**Лабораторная работа № 3. Работа с файлами на языке С++**

|  |  |  |
| --- | --- | --- |
| Задание | | Решение |
| 1. Изучить способы организации данных в виде *структуры*, выполнив программу, записанную в данном пункте.  Дополнить структуру дополнительными сведениями о работниках и вывести их вместе с фамилиями на экран. | | **#include <iostream>**  **using namespace std;**  **struct Worker**  **{ char lastName[30];**  **char position[30];**  **int year;**  **};**  **int main()**  **{ setlocale(LC\_ALL, "Russian");**  **const int size = 3;**  **const int currentYear = 2017;**  **int i, b, counter = 0;**  **Worker a[size];**  **for (i = 0; i < size; i++)**  **{ cout << "Введите фамилию " << i + 1 << "-ого работника " << endl;**  **cin >> a[i].lastName;**  **cout << endl << "Должность: ";**  **cin >> a[i].position;**  **cout<< endl << "Год поступления на работу: ";**  **cin >> a[i].year;**  **}**  **cout << endl << "Введите стаж работы ";**  **cin >> b;**  **cout << endl;**  **for (i = 0; i < size; i++)**  **{ if (b < currentYear - a[i].year)**  **{ cout << a[i].lastName << " ";**  **counter++;**  **}**  **else**  **if (i == size-1 && counter > 0)**  **cout << "Работников с более высоким стажем нет.";**  **}**  **return 0;**  **}**    #include <iostream>  using namespace std;  struct Worker  {  char lastName[30];  char position[30];  int year;  int age; // Новый атрибут - возраст  double salary; // Новый атрибут - зарплата  };  int main()  {  setlocale(LC\_ALL, "Russian");  const int size = 3;  const int currentYear = 2025;  int i, b, counter = 0;  Worker a[size];  for (i = 0; i < size; i++)  {  cout << "Введите фамилию " << i + 1 << "-ого работника: ";  cin >> a[i].lastName;  cout << "Должность: ";  cin >> a[i].position;  cout << "Год поступления на работу: ";  cin >> a[i].year;  cout << "Возраст: ";  cin >> a[i].age; // Ввод возраста  cout << "Зарплата: ";  cin >> a[i].salary; // Ввод зарплаты  }  cout << "\nВведите стаж работы: ";  cin >> b;  cout << endl;  for (i = 0; i < size; i++)  {  if (b < currentYear - a[i].year)  {  cout << "Фамилия: " << a[i].lastName<< ", Должность: " << a[i].position<< ", Год поступления: " << a[i].year<< ", Возраст: " << a[i].age<< ", Зарплата: " << a[i].salary << endl;  counter++;  }  }  if (counter == 0)  {  cout << "Работников с более высоким стажем нет." << endl;  }  return 0;  } |
| 2. В правой части приведен пример программы, в которой используется *указатель на структуру*.  Выполнить программу, меняя скорость вывода информации.  Написать пояснения к программе. | | #include <conio.h>  #include <iostream>  using namespace std;  #define DELAY 12800 // Определение задержки в цикле обновления  // Объявление функций для обновления и отображения времени  void update(struct MyTime\* t);  void display(struct MyTime\* t);  // Структура для хранения времени  struct MyTime  {  int hours; // Часы  int minutes; // Минуты  int seconds; // Секунды  };  int main(void)  {  struct MyTime St; // Создание переменной типа MyTime  St.hours = 0; // Инициализация часов  St.minutes = 0; // Инициализация минут  St.seconds = 0; // Инициализация секунд  for (; ;) // Бесконечный цикл  {  update(&St); // Обновление времени  display(&St); // Отображение времени  if (\_kbhit()) // Проверка, была ли нажата клавиша  return 0; // Выход из программы при нажатии клавиши  }  }  void update(struct MyTime\* t)  {  t->seconds++; // Увеличение секунд на 1  if (t->seconds == 60) { t->seconds = 0; t->minutes++; } // Если секунд 60, сбросить до 0 и увеличить минуты  if (t->minutes == 60) { t->minutes = 0; t->hours++; } // Если минут 60, сбросить до 0 и увеличить часы  if (t->hours == 24) // Если часов 24, сбросить до 0  t->hours = 0;  for (long int i = 1; i < DELAY; ++i); // Задержка для замедления обновления  }  void display(struct MyTime\* t)  {  cout << t->hours << '.' << t->minutes << '.' << t->seconds << endl; // Вывод текущего времени  } |
