Práctica 1: Elementos básicos de los lenguajes de programación

El objetivo de esta práctica es identificar los elementos fundamentales de los lenguajes de programación: nombres, marcos de activación, bloques de alcance, administración de memoria, expresiones, comandos, control de secuencia como lo es; selección iteración y recursión, subprogramas, y tipos de datos. Identificar estos conceptos en la aplicación propuesta para esta práctica. Entregar aquí en classroom el reporte en PDF de la práctica, generado a partir del contenido en su portafolio. No olviden añadir al reporte el enlace a su portafolio y al sitio estático, donde este reporte es la segunda entrada.

Nombres

Funciones: addBook, displayBooksRecursive, findBookById, removeBookById, addMember, findMemberById, issueBook, displayMembers, displayIssuedBooks, main.

Variables: new_book, bookCount, memberCount, library, members, bookID, memberID, bookFound, memberFound, choice, current_book, current_member, i, issued_count, issued_books, title, author, genre, id, name, MAX_TITLE, MAX_AUTHOR, MAX_NAME, MAX_ISSUED_BOOKS.

Tipos definidos: genre_t, book_t, member_t.

Marcos de Activación

addBook, displayBooksRecursive, findBookById, removeBookById, addMember, findMemberById, issueBook, displayMembers, displayIssuedBooks, main.

Bloques de Alcance

addBook, displayBooksRecursive, findBookById, removeBookById, addMember, findMemberById, issueBook, displayMembers, displayIssuedBooks, main, if, for, while, switch, case.

Administración de Memoria

malloc, realloc, free, new_book, library, members, issued_books, addBook, removeBookById, addMember, issueBook.

Expresiones

new_book = (book_t *)malloc(sizeof(book_t)), (*count)++, new_book->title[strcspn(new_book->title, "\n")] = '\0', library[*count] = new_book, memberFound->issued_books[memberFound->issued_count++] = bookID, bookFound = findBookById(library, bookCount, bookID), memberFound = findMemberById(members, memberCount, memberID), for (int i = 0; i < count; i++), if (bookFound && memberFound), printf("Book ID: %d\n", library[i]->id), realloc(*library, (*count + 1) * sizeof(book_t *)), free(library[i]).

Comandos (Sentencias)

new_book = (book_t *)malloc(sizeof(book_t)), (*count)++, library[*count] = new_book, new_book->title[strcspn(new_book->title, "\n")] = '\0', if (!new_book) return, for (int i = 0; i < count; i++), bookFound =

findBookById(...), memberFound = findMemberById(...), memberFound->issued_books[memberFound->issued_count++] = bookID, realloc(...), free(...), switch (choice), case 1: addBook(...), case 2: removeBookById(...), case 3: displayBooksRecursive(...), case 4: addMember(...), case 5: issueBook(...), case 6: displayMembers(...), case 7: displayIssuedBooks(...), case 8: break, default: printf(...), return.

Control de Secuencia

if, else, for, while, switch, case, default, return.

Subprogramas (Funciones)

addBook, displayBooksRecursive, findBookById, removeBookById, addMember, findMemberById, issueBook, displayMembers, displayIssuedBooks, main.

Tipos de Datos

int, char, void, book_t, member_t, genre_t, enum, struct, size_t, float, realloc, malloc, free.