2021年1月12日

システム開発実習　詳細設計書

1/6ページ

詳細設計書

所属：人事部

張　文コン

【プログラム設計書】

0-0.【main関数（int型）】

「引数」

なし

「機能」

★初期化

データベース（schedule.db）を“ｒ”モードで開く。

データ（予定、締切り、削除判定）を一時保存配列（schedule[][3]）へ格納：

while (fscanf(fp\_s, "%s%s%s", todo, deadline, isDeleted) != EOF) {

schedule[todoNum][0]のメモリを動的確保し、予定内容を格納。

schedule[todoNum][1]のメモリを動的確保し、締切りを格納。

schedule[todoNum][2]のメモリを動的確保し、削除判定を格納。

総予定数(todoNum)+1;

If(削除判定（isDeleted）が“１”の場合)｛

削除予定数(deletedNum)+1;

｝

｝

データベース（schedule.db）を閉じる。

while(1){

mode 1-1 (mainPage()）予定一覧を出力

ユーザー入力(select)：メニュー選択

switch(select[0])｛

case ‘a’: mode 2-1 (add())　予定追加;

case ‘c’: mode 4-1 (complete())　完成（削除）予定一覧;

case ‘q’: mode 5-1 (save()) 保存＆終了;

default:

if(selectは予定No.)｛

mode 3(edit (select)) 予定編集;

｝else{

“無効命令” ;

}

｝

}

[戻り値]

return 0;

2021年1月12日

システム開発実習　詳細設計書

2/6ページ

1-1．【mainPage関数(void型)】

「引数」

なし

「機能」

★予定一覧

画面をリセット

for(int i =0; i<総予定数; i++){

if(schedule[i][2]は“0”(==未削除)){

FORMAT2を用い、予定No.、予定内容(schedule[i][0])と締切り( schedule[i][1]))を画面へ出力。

schedule中の添字iと画面出力時の予定No.（変数num）をリンクし、配列indexへ格納。Index[num -1 ] = i

}

}

メニュー（a:予定追加　(No.)1～未削除予定数:予定編集　c:完成予定一覧　q:保存＆終了）を画面へ出力

[戻り値]

なし

2-1．【add関数(void型)】

「引数」

なし

「機能」

★予定追加

画面をリセット

ユーザー入力：予定内容

Schedule[todoNum][0] のメモリを動的確保し、予定内容を格納。

do{

ユーザー入力：締切り

}while(mode 7-1 (!deteFormateCheck(締切り) 書式チェック)

締切り書式変換: xx/oo => xxoo (ソートのため)

Schedule[todoNum][1]のメモリを動的確保し、締切りを格納。

Schedule[todoNum][2]のメモリを動的確保し、削除判定（“０”）格納

総予定数（todoNum）＋１

[戻り値]

なし

2021年1月12日

システム開発実習　詳細設計書

3/6ページ

３-０．【edit関数(void型)】

「引数」

int num (予定のNo.)

「機能」

★予定編集

配列indexを用い、num（予定No.）からscheduleの添字iを取得。int i == index[num-1];

画面をリセット

FORMAT2を用い、予定内容(schedule[i][0])と締切り( schedule[i][1]))を画面へ出力。

メニューを画面へ出力（1:修正　　2:完成/削除　　3:キャンセル）

while(1){

ユーザー入力(select)：メニュー番号

If(select[0] == ‘1’){

（mode 3-1 予定修正）

ユーザー入力（schedule[i][0]）：新たな予定内容

do{

ユーザー入力(newdeadline)：新たな締切り

}while(mode 7-1 (!deteFormateCheck(newdeadline)書式チェック)

締切り（newdeadline）書式変換: xx/oo￥0 => xxoo￥0 (ソートのため)

締切りを格納。

} else if(select[0] == ‘2’){

（mode 3-2 予定削除）

予定をロジック削除（schedule[i][2]を“０” =>　“１”）

indexからscheduleの添字iを削除。

削除予定数(deletedNum)+1;

}

}

}

[戻り値]

なし

2021年1月12日

システム開発実習　詳細設計書

4/6ページ

４-１．【complete関数(void型)】

「引数」

なし

「機能」

★完成（削除）予定一覧

画面をリセット

for (int i = 0; i < 総予定数; i++)

{

if (schedule[i][2]は"1"（ロジック削除項目）) {

FORMAT2を用い、予定No.、予定内容(schedule[i][0])と締切り( schedule[i][1]))を画面へ出力。

}

}

[戻り値]

なし

5-１．【save関数(void型)】

「引数」

なし

「機能」

★schedule配列中に臨時保存した予定をデータベース（schedule.db）へ出力してから、プログラム終了。

データベースファイル（schedule.db）を“ｗ”モードで開く。

（追加機能）mode 6-1(sort()) データを締切り昇順で整列。

schedule配列中のデータをデータベース（schedule.db）へ出力。（format:”%s　%s　%s　“）。

データベースファイル（schedule.db）を閉じる。

exit(0)でプログラム終了

[戻り値]

なし

2021年1月12日

システム開発実習　詳細設計書

5/6ページ

6-１．【sort関数(void型)】

「引数」

なし

「機能」

★配列schedule中のデーターを締切り(schedule[i][1])昇順でソートする。

締切りが「----」（締切り無し）の場合、「9999」へ書き換え。

バブルソート。

「9999」を「----」へ書き換え。

[戻り値]

なし

7-１．【dateFormatCheck関数(int型)】

「引数」

char\* dl

「機能」

★締切りのフォーマットをチェック.(フォーマット:xx/xx)

int flag = 1　判別フラッグ

判別１：締切り長さを５桁。（否の場合flag=0）

判別2：前の２桁が”月”の型と一致（01～12）。（否の場合flag=0）

判別2：後ろの２桁が”日”の型と一致（01～31）。（否の場合flag=0）

(注：28日と30日のみの月があるが、単独判別しない)

「--/--」（締切り無し）の場合、flag = 1;

[戻り値]

return flag

2021年1月12日

システム開発実習　詳細設計書

6/6ページ

【フローチャート】

プログラム設計書で定めた関数の動きをフローチャートで図示したもの。

別ファイル「【詳細設計】フローチャート」に該当

【画面遷移図】

ユーザー視点のシステム上の画面遷移を図示したもの。

別ファイル「【詳細設計】画面遷移図」に該当

【変数一覧表】

プログラム設計書で使用された変数の使用箇所や意図等を記載した文書。

別ファイル「【詳細設計】変数一覧表」に該当