| 3. В правой части приведен пример программы, которая осуществляет работу с данными, организованными в виде структуры, содержащей информацию о студентах: №,Фамилия, Год рождения, Факультет.  Проанализировать работу главной функции и функций, входящих в программу.  Добавить операторы выдачи сообщений об ошибках при вводе данных неправильного типа. | | #include <iostream>  #include <stdio.h>  int main(void)  {  setlocale(LC\_CTYPE, "Russian");  FILE\* fp;  errno\_t err;  char const\* st = "привет";  err = fopen\_s(&fp, "a.bin", "w+");  if (err != 0)  {  perror("ошибка открытия a.txt");  return EXIT\_FAILURE;  }  fwrite(st, strlen(st), 1, fp);  printf("Записан элемент\n");  fseek(fp, 0, 0);//читаем файл с начала  char buf[256];  fgets(buf, 256, fp);  printf("%s", buf);  fclose(fp);  return 0;  } |
| 4. Выполнить программу, записанную в правой части, которая реализует операции *позиционирования* в файле и *блоковый* вывод.  Изменить программу так, чтобы выводились на экран подряд все предложения с 1 по **n**. | | #include <stdio.h>  #include <fstream>  using namespace std;  int main()  {  int k; FILE\* fin;  fopen\_s(&fin, "name1.txt", "rt");  ofstream fout1("name2.txt");  ofstream fout2("name3.txt");  printf("Vvedite 4islo k\n");  scanf\_s("%d", &k);  while (!feof(fin))  {  char s[255] = "";  fgets(s, 254, fin);  if (strlen(s) <= k)  fout1 << s;  else  {  for (int i = strlen(s) - k - 1; i < strlen(s); i++)  fout1 << s[i];  }  if (strlen(s) < k)  fout2 << " " << endl;  else  fout2 << s[k - 1] << endl;  }  fclose(fin);  fout1.close();  fout2.close();  return 0;  } |
| 5. В соответствии со своим вариантом разработать программы для работы с файлами на ***языке С++.***  Для первой программы необходимо предварительно создать текстовый файл FILE1из нескольких строк и записать в него данные.  Во второй программе ввод информации с клавиатуры и вывод в консольное окно осуществить в главной функции, а запись в файл и чтение из файла − в функциях пользователя.  Встроенные функции для работы со строками не использовать.  Разработать ***блок-схему*** программы. | | |
| №Варианта  10 | | |
| Задание | Код | |
| 1. Скопировать из файла **FILE1** в файл **FILE2** все строки, в которых нет слов, совпадающих с первым словом. Определить количество согласных букв в первой строке файла **FILE2**.  2. Ввести с клавиатуры строку символов, состоящую из слов, разделенных пробелами, и записать ее в файл. Прочитать из файла данные и вывести все слова, которые содержат букву «**р**». | fstream fin("FILE1.txt");  fstream fout("FILE2.txt");  char word[50];  fin >> word;  fin.seekg(0);  while (!fin.fail()) {  char str[100];  fin.getline(str, 100);  int i = 0;  int j = 0;  bool flag = false;  while (str[i-1] != '\0') {  if (str[i] == ' ' || str[i] == '\0') {  if (word[j] == '\0') {  flag = true;  break;  }  }  if (word[j] != str[i]) {  j = 0;  while (str[i - 1] != '\0' && str[i-1] != ' ') i++;  }  i++;  j++;  }  if (flag) {  fout << str << '\n';  }  }  char str[100];  fout.seekg(0);  fout.getline(str, 100);  int sum = 0;  for (int i = 0; str[i] != '\0'; i++) {  if (!isvowel(str[i])) {  sum++;  }  }  cout << "Количество согласных:" << sum;  fout.close();  fin.close();  break;  bool isvowel(char ch) {  if (ch == ' ') return true;  char vowels[] = "aeiou";  for (int i = 0; vowels[i] != '\0'; i++) {  if (ch == vowels[i]) {  return true;  }  }  return false;  }        ofstream fout("FILE10\_2.txt");  char str[256];  cin.ignore(cin.rdbuf()->in\_avail());  cin.getline(str, 256);  fout << str;  fout.close();  ifstream fin("FILE10\_2.txt");  char buf[50];  while (fin >> buf) {  int i = 0;  while (buf[i] != '\0') {  if (buf[i] == 'p') {  cout << buf << ' ';  break;  }  i++;  }  }  fin.close(); | |

Допы

|  |  |
| --- | --- |
| №Варианта 11 | |
| Задание | Код/результат |
| 1. Скопировать из файла **FILE1** в файл **FILE2** все строки, которые содержат только одно слово. Подсчитать количество символов в файле **FILE2**.  