

\*\*\*\*\*\*\*\*\*\*\*\*\* pub.h\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#if !defined PUB

#define PUB

typedef enum {ptBook, ptMagazine} PubType;

typedef enum {ctHardcover, ctPaperback} CoverType;

typedef struct {

 PubType type;

 char title[100];

 int npages;

 CoverType cover;

 float price;

 union{

 struct{

 int issn;

 char vol;

 char issue;

 } details;

 unsigned int isbn;

 } more;

} Publication;

typedef struct {

 int count;

 Publication\* items;

} PubList;

Publication readpub();

int findpub(Publication\* , PubList );

PubList getallbooks(PubList );

void printpublist(PubList);

#endif

\*\*\*\*\*\*\*\*pub.c\*\*\*\*\*

#include <stdio.h>

#include <stdlib.h>

#include "pub.h"

Publication readpub(){

 Publication pub;

 char c;

 int v;

 printf("\nWhat type of publication (Book:b , Magazine:m)?");

 scanf("%c\n", &c);

 pub.type = c=='b'? ptBook:ptMagazine;

 printf("\nEnter the title:");

 gets(pub.title);

 printf("\nEnter # of pages:");

 scanf("%d\n", &(pub.npages));

 printf("\nWhat type of cover (Hardcover:h , Paperback:p)?");

 scanf("%c\n", &c);

 pub.cover = c=='h'? ctHardcover:ctPaperback;

 printf("\nEnter the price:");

 scanf("%f\n", &(pub.price));

 if (pub.type == ptBook) {

 printf("\nEnter the ISBN:");

 scanf("%u\n", &(pub.more.isbn));

 }

 else {

 printf("\nEnter the ISSN:");

 scanf("%d\n", &(pub.more.details.issn));

 printf("\nEnter the volume #:");

 scanf("%d\n", &v);

 pub.more.details.vol = (char)v;

 printf("\nEnter the issue #:");

 scanf("%d\n", &v);

 pub.more.details.issue = (char)v;

 }

 return pub;

}

int findpub(Publication\* pub, PubList pl){

 int i;

 for (i=0; i < pl.count; i++)

 if (pl.items[i].type == pub->type){

 if (pub->type == ptBook){

 if (pl.items[i].more.isbn == pub->more.isbn)

 return i;

 }

 else {

 if (pl.items[i].more.details.issn == pub->more.details.issn &&

 pl.items[i].more.details.vol == pub->more.details.vol &&

 pl.items[i].more.details.issue == pub->more.details.issue)

 return i;

 }

 }

 return -1;

}

PubList getallbooks(PubList pl){

 PubList result;

 Publication\* p;

 int i;

 result.count = 0;

 for (i=0; i<pl.count; i++)

 if (pl.items[i].type == ptBook) result.count++;

 result.items = (Publication\*) malloc(result.count\*sizeof(Publication));

 if ((p = result.items)){

 for (i=0; i<pl.count; i++){

 if (pl.items[i].type == ptBook)

 \*p++ = pl.items[i];

 }

 }

 return result;

}

void printpublist(PubList pl){

 int i;

 for (i=0; i < pl.count; i++){

 printf("\n[%s]\n===========\n", pl.items[i].type==ptBook?"Book":"Magazine");

 printf("%s\n", pl.items[i].title);

 printf("%d pages @ S.R.%.2f\n", pl.items[i].npages, pl.items[i].price);

 if (pl.items[i].type == ptBook) printf("ISBN: %010u\n", pl.items[i].more.isbn);

 else printf("ISSN: %08d Vol. %d(%d)\n", pl.items[i].more.details.issn,

 pl.items[i].more.details.vol,

 pl.items[i].more.details.issue);

 }

}

\*\*\*\*\*\*\*\*\*\*\*\*\*test.c\*\*\*\*\*\*\*\*\*\*

#include <stdio.h>

#include <stdlib.h>

#include "pub.h"

int main(){

 PubList pl, bl;

 Publication p;

 int i;

 float sum=0;

 printf("How many publication to process?");

 scanf("%d\n", &(pl.count));

 if ((pl.items = (Publication\*) malloc(pl.count\*sizeof(Publication)))) {

 for (i=0; i<pl.count; i++) {

 do {

 p = readpub();

 } while (findpub(&p, pl) != -1);

 pl.items[i] = p;

 printf("\n--------------------");

 }

 bl = getallbooks(pl);

 printpublist(bl);

 for (i=0; i < bl.count; i++)

 sum += bl.items[i].price;

 printf("\nTotal price for all books: %.2f", sum);

 free(pl.items);

 free(bl.items);

 return 0;

 }

 return -1;

}

\*\*\*\*\*\* data.txt\*\*\*\*\*\*\*\*

4

b

The C Programming Language

475

p

140.0

131103628

m

From data types to object types

26

p

25.0

8630593

26

1

m

Automated Quality Assessment of Metadata across Open Data Portals

28

p

25.0

19361955

8

1

b

The Linux Programming Interface: A Linux and UNIX System Programming Handbook

1552

h

200.0

1593272200