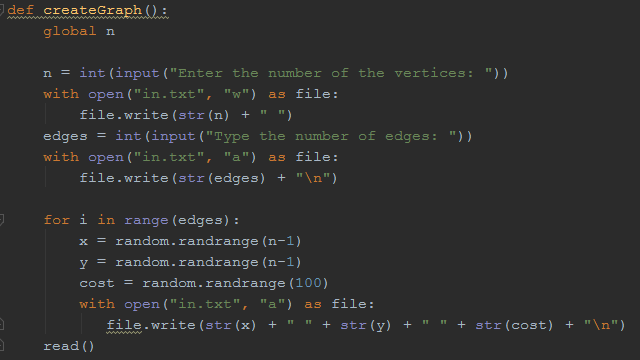
To add a new edge we need to add its vertices in the two dictionaires and in the cost.



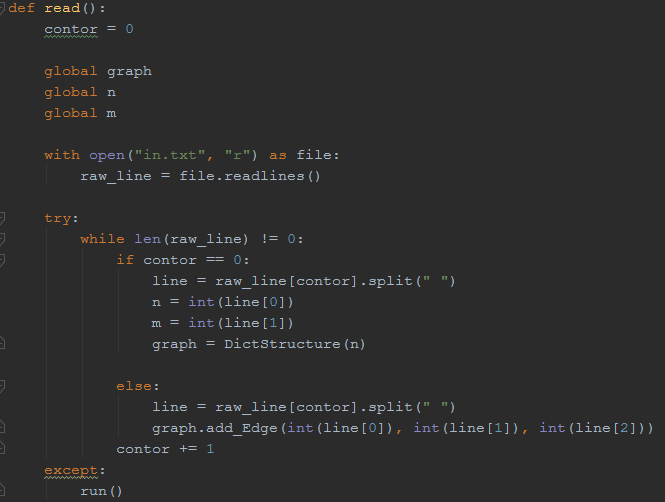
To remove an edge we have to remove the ends from dictIn and dictOut and then delete the element in the cost dictionary.



To remove a node we have to remove all its appearances from the two dictionaires, then delete it as a key in them and then delete every appearance of the vertex in the cost.



This is the function that creates a random graph.



This is the function that reads a graph from file.