2. Ввести с клавиатуры две строки символов, состоящих из слов, разделенных пробелами, и записать их в файл. Прочитать из файла данные. Найти самое короткое слово в первой строке и самое длинное во второй строке. | ifstream fin("FILE11\_1.txt");  ofstream fout("FILE11\_2.txt");  char buf[256];  while (fin >> buf) {  int i = 0;  bool flag = true;  while (buf[i] != '\0') {  if (buf[i] == ' ') {  if (buf[i + 1] != '\0' && buf[i + 1] != ' ') {  flag = false;  }  }  }  if (flag) {  fout << buf << '\n';  }  }  fin.close();  fout.close();  ifstream fin1("FILE11\_2");  int sum = 0;  while (!fin1.fail()) {  char ch = fin1.get();  sum++;  }  cout << "Количество символов в файле: " << sum;  break;      ofstream fout("FILE11\_3.txt");  char buf[256];  cout << "Первая строка:\n";  cin.ignore(cin.rdbuf()->in\_avail());  cin.getline(buf, 256);  fout << buf << '\n';  cout << "Вторая строка:\n";  cin.ignore(cin.rdbuf()->in\_avail());  cin.getline(buf, 256);  fout << buf;  fout.close();  ifstream fin("FILE11\_3.txt");  fin.getline(buf, 256);  int min = 256, i = 0, start = 0;  while (buf[i - 1] != '\n') {  if (buf[i] == ' ' || buf[i] == '\n') {  if (i - start < min) {  min = i - start;  }  start = i + 1;  }  i++;  }  cout << "Минимальная длина слова: " << min;  int max = 0;  i = 0;  start = 0;  while (buf[i - 1] != '\n') {  if (buf[i] == ' ' || buf[i] == '\n') {  if (i - start > max) {  max = i - start;  }  start = i + 1;  }  i++;  }  cout << "\nМаксимальная длина слова: " << max;  break; |

|  |  |
| --- | --- |
| №Варианта 12 | |
| Задание | Код/результат |
| 1. Скопировать из файла **FILE1** в файл **FILE2** все строки, в которых есть слова, совпадающие с первым словом. Определить количество букв в файле **FILE2**.  2. Ввести с клавиатуры строку символов, состоящую из цифр и слов, разделенных пробелами, и записать ее в файл. Прочитать из файла данные и вывести нечетные числа строки. | fstream fin("FILE12\_1.txt");  fstream fout("FILE12\_2.txt");  char word[100];  fin >> word;  fin.seekg(0);  while (!fin.fail()) {  char str[256];  fin.getline(str, 100);  int i = 0;  int j = 0;  bool flag = false;  while (str[i - 1] != '\0') {  if (str[i] == ' ' || str[i] == '\0') {  if (word[j] == '\0') {  flag = true;  break;  }  }  if (word[j] != str[i]) {  j = 0;  while (str[i - 1] != '\0' && str[i - 1] != ' ') i++;  }  i++;  j++;  }  if (flag) {  fout << str << '\n';  }  }  fout << '\0';  fout.close();  fin.close();  fin.open("FILE12\_2.txt");  int sum = 0;  char ch = fin.get();  while (ch != '\0') {  if (ch >= 'a' && ch <= 'z' || ch >= 'A' && ch <= 'Z')  sum++;  ch = fin.get();  }  cout << "Количество букв: " << sum;  fin.close();  break;      ofstream fout("FILE12\_3.txt");  char buf[256];  cin.ignore(cin.rdbuf()->in\_avail());  cin.getline(buf, 256);  fout << buf;  fout.close();  ifstream fin("FILE12\_3.txt");  while (fin>>buf) {  int i = 0;  bool flag = true;  while (buf[i] != '\0') {  if (!(buf[i] >= '0' && buf[i] <= '9')) {  flag = false;  break;  }  i++;  }  if (flag) {  if ((buf[i - 1] - '0') % 2 == 1) {  cout << buf << ' ';  }  }  }  fin.close();  break; |
| №Варианта 13 | |
| Задание | Код/результат |
| 1. Даны два файла целых чисел с именами **fileA** и **fileB**. Получить новый файл с именем **fileC**, который содержит сумму элементов файлов **fileA** и **fileB**.  2. Компоненты файла**file1** – целые двухзначные (отличные от нуля) числа, причем сначала записаны 5 положительных чисел, затем 5 отрицательных, и т. д. Получить файл **file2***,* в котором записаны числа из файла **file1**,сначала 10 положительных чисел, затем 10 отрицательных и т. д. | ifstream fin("FILE13\_1.txt");  ofstream fout("FILE13\_2.txt");  char buf[256];  while (!fin.fail()) {  fin.getline(buf, 256);  int k = 0, i = 0;  while (buf[i] != '\0') {  if (buf[i] == ' ') {  if (!(buf[i + 1] == ' ' || buf[i + 1] == '\0')) {  k++;  }  }  i++;  }  if (k > 1) {  fout << buf << '\n';  }  }  fin.close();  fout.close();  fin.open("FILE13\_2.txt");  int max = -1, j = 1, maxj = 0;  while (fin>>buf) {  int i = 0;  int sum = 0;  while (buf[i] != '\0') {  if (isvowel(buf[i])) {  sum++;  }  i++;  }  if (sum > max) {  max = sum;  maxj = j;  }  j++;  }  cout << "Максимальное количество глассных: " << max << " в слове под номером: " << maxj;  break;        ofstream fout("FILE13\_3.txt");  char buf[256];  cin.ignore(cin.rdbuf()->in\_avail());  cin.getline(buf, 256);  fout << buf<<'\0';  fout.close();  ifstream fin("FILE13\_3.txt");  char ch = fin.get();  char skobki[6] = { '(',')','[',']','{','}' };  int sum[6] = { 0,0,0,0,0,0 };  while (ch != '\0') {  for (int i = 0; i < 6; i++) {  if (ch == skobki[i]) {  sum[i]++;  }  }  ch = fin.get();  }  for (int i = 0; i < 6; i++) {  cout << "Количество скобок вида: " << skobki[i] << " - " << sum[i]<<'\n';  }  break; |