

# 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

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

Montilivi, Girona, 22 de març de 2017

## 1. Tipus Abstractes de dades

### Tipus Preferència

Descripció general: Conté informació d'una preferència.

*Pre: --*

*Post: Es crea una preferència de nom "n"*

Preferència(String n)

### Tipus Activitat

Descripció general: Conté informació d'una activitat, que conté preferències associades.

*Pre: --*

*Post: Es crea una activitat de nom "n" i preferències "prefs"*

Activitat(String n, vector<Preferencia> prefs)

*Pre: --*

*Post: Retorna un iterador a les preferències de l'activitat*

Iterator<Preferencia> obtenirPreferencies()

### 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, vector<Preferencia> prefs)

*Pre: --*

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

Iterator<Preferencia> obtenirPreferencies()

## Tipus FranjaHoraria

Descripció general: Conté una franja horària compresa entre dos hores determinades d'un dia.

*Pre: --*

*Post: Crea una franja horària a partir d'un dia i d'una hora d'inici i final (java.time)*

FranjaHoraria(LocalTime inici, LocalTime fi, DayOfWeek dia)

*Pre: --*

*Post: Retorna cert si l'instant de temps "inst" es troba dins de l'horari de la franja*

Boolean pertanyFranja(Instant inst)

## Tipus Lloc

Descripció general: Conté llocs visitables o allotjaments amb diferents característiques i/o activitats associades.

*Pre: --*

*Post: Es crea un lloc de nom "n", i amb les preferències i activitats "prefs" i "acts" respectivament. Visitable o no depenent de "vist" (Cert visitable, fals allotjament) i el preu del lloc visitable/allotjament (Per defecte 0). En cas de ser un lloc visitable, també s'entra el temps de visita recomanat (En format Duration de java.time).*

Lloc(String n, vector<Preferencia> prefs, vector<Activitat> acts,  
boolean vist, double preu=0, Duration tempsV=0)

*Pre: --*

*Post: Retorna el nom del lloc*

String obtenirNom()

*Pre: --*

*Post: Retorna cert si el lloc és visitable, fals altrament*

Boolean esVisitable()

*Pre: --*

*Post: Retorna un iterador que apunta a les preferències del lloc*

Iterator<Preferencia> obtenirPreferencies()

*Pre: --*

*Post: Retorna un iterador que apunta a les activitats del lloc*

Iterator<Activitat> obtenirActivitats()

*Pre: --*

*Post: Retorna el preu del lloc visitable (per persona) o del allotjament (per habitació doble), depenent si és un lloc visitable o un allotjament.*

double obtenirPreu()

*Pre: Lloc visitable*

*Post: Retorna el temps en format Duration (java.time) de visita recomanat*

Duration obtenirTempsVisita()

*Pre: Lloc visitable*

*Post: Retorna cert si el lloc visitable està obert en un determinat moment “act” (java.time)*

Boolean estaObert(Instant act)

*Pre: Lloc allotjament*

*Post: Retorna la categoria de l'allotjament (Enter que representa les estrelles, +gran +categoria)*

Int obtenirCategoria()

*Pre: --*

*Post: Retorna un iterador amb els noms dels transports directes disponibles en el lloc*

Iterator<String> obtenirNomMitjansDirectes()

## **Tipus LlocPrimari refina Lloc**

Descripció general: Conté un lloc primari. És un lloc (normalment ciutat) genèric que pot contenir altres llocs secundaris.

*Pre: --*

*Post: Retorna cert si el lloc primari conté llocs secundaris associats*

Boolean teSecundaris()

*Pre: --*

*Post: Retorna un iterador als llocs secundaris del lloc primari*

Iterator<LlocSecundari> obtenirSecundaris()

*Pre: --*

*Post: Retorna un iterador amb els noms dels transports indirectes disponibles en el lloc*  
Iterator<String> obtenirNomMitjansIndirectes()

## **Tipus LlocSecundari refina Lloc**

Descripció general: Conté un lloc secundari. És un lloc que està associat a un lloc primari.

*Pre: --*

*Post: Retorna el nom del LlocPrimari al que està associat*  
String obtenirNomPrimari()

## **Tipus Trajecte**

Descripció general: Conté informació del temps i el preu d'un trajecte realitzat per un mitjà de transport.

*Pre: --*

*Post: Crea un trajecte de duració "d" i cost "c"*  
Trajecte(Duracio d, double c)

*Pre: --*

*Post: Retorna la duració del trajecte*  
Duracio obtenirDuracio()

*Pre: --*

*Post: Retorna el cost del trajecte*  
Double obtenirCost()

## **Tipus MitjaTransportDirecte**

Descripció general: Conté informació del mitjà de transport directe i les dades del temps i del cost per a cadascun dels casos.

*Pre: --*

*Post: Crea un mitjà de transport directe de nom "n"*  
MitjaTransportDirecte(String n)

*Pre: --*

*Post: Retorna el trajecte entre dos llocs, si existeix. Altrament retorna null*

Trajecte obtenirTrajecte(Lloc origen, Lloc destí)

## **Tipus MitjaTransportIndirecte**

Descripció general: Conté informació del mitjà de transport indirecte i les dades del temps i del cost per a cadascun dels casos.

*Pre: --*

*Post: Crea un mitjà de transport indirecte de nom "n"*

MitjaTransportIndirecte(String n)

*Pre: --*

*Post: Retorna el trajecte entre dos llocs, si existeix. Altrament retorna null*

Trajecte obtenirTrajecte(Lloc origen, Lloc destí)

*Pre: --*

*Post: Retorna el temps de trasllat d'origen del mitjà de transport indirecte en funció del lloc*

Duration obtenirTempsTrasOr(Lloc origen)

*Pre: --*

*Post: Retorna el temps de trasllat de destí del mitjà de transport indirecte en funció del lloc*

Duration obtenirTempsTrasDest(Lloc destí)

## **Tipus Circuit**

Descripció general: Conté el conjunt de llocs i desplaçaments dels viatges.

*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 amb els llocs d'origen i destí, juntament amb una possible duració extra per el fet de transportar-se cap al mitjà de transport (si hi és)*

void afegeixTrajecte(Trajecte t, Lloc origen, Lloc destí, Duracio extra=0)

*Pre: --*

*Post: Mostra el circuit per la sortida estàndard*

void mostrar()

## **Tipus CalculCircuits**

Descripció general: Mòdul funcional que conté algorismes per el càlcul de circuits.

*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)

## 2. Mòduls funcionals

