МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

КАФЕДРА ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ ТЕХНОГОГИЙ

Программное обеспечение для управления базой данных

«Ведомость на выдачу стипендии в ВУЗе»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ

ПО ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРВАНИЯ»

КР.ПО6.33203 - 01 81 00

Руководитель М. В. Хацкевич

Выполнил Мартынович Д. М.

Консультант

по ЕСПД М. В. Хацкевич

Брест 2021

Форма № 17

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Факультет | | ЭИС | | | | | | | | Кафедра | | | ИИТ | |
|  | |  | | | | | | | |  | | |  | |
| УТВЕРЖДАЮ | | | | | | | | | | | | | | |
| Зав. кафедрой | | | |  | | | | | |  | | | | |
|  | | | | (подпись) | | | | | |  | | | | |
| « 1 » | 03 | | | | | | | | 2021 г. | | | | | |
|  |  | | | | | | | |  | | | | | |
| ЗАДАНИЕ | | | | | | | | | | | | | | |
| по курсовой работе | | | | | | | | | | | | | | |
| Студенту | | Мартыновичу Даниилу Михайловичу | | | | | | | | | | | | |
| 1. Тема проекта | | | Программное обеспечение для управления базой данных  «Ведомость на выдачу стипендии в ВУЗе». | | | | | | | | | | | |
| **Вариант №9**  Массив данных – «Ведомость на выдачу стипендии в ВУЗе».   1. фамилия студента (char[]); 2. сумма стипендии (longint); 3. средний балл студента (float);   Выполнить из общего списка запросы с номерами 3, 7 и 8. | | | | | | | | | | | |
|  | | | | | | | | | | | | | | |
| 2. Сроки сдачи студентом законченного проекта | | | | | | | | | | | 30.05.2021 | | | |
| 3. Исходные данные к проекту | | | | | | | |  | | | | | | |
| Среда программирования на языке С\С++ | | | | | | | | | | | | | | |
| Выбрать предметную область (в соответствии с вариантом) для базы данных и | | | | | | | | | | | | | | |
| доработать структуру для описания отдельных записей базы данных. Выбранная | | | | | | | | | | | | | | |
| структура должна иметь не менее пяти полей (элементов) двух и более типов, | | | | | | | | | | | | | | |
| включая пользовательский тип union и enum. | | | | | | | | | | | | | | |
| Работа содержит описание разработанного студентом программного обеспечения по | | | | | | | | | | | | | | |
| обработке заданного массива структур. Для всех вариантов обеспечить реализацию | | | | | | | | | | | | | | |
| следующих запросов к заданному массиву структурированной информации: | | | | | | | | | | | | | | |
| 1. Ввод информации из текстового файла в массив указателей на записи | | | | | | | | | | | | | | |
| 1. Добавление новых элементов в конец массива | | | | | | | | | | | | | | |
| 1. Просмотр всех элементов массива | | | | | | | | | | | | | | |
| 1. Вывод информации из массива в текстовый файл | | | | | | | | | | | | | | |
| 1. Корректировка полей выбранного элемента | | | | | | | | | | | | | | |
| 1. Удаление выбранного элемента | | | | | | | | | | | | | | |
| Необходимо обеспечить в соответст­вии с заданием варианта реализацию еще | | | | | | | | | | | | | | |
| трех запросов: | | | | | | | | | | | | | | |
| 1. Удаление элементов по условию (поле < или > заданного значения) | | | | | | | | | | | | | | |
| 1. Сортировка массива по числовому полю | | | | | | | | | | | | | | |
| 1. Вставка нового элемента перед выбранным элементом | | | | | | | | | | | | | | |
| 1. Вставка нового элемента после выбранного элемента | | | | | | | | | | | | | | |
| 1. Замена выбранного элемента | | | | | | | | | | | | | | |
| 1. Удаление элементов, начиная от выбранного | | | | | | | | | | | | | | |
| 1. Просмотр элементов и вычисление среднего на множестве тех элементов, кото­рые | | | | | | | | | | | | | | |
| попадают в заданный диапазон по заданному полю (поле типа float или longint) | | | | | | | | | | | | | | |
| 1. Просмотр элементов и вычисление минимума и максимума на множестве тех | | | | | | | | | | | | | | |
| эле­ментов, которые попадают в заданный диапазон по заданному полю (поле типа | | | | | | | | | | | | | | |
| float или longint) | | | | | | | | | | | | | | |
| Условия и ограничения | | | | | | | | | | | | | | |
| 1. Главную процедуру программы с реализацией простейшего меню следует опре­делить в отдельном модуле | | | | | | | | | | | | | | |
| 1. Процедуры, реализующие запросы, должны быть размещены в одном или более модулях | | | | | | | | | | | | | | |
| 1. Глобальные данные использовать нельзя | | | | | | | | | | | | | | |
| 1. На экран выводить элементы в виде таблицы (один элемент – одна строка таб­лицы) | | | | | | | | | | | | | | |
| 1. Если после выполнения запроса изменяется хотя бы один элемент, то заканчи­вать запрос выводом всего множества элементов | | | | | | | | | | | | | | |
| 1. Тестами к заданиям служат 2 текстовых файла с правдоподобной информацией | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | |
| 4. Содержание расчетно-пояснительной записки (перечень подлежащих разработке вопросов) | | | | | | | | | | | | | | |
| Введение | | | | | | | | | | | | | | |
| 1.Постановка задачи | | | | | | | | | | | | | | |
| 2.Разработка алгоритмов | | | | | | | | | | | | | | |
| 3.Разработка программы | | | | | | | | | | | | | | |
| 4.Тестирование | | | | | | | | | | | | | | |
| Заключение | | | | | | | | | | | | | | |
| Список использованных источников | | | | | | | | | | | | | | |
| Приложение А Текст программы | | | | | | | | | | | | | | |
| Приложение Б Графический материал | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | |
| 5. Перечень графического материала (с точным указанием обязательных чертежей и | | | | | | | | | | | | | | |
| графиков) | |  | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | |
| 6. Консультанты по проекту (с указанием относящихся к ним разделов проекта) | | | | | | | | | | | | | | |
| Хацкевич М.В. по ЕСПД | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | |
| 7. Дата выдачи задания | | | | | | | 1.03.2021 г. | | | | | | | |
|  | | | | | | |  | | | | | | | |
| 8. Календарный график работы над проектом на весь период проектирования (с указанием | | | | | | | | | | | | | | |
| сроков выполнения и трудоемкость отдельных этапов) | | | | | | | | | | | |  | | |
|  | | | | | | | | | | | | | | |
| Раздел 1 : 31.03 – 13.04: 30% | | | | | | | | | | | | | | |
| Раздел 2 : 14.04 – 04.05: 30% | | | | | | | | | | | | | | |
| Раздел 3,4 : 05.05 – 25.05: 40% | | | | | | | | | | | | | | |
| Оформление проекта: 26.05 – 29.05: 10% | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | |
|  | | | | | | Руководитель | | | |  | | | |  |
|  | | | | | |  | | | | (подпись) | | | |  |
|  | | | | | | | | | | | | | | |
| Задание принял к исполнению (дата) | | | | | | | | | | 1.03.2021г. | | | |  |
|  | | | | | | | | | | | | | | |
| (подпись студента) | | | | |  | | | | | | | | |  |

