**ESCOLA POLITÈCNICA SUPERIOR**

**UNIVERSITAT DE GIRONA**



**PROJECTE: Tipus abstractes i mòduls funcionals**

Elaborat per:

**David Martínez, u1939690**

**Roger Barnés, u1939667**

**Matèria:** Projecte de Programació

**Grup:** Pràctiques L1, GEINF

**Versió:** 2.0

**Professor:** Dr. Francesc Castro & Dr. Miquel Bofill

Montilivi, Girona, 22 de març de 2017

1. **Tipus Abstractes de dades**

**Tipus Client**

Descripció general: Conté informació d’un client i les seves preferències.

*Pre: --*

*Post: Es crea un client amb nom “n” i preferències “prefs”*

Client (String n, Collection<String> prefs)

*Pre: --*

*Post: Retorna un iterador a les preferències del client*

Iterator<String> obtenirPreferencies()

**Tipus grupClients**

Descripció general: Conté un grup de clients, amb una categoria d’allotjament preferent i uns llocs que s’han de visitar prefixats.

*Pre: --*

*Post: Es crea un conjunt de clients amb tants clients com té “clients”, categoria desitjada i punts d’interès a visitar.*

Clients (Collection<Client> clients, Integer catDesit, Collection<PuntInteres> pI)

*Pre: --*

*Post: Retorna un enter que representa el nombre de clients del conjunt que tenen la preferència “pref” entre les seves preferències personals.*

Integer obtenirSatisfaccioPreferencia(String pref)

*Pre: --*

*Post: Retorna cert si l’allotjament “hotel” és de la categoria que desitja el conjunt de clients.*

Boolean categoriaDesitjada(Allotjament hotel)

**Tipus HoraDia**

Descripció general: Conté informació d’una hora d’un dia de la setmana

*Pre: dia ha d’existir dins de «dl,dm,dc,dj,dv,ds,dg»,*

*0 <= hora <= 23. 0 <= minuts <= 59*

*Post: Crea una Hora d’un dia a partir del dia de la setmana “dia” i les hores i minuts en format 24h.*

HoraDia(String dia, Integer hora, Integer minuts)

**Tipus FranjaHoraria**

Descripció general: Conté una franja horària compresa entre dos hores determinades.

*Pre: --*

*Post: Crea una franja horària a partir d’un HoraDia inici i final.*

FranjaHoraria(HoraDia inici, HoraDia fi)

*Pre: --*

*Post: Retorna cert si l’HoraDia “inst” es troba dins de l’horari de la franja*

Boolean pertanyFranja(HoraDia inst)

**Tipus PuntInteres**

Descripció general: Representa un punt on els clients tenen interès, bé sigui per visitar o bé allotjar-se.

*Pre: --*

*Post: Es crea un punt d’interès de nom “nom” amb les activitats que ofereix “acts” i el preu “cost”*

PuntInteres(String nom, Collection<String> acts, Double cost)

*Pre: --*

*Post: Retorna cert si el punt d’interès satisfà la preferència “pref”*

Boolean satisfaPreferencia(String pref)

*Pre: --*

*Post: Retorna el nom del punt d’interès.*

String obtenirNom()

*Pre: --*

*Post: Retorna el preu que costa visitar/allotjar-se al punt d’interès. Si és gratis retorna 0.*

Double obtenirPreu()

*Pre: --*

*Post: Retorna el lloc on està vinculat el punt d’interès*

Lloc obtenirLlocPare()

**Tipus PuntVisitable refina PuntInteres**

Descripció general: Representa un punt visitable, amb les seves franges horàries i temps de visita.

*Pre: –*

*Post: Es crea un punt visitable a partir de la informació del punt d’interès i el temps mitjà de visita.*

PuntVisitable (PuntInteres pI, Integer tempsVisita)

*Pre: –*

*Post: S’ha afegit la franja horària “fh” al punt visitable.*

void afegirFranja(FranjaHoraria fh)

*Pre: –*

*Post: Retorna el temps mitjà de visita.*

Integer obtenirTempsVisita(FranjaHoraria fh)

*Pre: –*

*Post: Retorna cert si el punt visitable està obert en un DiaHora determinat.*

Boolean estaObert(DiaHora inst)

**Tipus Allotjament refina PuntInteres**

Descripció general: Representa un allotjament, amb la seva categoria.

*Pre: 1<= categoria <= 5*

*Post: Es crea un allotjament a partir de les dades del punt d’interès i la categoria del allotjament.*

Allotjament(PuntInteres pI, Integer categoria)

*Pre: --*

*Post: Retorna la categoria de l’allotjament*

Integer obtenirCat()

**Tipus Coordenades**

Descripció general: Representa una localització amb latitud i longitud, juntament a la zona horària on està compromesa

*Pre: --*

*Post: Es crea unes coordenades amb latitud, longitud i zona horària*

Coordenades(String latitud, String longitud, Integer zH)

*Pre: --*

*Post: Retorna la zona horària de les coordenades*

Integer obtenirZonaHoraria()

**Tipus Lloc**

Descripció general: Representa un lloc amb les seves coordenades, generalment una ciutat, que pot contenir punts d’interès.

