Task 6: Working with inheritance

1. Be Graph a class for representing unweighted Graphs. This tasks requires to implement several classes deriving Graph.

Implement a class WeightedGraph in which every edge is assumed to have a weight (of type int). Assume all the weights are positive.

WeightedGraph offers a public member function int getWeight(int v, int w), which returns the weight of the edge (v,w) or -1 if the edge does not exist.

In the class WeightedGraph , getWeight(a,b)==getWeight(b,a) always evaluates to true.

Implement also a class DirectedGraph which is also a Graph but with no weighted edges. In this class the existence of a edge from vertex u to vertex w is independent of the existence of a vertex from w to u.

Implement also a class DirectedWeightedGraph which is a directed graph in which edges are weighted. In this class, getWeight(a,b) may be different than getWeight(b,a).

Implement a search method int minPath(a,b). This method should return the minimum number of edges to go from a to b in a non-weighted graph, and the minimum weight to go from a to b in a weighted graph. If there is no path between a and b, the method should return -1.

Include some examples testing your classes.

2. Be the following class modeling a estimation of how many kilometers a car can still run with the current fuel in its tank

```
class KilometersEstimator
{
    public:
        virtual int getKilometersLeft() const {
                return getKilometersPerLiter() * getLitersLeft();
        }
        virtual void setLitersLeft(int inValue) { mLitersLeft = inValue; }
        virtual int  getLitersLeft() const { return mLitersLeft; }
    private:
        int mLitersLeft;
        int getKilometersPerLiter() const { return 20; }
};
```

A new more efficient estimator has been developed recently and it has been proved that the number of kilometers per liter is 17 instead of 20. Use inheritance to develop a new class EfficientKilometerEstimator. You can do any change you want except changing the visibility of any of the methods already in the interface of KilometersEstimator.