**СОДЕРЖАНИЕ**

1.

2.

3.

#include<iostream>

#include<string>

#include<stdio.h>

#include<ctime>

#include<utility>

#include <iomanip>

#include <fstream>

using namespace std;

struct PC

{

char student[100];

long int stependia;

float sredball;

;

};

void VvodStruct(struct PC\* compon, int\* size);

void VivodStruct(struct PC\* compon, int\* size);

void Sortirovka(struct PC\* compon, int\* size);

void DeleteStroks(struct PC\* compon, int\* size);

int main()

{

setlocale(LC\_ALL, "ru");

srand(time(0));

int size = 10;

struct PC\* compon = new struct PC[size];

int a;

cout << setw(10) << "МЕНЮ" << endl;

do {

cout << "\t1 - ввод элементов в структуру\t2 - вывод структуры на экран\t\n\n\t3 - сортировка по полю\t4 - удаление выбранной строки\t\n\n\t5 - exit\t\n\n";

cin >> a;

switch (a)

{

case 1:system("cls"); VvodStruct(compon, &size); system("cls"); break;

case 2:VivodStruct(compon, &size); system("pause"); system("cls"); break;

case 3:Sortirovka(compon, &size); system("pause"); system("cls"); break;

case 4:system("cls"); DeleteStroks(compon, &size); system("pause"); system("cls"); break;

}

} while (a != 5);

delete[]compon;

system("pause");

return 0;

}

void VvodStruct(struct PC\* compon, int\* size)

{

////////ВВОД ЭЛЕМЕНТОВ СТРУКТУРЫ

int a;

bool vod = false;

do

{

if (\*size == 10)

{

\*size = 0;

}

cout << "Введите фамилию студента: ";

cin >> compon[\*size].student;

cout << "Введите стипендию студента: ";

cin >> compon[\*size].stependia;

cout << "Введите средний балл студента: ";

cin >> compon[\*size].sredball;

\*size = \*size + 1;

system("cls");

cout << "Продолжить ввод структур\n1 - Да\n2 - Нет" << endl;

cin >> a;

switch (a)

{

case 1:vod = false; break;

case 2:vod = true; break;

}

system("cls");

} while (vod == false);

cout << endl;

}

void VivodStruct(struct PC\* compon, int\* size)

{

int stroka = 0;

cout <<fixed<< "№\tФамилия\t\tСтепендия\tСр. балл" << endl;

for (int i = 0; i < \*size; i++)

{

cout <<fixed<< stroka++ << " \t" << compon[i].student << " \t\t" << compon[i].stependia << " \t\t" << compon[i].sredball << endl;

}

cout << endl;

}

void Sortirovka(struct PC\* compon, int\* size)

{

///////////СОРТИРОВКА

cout << "Структура отсортирована по степендии" << endl;

cout << endl;

for (int i = 0; i < \*size; i++)

{

for (int j = 0; j < \*size; j++)

{

if (compon[i].stependia < compon[j].stependia)

{

swap(compon[i], compon[j]);

}

}

}

///////////СОРТИРОВКА

}

void DeleteStroks(struct PC\* compon, int\* size)

{

int stroka = 0;

cout << "№\tФамилия\tСтепендия\tСр. балл" << endl;

for (int i = 0; i < \*size; i++)

{

cout << stroka++ << " \t" << compon[i].student << " \t" << compon[i].stependia << " \t" << compon[i].sredball << endl;

}

cout << endl;

int del;

cout << "Введите № строки для удаления: ";

cin >> del;

cout << endl;

\*size = \*size - 1;

struct PC\* newCompon = new struct PC[\*size];

for (int i = 0; i < \*size; i++)

{

if (i != del)

{

newCompon[i] = compon[i];

}

else

{

newCompon[i] = compon[i + 1];

}

}

for (int i = 0; i < \*size; i++)

{

compon[i] = newCompon[i];

}

delete[]newCompon;

}