|  |  |  |
| --- | --- | --- |
|  | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования FPMI_ngtu_neti_rgb_polya«Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра прикладной математики | | |
| Практическое задание № 4 | | |
| по дисциплине «Структуры данных и алгоритмы» | | |
| **Управление таблицами** | | |
|  | | |
|  | Бригада 3 | лезнёв артём |
| Группа ПМ-15 | черновская мария |
| Вариант 2а | зуев андрей |
|  |  |
|  |  |
|  |  |
| Преподаватели | Тракимус Юрий Викторович |
|  |  |
| Новосибирск, 2022 | | |

1. **Условие задачи**

В файле WORK содержатся результаты работы цеха за день. Элемент файла включает: шифр изделия (8-символьный код), наименование изделия, количество (штук). Построить таблицу, содержащую результаты работы за день, считая ключом шифр изделия. Элемент таблицы имеет ту же структуру, что и элемент файла. Содержащаяся в файле информация с равными ключами должны быть помещена в таблицу один раз с общим количеством штук изделия. Организовать таблицу как: неупорядоченную.

1. **Анализ задачи**

Входные данные: **WORK.txt** - таблица с результатами работы цеха за день.

Выходные данные: сортированная таблица с данными по каждому изделию (шифр, название, количество изготовленных штук).

Структуру входной таблицы с данными определим следующим образом:

|  |  |  |
| --- | --- | --- |
| **Шифр** | **Название** | **Количество** |
| 00000001 | Название1 | 30 |
| 00000002 | Название2 | 20 |
| 00000001 | Название1 | 40 |

Итоговую таблицу будем сортировать по 8-значному коду.

Обработка входного файла:

1. Считываем код, название и количество произведенных штук одного изделия.

2. Проверяем код на уникальность.

3. Если код уникален, добавляем данные изделия в таблицу, иначе при повторном коде добавляем к количеству произведенных штук изделия количество при повторном коде.

Сортировка таблицы происходит по следующему алгоритму:

Функция **insert\_or\_assign** добавляет новый элемент **item** в таблицу или возвращает указатель на старый, если он уже есть в таблице.

1. Считываются данные с входного файла.

2. Происходит проверка таблицы на полноту.

2. Далее идет поиск элемента с таким же ключом.

3. Если поиск не удался элемент помещается в массив.

4. Иначе размер таблицы уменьшается на единицу.

5. К количеству изготовленных штук изделия прибавляется текущее количество. (Если элемент уникальный, количество изготовленных изделий такое же как в входном файле. Если не уникальный, то количества изготовленных штук изделия складываются).

Таблица после подсчета всех голосов:

|  |  |  |
| --- | --- | --- |
| **Шифр** | **Название** | **Количество** |
| 00000001 | Название1 | 70 |
| 00000002 | Название2 | 20 |

После сортировки таблица выводится в консоль.

1. **Текст программы**

**product.h**

#ifndef H\_PRODUCT

#define H\_PRODUCT

#include <windows.h>

const char CODE\_SIZE = 9, NAME\_SIZE = 10;

struct product

{

char code[CODE\_SIZE] = { },

name[NAME\_SIZE] = { };

ULONGLONG amount = 0;

};

#endif

**table.h**

#ifndef H\_TABLE

#define H\_TABLE

#include <stdio.h>

#include <windows.h>

#include "product.h"

const char Nmax = 10;

struct table

{

private:

ULONGLONG size = 0;

product \*data[Nmax] = { };

bool index\_of(product \*item, ULONGLONG &i);

public:

/// <summary>

/// Добавляет новый элемент item в таблицу или возвращает указатель на

старый, если он уже есть в таблице

/// </summary>

product \*insert\_or\_assign(product \*item);

/// <summary>

/// Выводит содержимое таблицы на консоль

/// </summary>

void print();

/// <summary>

/// Строит таблицу по содержимому файла f

/// </summary>

void input(FILE \*f);

};

#endif

**table.cpp**

#include <stdio.h>

#include <windows.h>

#include "item.h"

#include "table.h"

#include "product.h"

bool table::index\_of(product \*item, ULONGLONG &i)

{

i = 0;

for (bool equals = true; i < size; i++) // Поиск элемента с таким же ключом

{

equals = true;

for (char j = 0; j < sizeof data[i]->code && equals;

equals = data[i]->code[j] == item->code[j], j++);

if (equals)

return true;

}

return false;

}

product \*table::insert\_or\_assign(product \*item)

{

if (size == Nmax)

return NULL;

ULONGLONG i = 0;

if (!this->index\_of(item, i))

{

data[size++] = item;

return NULL;

}

else return data[i];

}

void table::input(FILE \*f)

{

char c;

do

{

product \*p = new product();

fscanf\_s(f, "%s %s %lli", &p->code, (UINT)sizeof p->code,

&p->name, (UINT)sizeof p->name,

&p->amount);

product \*fp = this->insert\_or\_assign(p);

if (fp)

fp->amount += p->amount;

} while (fscanf\_s(f, "%c", &c, (UINT) sizeof c) != EOF);

}

void table::print()

{

printf\_s("%6s %11s %12s\n", "Шифр", "Название", "Количество");

for (ULONGLONG i = 0; i < size; i++)

{

product \*p = data[i];

printf\_s("%s %10s %11lli\n", p->code, p->name, p->amount);

}

}

**main.cpp**

#include <conio.h>

#include <stdio.h>

#include <locale.h>

#include "table.h"

int main()

{

setlocale(0, "");

FILE \*f = NULL;

errno\_t err = fopen\_s(&f, "WORK.txt", "r");

if (f)

{

table t;

t.input(f);

fclose(f);

t.print();

}

else perror("Не удалось открыть файл с данными.");

return 0 \* \_getch() + err;

}

1. **Набор тестов**

|  |  |  |
| --- | --- | --- |
| **№** | **Входные данные** | **Назначение** |
| 1 | 11111111 Toy 20  98874300 Book 100  87654321 Dress 209  33091265 Pencil 41 | Проверка программы на все уникальные коды. |
| 2 | 32345235 Short 899  00120034 Computer 34  67088176 Table 125  33091265 Window 211  00120034 Computer 111  32345235 Short 96  67088176 Table 11  33091265 Window 1  33091265 Window 78 | Проверка программы на повторяющиеся коды всех изделий. |
| 3 | 00000001 Cup 3276  00120034 Knife 3  00120034 Knife 3  10203040 Spoon 13098  20040030 Fork 2001  00120034 Knife 3  33557799 Plate 111  00120034 Knife 3  00120034 Knife 3 | Проверка программы на один много раз повторяющийся код |

1. **Результаты работы программы**

|  |  |
| --- | --- |
| **№** | **Ввод/вывод программы** |
| 1 | Шифр Название Количество  11111111 Toy 20  98874300 Book 100  87654321 Dress 209  33091265 Pencil 41 |
| 2 | Шифр Название Количество  32345235 Short 995  00120034 Computer 145  67088176 Table 136  33091265 Window 290 |
| 3 | Шифр Название Количество  00000001 Cup 3276  00120034 Knife 15  10203040 Spoon 13098  20040030 Fork 2001  33557799 Plate 111 |