Entrega

• Especificació detallada de les classes compartides, de fitxers i sistemes de directoris compartits, de formats d'entrada/sortida compartits, etc.

Asignaciones

1. Teclado

- Asignación
- Distancia
- Estadísticas

2. Planetas

- Elemento
- Adaptador del QAP
- CtrlPersistencia

3. Biblioteca

- Asignación
- Conjunto de nodos
- Nodos

Grupo Teclado

Distancia:

- **Explicación:** Representa la distancia que tiene cada nodo con todos los otros.
- **Cardinalidad:** La cardinalidad de la clase viene definida por el número de nodos del problema.
- **Atributos:** distancia
    - distancia (Integer): Distancia entre dos nodos *[1, 5, …].*
- **Asociaciones:** Es una clase asociativa entre dos nodos.
    - Puesto que una distancia se calcula sobre la combinación de todos los nodos con los demás, la asociación recursiva Distancia tiene multiplicidad * en ambos lados.
- **Funciones:**
    - calcularDistancia:
      Explicación: calcula la distancia entre dos nodos. Cada nodo viene definido por sus coordenadas “x” e “y” en relación con el conjunto de nodos.
      Parámetros: pos x, pos y, pos x’, pos y’
      Visibilidad: public
    - getDistancia:
      Explicación: consulta la distancia entre dos nodos de la tabla de distancias.
      Parámetros: elemento a, elemnto b
      Visibilidad: public

Estadísticas:

- **Explicación:** Ordena los objetos por afinidad. Ésta viene determinada por el usuario.
- **Cardinalidad:** La cardinalidad es 1 puesto que se sacan unas únicas estadísticas de unos mismos datos.
- **Atributos:** afinidades
    - afinidades(Pair<(Pair<elemento, elemento>), Integer>): tabla de afinidades.
- **Asociaciones:** La clase es una clase asociativa entre dos objetos.

    - Puesto que unas estadísticas se llevan a cabo sobre la combinación de todos los objetos con los demás, la asociación recursiva Objeto tiene multiplicidad * en ambos lados.
- **Funciones:**
    - getAfinidad:
      Explicación: consulta la afinidad entre dos objetos de la tabla de afinidades.
      Parámetros: objeto a, objeto b
      Visibilidad: public

Grupo Planetas

Elemento:

- **Explicación:** Representa el contenido de un nodo.
- **Cardinalidad:** La cardinalidad de la clase viene definida por el número de nodos del problema.
- **Atributos:**
    - id: integer que representa el identificador del elemento.
- **Asociaciones:** -
- **Funciones:**
    - elemento(): Constructora por defecto de la clase elemento.
    - elemento(id: integer): Constructora de la clase elemento.
    - setId(id: integer): Función que asigna un identificador a un elemento.
    - getId(): Función que devuelve el identificador de un elemento.

AdaptadorQAP:

- **Explicación:** Es la clase encargada de adquirir los datos necesarios y adaptarlos para el algoritmo que soluciona el problema cuadrático.
- **Cardinalidad:** La cardinalidad es 1 puesto que sólo será necesario un objeto para recoger los datos.
- **Atributos:**
    - tam (Integer): tamaño del problema

    - matrizAfinidad (int[ ][ ]): matriz de afinidad entre nodos
    - matrizDistancia (int[ ][ ]): matriz de distancias entre nodos
- **Asociaciones:**-
- **Funciones:**
    - adaptadorQAP(): Constructora por defecto de la clase
    - getTam(): Obtiene el tamaño de las matrices de afinidad y distancia.
    - getMatrizAfinidad(): Devuelve la matriz de afinidad.
    - getMatrizDistancia(): Devuelve la matriz de distancias.
    - setMatrizDistancia(d:int [ ][ ]): Asigna al parámetro implicito la matriz de distancias d.
    - setMatrizAfinidad(a:int [ ][ ]): Asigna al parámetro implícito la matriz de afinidades a.
    - setTam(int t): Asigna al parámetro implícito el tamaño t.

CtrlPersistencia

- **Descripción:** Es la clase encargada de gestionar la interacción con los datos almacenados.
- **Cardinalidad:** La cardinalidad es 1 puesto que sólo será necesario un objeto para interactuar con los datos.
- **Atributos:**
- **Asociaciones:** -
- **Funciones**:
    - ctrlPersistencia: Constructora por defecto de la clase.
    - leerJuegoDePrueba(String ruta)
      Explicación: Lee el fichero de prueba de nombre ruta
      Resultado: El controlador se guarda el tamaño de las matrices y las matrices de afinidad y distancia.
    - guardarResultado(int[] res, String ruta)
      Explicación: Guarda en el fichero de nombre ruta la asignacion res
      Resultado: Se crea un fichero de nombre ruta + “SOL.dat” que contiene el vector solución res.

Grup Biblioteca

Nodo

- **Descripción:** Representa un nodo
- **Atributos:**
    - x: int que representa la primera coordenada de un nodo dentro de un conjunto de nodos.
    - y: int que representa la segunda coordenada de un nodo dentro de un conjunto de nodos.
- **Funciones**:
    - nodo(x: int, y: int): Constructora de la clase nodo.
    - setX(x: int): Función que asigna una coordenada x a un nodo.
    - setY(y: int): Función que asigna una coordenada y a un nodo.
    - getX(): Función que devuelve la coordenada x d'un nodo.
    - getY(): Función que retorna la coordenada y d'un nodo.
- **Relaciones:**
    - De agregado a conjuntoNodos.

ConjuntoNodos

- **Descripción:** Conjunto de nodos
- **Atributos:**
    - nodos: Lista con los nodos del conjunto.
- **Funciones**:
    - conjuntoNodos(): Constructora por defecto de la clase conjuntoNodos.
    - addNodo(n: Nodo): Función que añade un nodo n al conjunto.
    - getNodo(n: int): Función que devuelve el nodo de la posición n.
    - deleteNodo(n: int): Función que elimina el nodo de la posición n.
- **Relaciones:**
    - De agregador a nodo.