Програмни среди, КСТ

Лабораторни упражнения

З А Д А Ч А № 1.1

Идентификация на елементи на програмната среда – Win32 CRT конзолно приложение

Чрез предоставения минималистичен (“скелетен“) програмен проект от вида Win32 Console Application, или т.нар. CRT базирано Win32 конзолно приложение, да се изпълнят посочените по-долу задачи:

I. При наличен програмен проект :

1. Да се построи и да се тества работата на програмата.
2. Да се разгледат внимателно всички файлове (модули) с изходен код в състава на проекта и :
   2.1 Да се изгради хипотеза, а оттам и обяснение, за ролята на всеки един от изходните модули в състава на проекта ! Особено важно е това да се направи за модулите *stdafx.h* и *stdafx.c* .
   2.2 Да се проучи защо името на входната точка на програмата не отговаря на строгото изискване на програмния език, на който е съставена програмата и как е възможно въпреки това тя да е коректна.
3. След много внимателен анализ на съдържанието на изходните модули, да се идентифицират всички използвани елементи на обкръжението, в което работи програмата ! В случая те са :
   3.1 типове данни, нехарактерни за програмния език.
   3.2 макроси за препроцесора.
4. За всеки от идентифицираните елементи на обкръжението да се установи header-ния модул, в който се намира обявяването на идентификатора на елемента, както и :
   4.1 За всеки от типовете данни – декларацията му и на какъв друг тип данни той е синоним. Ако еквивалентния синоним отново или самият той е елемент на обкръжението или съдържа съставен компонент, който е елемент на обкръжението, то да се продължи рекурсивно издирването до достигане на еквиваленти единствено измежду вградените в програмния език типове.
   4.2 За всеки от макросите на препроцесора – обявяването на макроса с *#define*. Ако в обявяването му се съдържа друг макрос, то издирването продължава рекурсивно.
5. Резултатите от работата по т. I.3 и т. I.4 да се попълнят прегледно в таблица, напр. в предоставения благородно и наготово шаблон от приложения документ.

II. Да се разработи програмно решение на задача по-долу, което да се вложи в скелетния програмен проект :

1. Като първи параметър от командния ред да се подава път до текстов файл с формат и съдържание същите, като приложения текстов файл. Може да се ползва и самият приложен файл, от мен да мине !
2. Програмата отваря файла като текстов и ред по ред прочита съдържанието му в структури от данни в оперативната памет. Данните от всеки ред на файла да се разположат в самостоятелна съставна структура от данни. Всички прочетени редове да се организират в едномерен масив от такива еднотипни структури от данни. NB! : Факултетният номер да се съхранява задължително в член на съставната структура от целочислен тип ! Hint: Размерът на всеки масив в този живот трябва да се определи предварително, преди създаването му и всякаква възможна работа с него !
3. Според стойността на втори параметър от командния ред, програмата извършва едно от следните неща :
   3.1 Чрез използване (задължително!) на библиотечната функция *qsort* се сортира масива от съставни структури от данни по стойността на факултетния номер, в asc ред, или
   3.2 Чрез използване (задължително!) на библиотечната функция *qsort* се сортира масива от съставни структури от данни по група в desc ред (първично) и на фамилно име в asc ред (вторично).
4. Съдържанието на сортирания масив да се изведе на конзолата ред по ред във вид, наподобяващ формата на текстовия файл.
5. При завършването си програмата да освободи всякаква динамично резервирана памет, ако има такава !

III. След окончателно настройване на програмното решение по т.II :

1. Да се извърши отново анализа, описан в т. I.3, но вече върху модифицирания изходен програмен код. Този път, обаче, да се идентифицират елементи на програмната среда от всичките три категории :

   1.1 входни точки на програмната среда – външни за програмата функции.

   1.2 типове данни, нехарактерни за програмния език.

   1.3 макроси на препроцесора.

2. Да се добавят в таблицата установените нови използвани елементи на обкръжението, както са описани в т. I.3 и т. I.4, и то така, че да си личи, че са новоустановени. За всяка категория елементи да се посочи :

   2.1 за библиотечна функция (входна точка) – прототип на функцията.

   2.2 за тип данни – декларацията му.

   2.3 за макрос за препроцесора – обявяването на макроса с *#define*.

Съдържание на протокола : списък на модулите с изходен код, с обяснение за ролята на всеки в състава на проекта ; прототип на функцията, която се явява входна точка на програмата, с обяснение за името ѝ и за формалните ѝ параметри ; листинг на изходния програмен код след имплементиране на решението на задачата по т.II ; списък с идентифицираните елементи на обкръжението, с установения декларативен еквивалент за всеки, според указаното в т. I.4 и т. III.2 .

Информационни източници :

fscanf, _fscanf_l, fwscanf, _fwscanf_l
https://docs.microsoft.com/en-us/cpp/c-runtime-library/reference/fscanf-fscanf-l-fwscanf-fwscanf-l?view=msvc-170

fscanf_s, _fscanf_s_l, fwscanf_s, _fwscanf_s_l
https://docs.microsoft.com/en-us/cpp/c-runtime-library/reference/fscanf-s-fscanf-s-l-fwscanf-s-fwscanf-s-l?view=msvc-170

fseek
https://learn.microsoft.com/en-us/cpp/c-runtime-library/reference/fseek-fseeki64?view=msvc-170

qsort
https://docs.microsoft.com/en-us/cpp/c-runtime-library/reference/qsort?view=msvc-170

qsort_s
https://docs.microsoft.com/en-us/cpp/c-runtime-library/reference/qsort-s?view=msvc-170