

SSL 2005.1 – Simulación de un Año Académico de SSL – Entrega #1 – TAD Actividad.

1.0.1.20050515

por José María Sola y Jorge Muchnik.

Este TAD puede ser extendido más adelante para cubrir las necesidades globales del TP.

Requisitos

- Haber aprobado TP #0
- Haber leído y comprendido el documento anterior: "SSL 2005.1 – Presentación General del Trabajo Práctico – Simulación de un Año Académico de SSL"

Objetivos

- **Especificar** el TAD Actividad.
- **Implementar** el TAD Actividad en una **Biblioteca**.
- **Probar** la implementación del TAD Actividad mediante una **aplicación de prueba**.
- **Capturar** las **salidas** de los **procesos de traducción** y de la **aplicación de prueba**.
- **Comenzar** a analizar y a diseñar el **TAD Calendario** y el programa de **Simulación**.

Sobre el TAD Actividad

Una **Actividad** es una **cuatro-upla (Día, Mes, Tipo de actividad, Descripción)** que representa una **actividad, tarea, evento ó suceso** relacionado con SSL. Un valor del tipo **CalendarioAcademico** estará formado por valores del tipo **Actividad**. El tercer componente, tipo de actividad, pertenece al conjunto {SinActividad, ClaseNormal, EvaluacionGrupalEscrita, EvaluacionGrupalOral, EvaluacionIndividual, ExamenFinal, InscripcionACurso, InscripcionAExamenFinal, PrimeraClase, PrimerRecuperatorio, SegundoRecuperatorio, UltimaClase}.

Operaciones

Creación

- **Creación.** Dado un Natural (día), una Cadena (mes), un TipoDeActividad y otra Cadena (descripción) produce una Actividad. La primer cadena representa el nombre de un mes codificado como tres letras. Dado que la descripción es una cadena de una longitud arbitraria, es muy conveniente utilizar reserva dinámica de memoria en la implementación.

Comparación

- **SonIguales** (predicado), dados dos valores del TAD Actividad indica si son iguales o no.

Consulta

- **EsDiaConActividad** (predicado), indica si el valor del TAD Actividad dado tiene asignado alguna actividad, tarea, evento ó suceso.

Acceso

- **Dia.** Dada una Actividad retorna un Natural.
- **Mes.**
 1. *NumeroMes*, dada una Actividad retorna un Natural.
 2. *MesCorto*, dada una Actividad retorna una cadena.
 3. *MesLargo*, dada una Actividad retorna una cadena.
- **TipoDeActividad.**
 1. *TipoDeActividad*, dada una Actividad retorna su TipoDeActividad.
 2. *TipoDeActividadComoCadena*, dada una Actividad retorna una cadena que describe el tipo de actividad.
- **Descripción.** Dada una Actividad retorna una Cadena.

Específicas de la implementación

- **Destrucción.** Dada una variable del TAD Actividad, libera los recursos previamente asignados.

Restricciones para la Implementación

```
typedef enum {
    SinActividad = 0,
    ClaseNormal,
    EvaluacionGrupalEscrita,
    EvaluacionGrupalOral,
    EvaluacionIndividual,
    ExamenFinal,
    InscripcionACurso,
    InscripcionAExamenFinal,
    PrimeraClase,
    PrimerRecuperatorio,
    SegundoRecuperatorio,
    UltimaClase
} TipoDeActividad;

static const char* meses[][2] = {
    {"Ene", "Enero"},
    {"Feb", "Febrero"},
    {"Mar", "Marzo"},

```

```
    {"Abr", "Abril"},
    {"May", "Mayo"},
    {"Jun", "Junio"},
    {"Jul", "Julio"},
    {"Ago", "Agosto"},
    {"Sep", "Septiembre"},
    {"Oct", "Octubre"},
    {"Nov", "Noviembre"},
    {"Dic", "Diciembre"}
};

typedef struct {
    int fecha;
    TipoDeActividad tipo;
    const char* descripcion;
} Actividad;
```

El campo fecha contendrá valores entre 1 y 366.