*Pre: --*

*Post: Es crea un lloc de nom “nom” i coordenades coor.*

Lloc(String nom, Coordenades coor)

*Pre: --*

*Post: Retorna el nom del lloc*

String obtenirNom()

*Pre: --*

*Post: Retorna les coordenades del Lloc*

Coordenades obtenirCoordenades()

*Pre: --*

*Post: Afegeix el punt d’interès “pI” als punts d’interès del lloc*

void afegirPuntInteres(PuntInteres pI)

**Tipus MitjaTransport**

Descripció general: Conté informació del mitjà de transport.

*Pre: --*

*Post: Es crea un mitjà de transport de nom “nom”*

MitjaTransport(String nom)

*Pre: --*

*Post: Afegeix el trajecte “traj” als trajectes que fa el mitjà de transport*

*void afegirTrajecte(Trajecte traj)*

**Tipus MitjaTransportDirecte**

Descripció general: Conté informació del mitjà de transport directe i als llocs o punts d’interès on està disponible.

*Pre: --*

*Post: Afegeix el lloc “ciutat” a la col·lecció de ciutats on està disponible aquest tipus de transport directe, i assigna un preu i temps estàndard de recorregut. Genera els trajectes entre llocs d’interès necessaris*

void afegirLloc(Lloc ciutat, Double preu, Integer temps)

**Tipus MitjaTransportIndirecte**

Descripció general: Conté informació del mitjà de transport indirecte i les seves estacions

*Pre: --*

*Post: Afegeix la estació est a la llista d’estacions del mitjà de transport indirecte*

void afegirEstacio(Estacio est)

**Tipus Estacio**

Descripció general: Representa un HUB d’un mitjà de transport indirecte en una ciutat.

*Pre: --*

*Post: Crea una estació del mitjà de transport indirecte “mtI” al lloc “ciutat”, amb un temps d’origen i destí en minuts.*

Estacio(MitjaTransportIndirecte mtI, Lloc ciutat, Integer tOrigen, Integer tDesti)

**Tipus Desplaçament**

Descripció general: Conté informació d’un desplaçament juntament amb el mitjà de transport que l’ofereix.

*Pre: --*

*Post: Es crea un Desplaçament d’un mitjà de transport mT amb una durada en minuts i un preu.*

*D*esplaçament(MitjaTransport mT, Integer durada, Double preu)

*Pre: --*

*Post: Retorna el mitjà de transport que ofereix el desplaçament*

*MitjaTransport obtenirMitja*()

*Pre: --*

*Post: Retorna la durada del desplaçament en minuts*

*Integer obtenirDurada*()

*Pre: --*

*Post: Retorna el preu del desplaçament*

*Double obtenirPreu*()

**Tipus Trajecte estén Desplaçament**

Descripció general: Conté un desplaçament entre dos llocs o punts d’interès.

*Pre: --*

*Post: Retorna el desplaçament del trajecte*

Desplaçament obtenirDesplaçament()

**Tipus TrajecteExtern refina Trajecte**

Descripció general: Conté un trajecte entre llocs.

*Pre: --*

*Post: Crea un TrajecteExtern a partir d’un desplaçament, una distància, i llocs d’origen i destí*

TrajecteExtern(Lloc origen, Lloc desti, Desplaçament desp, Double dist)

*Pre: --*

*Post: Retorna el lloc d’origen del trajecte*

Lloc obtenirOrigen()

*Pre: --*

*Post: Retorna el lloc de destí del trajecte*

Lloc obtenirDesti()

*Pre: --*

*Post: Retorna la distància entre el lloc origen i el lloc destí*

Double obtenirDist()

**Tipus TrajecteIntern refina Trajecte**

Descripció general: Conté un trajecte entre punts d’interès.

*Pre: --*

*Post: Crea un TrajecteInterm a partir d’un desplaçament i punts d’interès d’origen i destí*

TrajecteInterm(PuntInteres origen, PuntInteres desti, Desplaçament desp)

*Pre: --*

*Post: Retorna el punt d’interès d’origen del trajecte*

PuntInteres obtenirOrigen()

*Pre: --*

*Post: Retorna el punt d’interès de destí del trajecte*

PuntInteres obtenirDesti()

**Tipus Circuit**

Descripció general: Conté el conjunt de punts d’interès i trajectes dels viatges, juntament amb els llocs on s’ha passat.

*Pre: --*

*Post: Crea un circuit inicial buit*

Circuit()

Pre: --

Post: Afegeix un lloc al circuit

void afegeixLloc(Lloc ll)

*Pre: --*

*Post: Afegeix un trajecte al circuit*

void afegeixTrajecte(Trajecte t)

*Pre: --*

*Post: Afegeix un punt d’interès al circuit*

void afegeixPuntInteres(PuntInteres pI)

**Tipus Mapa**

Descripció general: Mapa de Llocs amb els seus TrajecteExtern i Punts d’interès amb els seus TrajecteInterns, on els punts d’interès estan interconnectats amb els llocs.

