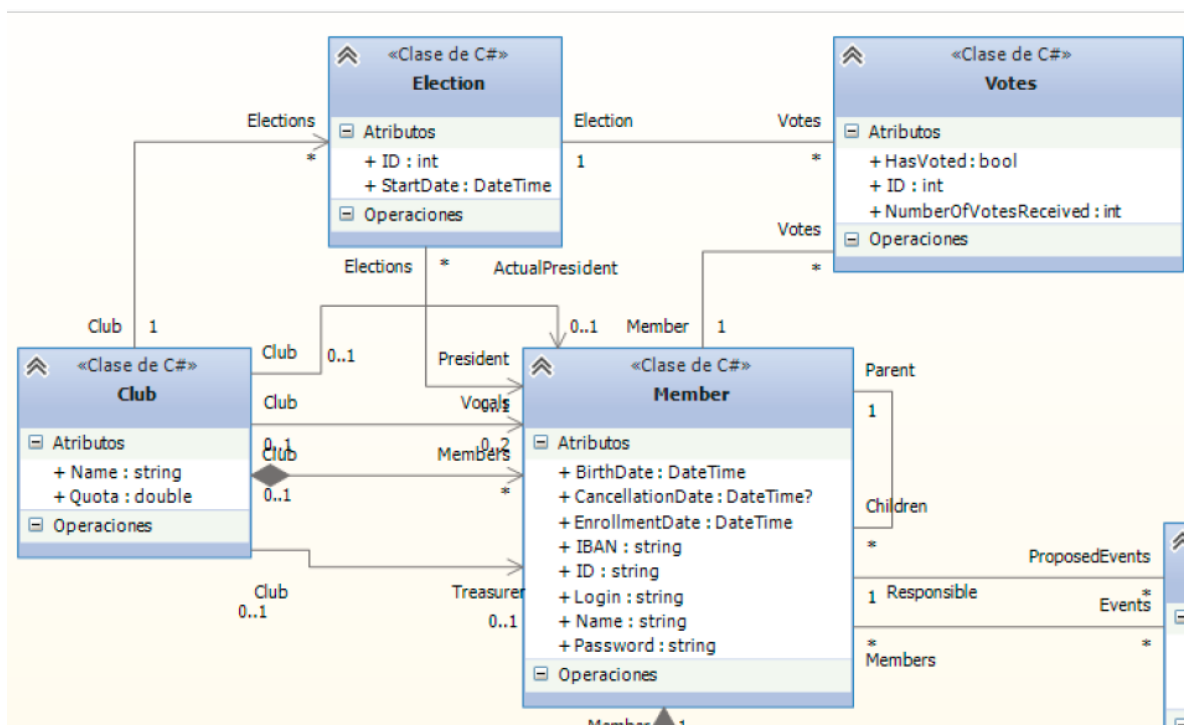


Ejercicio 1. Diseñad los casos de prueba para el siguiente fragmento de código siguiendo la técnica del camino básico (dibujad el grafo de flujo, calculad la complejidad ciclomática, especificad los caminos independientes y los casos de prueba asociados a cada camino).

El código corresponde a un método que genera una lista de los miembros más votados en función del método de recuento elegido. El método recibe como parámetros una fecha válida, el valor a partir del cual se considera que un miembro ha obtenido un número significativo de votos y el método de recuento. El resultado del método es una lista de miembros más votados ordenada de más a menos votos.

```
List<Votes> generarResultado(DateTime fecha, int valorSignif, TipoRecuento tipoRecuento)
{
    Election eleccion = findElectionByStartDate(fecha);
    if (eleccion == null) throw new ServiceException("La elección no existe.");
    if (getLoggedMember() != eleccion.President) throw new ServiceException("Sólo...");
    List<Votes> votosSignificativos = new List<Votes>();
    foreach(Votes voto in eleccion.getVotes())
    {
        if (voto.NumberOfVotesReceived >= valorSignif)
            votosSignificativos.Add(voto);
    }
    List<Votes> votosOrdenados;
    if (tipoRecuento == TipoRecuento.Hondt)
        votosOrdenados = recuentoHondt(votosSignificativos);
    else votosOrdenados = recuentoSainte(votosSignificativos);
    return votosOrdenados;
}
```



Ejercicio 2. Diseñad los casos de prueba para el siguiente fragmento de código siguiendo la técnica del camino básico (dibujad el grafo de flujo, calculad la complejidad ciclomática, especificad los caminos independientes y los casos de prueba asociados a cada camino). El código se corresponde con un método de una clase Club que devuelve una cadena que contiene los resultados de participación de la elección que se pasa por parámetro.

Nota: para el diseño del test case, debe describirse el contenido del objeto Club que se usará en la prueba (número de miembros, número miembros que han votado, etc.). Además, podemos considerar que el Club contendrá una sola elección correspondiente a DateTime.Now.

```
public string getParticipationResults(DateTime electionDate)
{
    Election election = this.findElectionByDate(electionDate);
    if (election == null)
    {
        throw new ServiceException("Elecction doesn't exists");
    }
    else if (this.Members.Count > 0)
    {
        int votes = 0;
        foreach (Member member in this.Members)
        {
            if (member.hasVotedIn(election))
                votes++;
        }
        double participation = votes / this.Members.Count;
        return "The participation percentage at " + DateTime.Now.ToShortDateString() + " is "
            + participation.ToString("P2");
    }
    else
    {
        throw new ServiceException("There are not possible electors");
    }
}
```

