

Implementación TAD Agenda

Jane Cahal

acción iniciar Agenda (sal agenda A)

principio
A = NULL
fin

Representación

tipo

Nodo = registro

persona per

puntero a nodo sig

preg

agenda = puntero a nodo.

acción añadir (els agenda A, persona p)

principio

// buscamos a ver si la persona esta en la agenda

agenda nuevo

nuevo = reservar(nodo)

nuevo = p mientras que (nuevo != NULL) hacer

si (strcmp(nombre(nuevo.per), nombre(p)) < 1) entonces

nuevo = (*nuevo).sig

SI-NO-SI (strcmp(nombre(nuevo.per), nombre(p)) == 1) entonces

copiar Persona (nuevo p, nuevo.per)

nuevo = (*nuevo).sig

SI-NO

nuevo = (*nuevo).sig

SI

fin

fin

si (agenda == NULL) hacer
pagar
si no, si agenda == p
salir
si no
añadir

funcion existe (agenda A, persona p) dev bool

principio
agenda Aux: bool esta = false

Aux = A

while mientras (aux != NULL AND esta = false) hacer

si personas_Iguales (aux.per, p) == true entonces
esta = true

si

aux = cest (aux).sig

fin

dev (esta)

accion consultar (ent agenda A, ent cadena nomb,
sal cadena thfo) // Presumimos que los nombres
son unicos

principio

bool esta = false

~~while~~ agenda Aux = A

mientras (Aux != NULL AND esta == false) hacer

si (strcmp (nombre (Aux.per), nomb) == 0) entonces
esta = true; thfo = telefono (aux.per);

si

aux = (*aux).sig

fin

accion borrar (els agenda A, ent cadena nomb)

// funcionamos igual que la copia, pero cuando
// encontramos el que buscamos nos lo "saltamos"

principio

aux = A

mientras que (aux != NULL ~~AND~~)

si (strcmp (nombre (A.per), nomb) == 0)

a = (*a).sig // Nos lo "saltamos"

si-no

añadir (aux, A.per)

a = (*a).sig

si

fin

acción listar / el agenda A)

Jaime Bohor

principio

agenda aux = A

mientras que (aux \neq null) hacer
escribir (nombre(aux.per), telefono(aux.per))

a = (*a.sig)

fin

Para todos los accionera tenemos en cuenta que al añadir añade en orden a las personas, por lo que no hace falta reordenarlas.