Identificadores para las Funciones que Implementan las Operaciones

- | | |
|---------------------------------|---|
| 1. Actividad_Crear | 7. Actividad_GetTipoDeActividadComoCadena |
| 2. Actividad_GetDia | 8. Actividad_GetDescripcion |
| 3. Actividad_GetNumeroMes | 9. Actividad_EsDiaConActividad |
| 4. Actividad_GetMesCorto | 10. Actividad_SonIguales |
| 5. Actividad_GetMesLargo | 11. Actividad_Destruir |
| 6. Actividad_GetTipoDeActividad | |

Conceptos de ANSI C Necesarios

- Tipos de datos de primitivos.
- Cadenas.
- Arreglos (matrices).
- Estructuras (struct).
- Enumeraciones (enum).
- Reserva dinámica de memoria (malloc, free)
- Funciones.

Sobre la Prueba

Se debe diseñar una aplicación de prueba con datos constantes, que verifique la corrección de todos los casos posibles. Utilice el concepto de partición de conjuntos, y pruebe los casos extremos y un caso representativo de cada partición.

Otras Restricciones

- Para la implementación debe utilizarse el lenguaje y los elementos de la biblioteca estándar especificados en ANSI C [M3].
- Luego de que la biblioteca y la aplicación de prueba hayan sido construidas con la herramienta de desarrollo elegida por el equipo, deben ser verificadas mediante la herramienta de desarrollo “Borland C++ Compiler 5.5 with Command Line Tools” (BCC32) configurada tal como dicta la cátedra [TP #0].
- Los procesos de compilación con BCC32 no deben emitir warnings (mensajes de advertencia) ni, por supuesto, mensajes de error.
- Los identificadores de las funciones que implementan las operaciones, de las variables, tipos y demás elementos así como los nombres de los archivos deben ser los indicados a lo largo de este enunciado.
- Cualquier decisión que el equipo tome sobre algún punto no aclarado en el enunciado debe ser agregada como hipótesis de trabajo.

Sobre La Entrega

Tiempo

- Siete (7) días después de analizado el enunciado en clase.

Forma

El trabajo debe presentarse en hojas **A4 abrochadas** en la esquina superior izquierda. En el encabezado de cada hoja debe figurar el **título** del trabajo, el código de **curso** y los **apellidos** de los integrantes del equipo. Las hojas deben estar enumeradas en el pie de las mismas con el formato “**Hoja n de m**”.

El código fuente de cada componente del TP debe comenzar con un comentario encabezado, con todos los datos del equipo de trabajo: **curso**; **legajo**, **apellido** y **nombre** de cada integrante del equipo y **fecha de última modificación**. La fuente a utilizar en la impresión debe ser una fuente de ancho fijo (e.g. Courier New, Lucida Console).

1. TAD Actividad

- 1.1. **Especificación.** Especificación completa, extensa y sin ambigüedades de los valores y de las operaciones del TAD.
- 1.2. **Implementación**
 - 1.2.2. Biblioteca que implementa el TAD
 - 1.2.2.1. Listado de código fuente del archivo encabezado, **parte pública**, **Actividad.h**.
 - 1.2.2.2. Listado de código fuente de la definición de la Biblioteca, **parte privada**, **Actividad.c**.
 - 1.2.3. **Salidas.** Captura impresa de la salida del proceso de traducción (BCC32 y TLIB). Utilizar fuente de ancho fijo.

2. Aplicación de Prueba

- 2.1. **Código Fuente.** Listado del código fuente de la aplicación de prueba, **ActividadAplicacion.c**.
- 2.2. **Salidas**
 - 2.2.1. Captura impresa de la salida del proceso de traducción (BCC32). Utilizar fuente de ancho fijo.
 - 2.2.2. Captura impresa de las salidas de la aplicación de prueba. Utilizar fuente de ancho fijo.

3. Copia Digitalizada

CD ó disquette con copia de solamente los 3 archivos de código fuente (Actividad.h, Actividad.c y ActividadAplicacion.c). No se debe entregar ningún otro archivo.

4. Formulario de Seguimiento de Equipo.