*Pre: --*

*Post: Crea un mapa amb una mida prefixada (Opcional)*

Mapa(Integer midaDefecte)

*Pre: --*

*Post: Afegeix un punt d’interès al mapa.*

void afegeixPuntInteres(PuntInteres pI)

*Pre: --*

*Post: Afegeix un desplaçament al mapa entre dos llocs a partir d’un trajecte*

void afegeixDesplaçament(Trajecte traj)

**Tipus CalculCircuits**

Descripció general: Mòdul funcional que conté algorismes per el càlcul de circuits a partir del mapa.

*Pre: --*

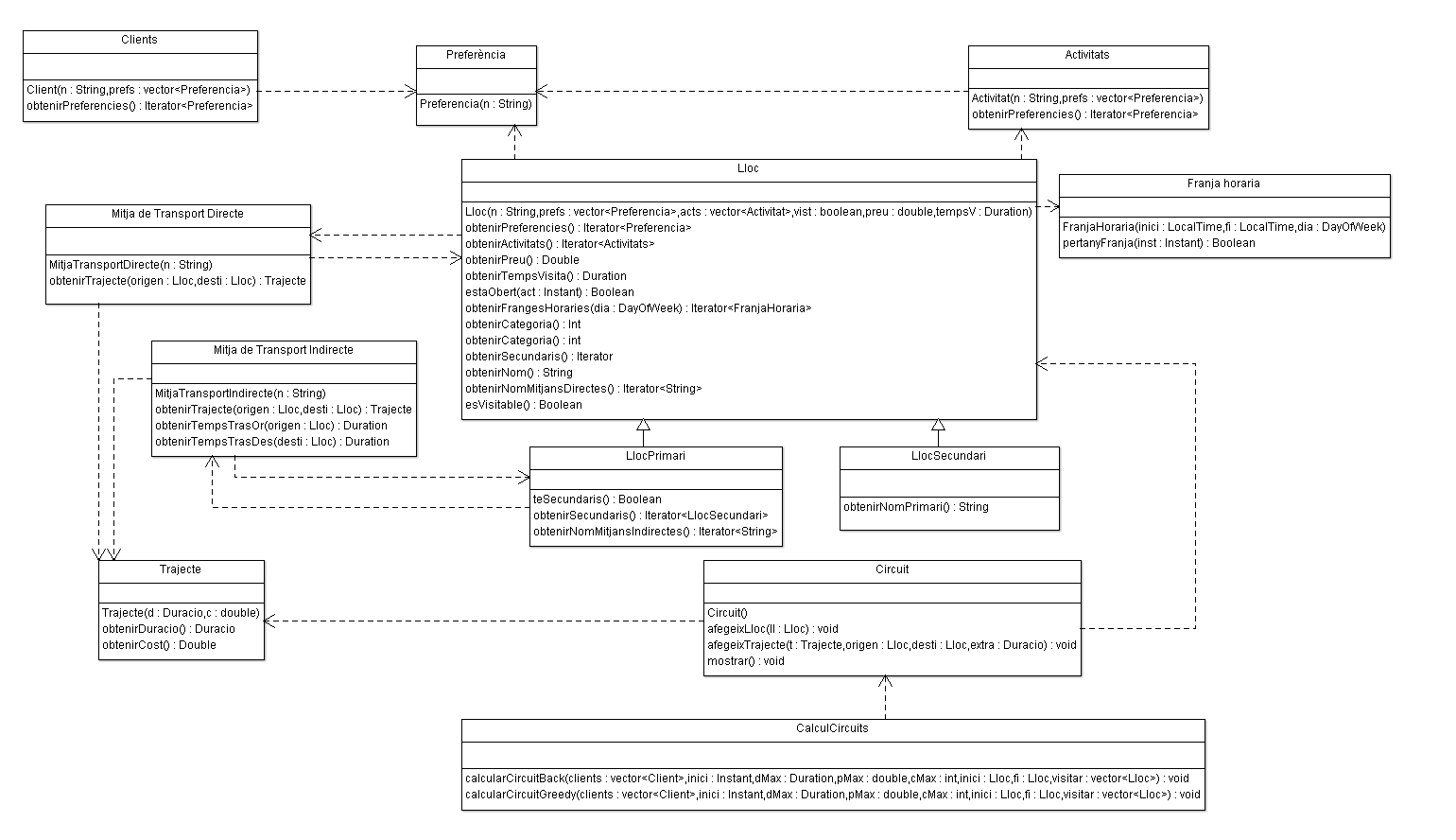
*Post: Calcula un circuit mitjançant backtraking*

void calcularCircuitBack(vector<Client> clients, Instant inici, Duration dMax, double pMax, int cMax, Lloc inici, Lloc fi, vector<Lloc> visitar)

*Pre: --*

*Post: Calcula un circuit mitjançant un algorisme voraç*

void calcularCircuitGreedy(vector<Client> clients, Instant inici, Duration dMax, double pMax, int cMax, Lloc inici, Lloc fi, vector<Lloc> visitar)

1. **Mòduls funcionals**