ESTRUCTURA DE DADES PRÀCTICA 4

Anàlisi de requeriments

Els principals requisits per a dur a terme aquesta pràctica son principalment 3:

-tenir una sèrie de certs coneixements de Java i l'ús d'un programa per a implementar aquest codi.

-dispondre d'un editor de text per a realitzar l'informe posterior tal com el Word i passar-lo després a PDF.

-tenir els recursos suficients per superar tots el problemes que et trobis durant la realització de la practica, com ara molta paciència i ganes de treballar.

Estudi Previ

Per a anar iniciant el desenvolupament de la practica primerament tractarem la implementació dels diferents TADs que empraré, per fer això, crearé un TAD de partit per emmagatzemar la informació d'un partit, tot seguit crearé una classe que serà el TAD del graf(on tindre dos subclasses mes per guardar la informació de tots els partits d'una lliga, la classe Node Vertex i Node Aresta) i per altra banda una TAD de partits on donaré us als mètodes del graf per dur a terme les consultes que es demana.

Per anar finalitzant generaré una classe Lliga per a poder tractar tots els partits i equips segons la lliga que estan disputant i posteriorment generaré totes les classes necessàries per controlar les excepcions definides. Últimament crearé el main on ajuntaré totes les funcions descrites al llarg de les classes per a generar finalment un programa 100% funcional.

Decisions preses durant el desenvolupament de la pràctica

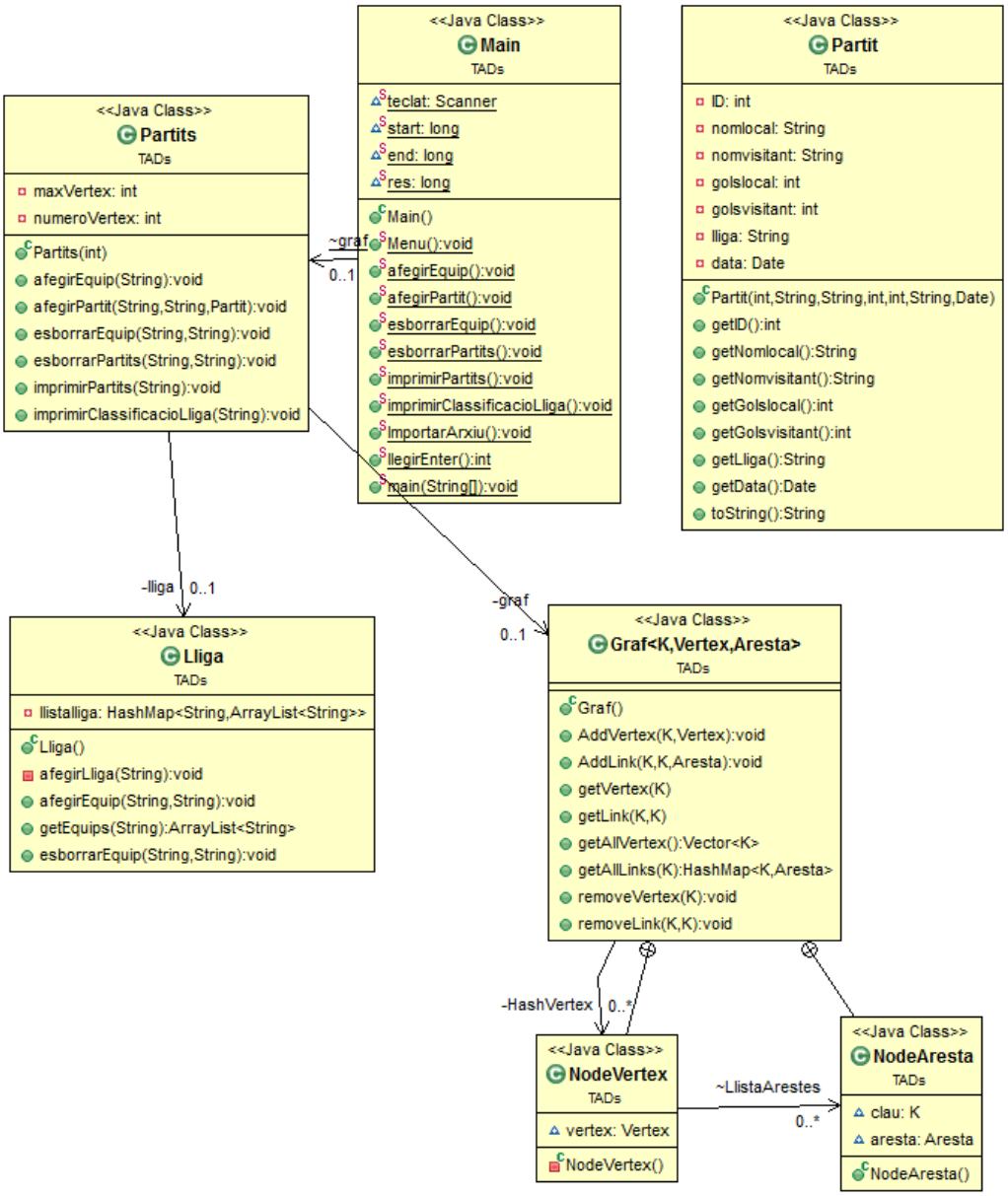
-Per a controlar que el programa no deixi de funcionar durant el seu funcionament, he decidit crear cinc excepcions diferents, una que per controlar que no hi hagi dos equips a la taula de equips amb el mateix nom, una altra per controlar quan no existeix un equip a la taula, una altra per comprovar quan la taula esta plena i no hi caben mes equips, una altra per comprovar si el partit en qüestió esta a la taula i finalment un altre per comprovar si la lliga que volem consultar existeix. També he decidit crear un mètode anomenat llegirenter() per tal de controlar que el valor introduït sigui un enter on es demana un numero, evitant així molts errors.

-Per a la implementació de la classe graf, he crear dos subclasses per poder guardar de forma correcta els partits i els equips, per fer això he generat un NodeVertex, que contindrà un equip(que serà la clau) i una arraylist de NodeAresta(que contindrà una clau i el partit) amb tots els equips als que s'enfronta; finalment he guardat aquestes dos estructures en una hashtable.(Tal com esta dissenyat no es podrà inserta dos vegades el mateix partit).

-Per a poder fer la primera part opcional, he decidit generar una altre classe, la classe Lliga amb els mètodes de afegirLliga, afegirEquip, esborrarEquip i getEquips(que retorna tots els

equips de la lliga); en la qual dintre d'una hashmap, guardar la lliga com a string i una arraylist de string amb totes els equips d'aquesta lliga.

Diagrama de classes utilitzades



Cost temporal de les operacions

COSTOS TEMPORALS	PARTITS
afegirEquip	N
afegirPartit	1
esborrarEquip	N^2
esborrarPartits	N
imprimirPartits	N
imprimirClassificacióLliga	N^2

Aspectes millorables

Els principals aspectes a millorar suposo que seria intentar disminuir el codi que he implementat segons els mètodes, però ara per ara no disposo d'uns coneixements tan desenvolupats de java per reduir l'ocupació del codi mantenint la mateixa eficiència.

Un altre aspecte a millorar a part de disminuir l'ocupació del codi seria intentar fer mètodes encara mes eficients amb uns costos temporals mínims.

Per finalitzar també es podria implementar una interfície gràfica per no tenir que treballar mitjançant la consola tot el temps.