Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа 5

По дисциплине «Основы алгоритмизации и программирования»

На тему «Объединения, перечисления, битовые поля»

Выполнил:

Студент 1 курса 6 группы

Кучерук Николай Петрович

Преподаватель: асс. Андронова М.В.

2023, Минск

**Задание**

| **№ варианта** | **Условие задачи** |
| --- | --- |
| **9** | **Вокзал**. Номер поезда, пункт назначения, дни следования, время прибытия, время отправления. Выбор по пункту назначения. Дни следования реализовать с помощью перечисления. Время выбытия и прибытия реализовать с помощью битового поля (часы, минуты). |

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <string>  # define size 8 //можно ввести до 8 составов  using namespace std;  void enter\_new();  void out();  void search(string);  enum weekdays {  Monday = 1,Tuesday,Wednesday,Thursday,Friday,Saturday,Sunday  };  struct train//создаем структуру поезд,которая содержит нужные нам параметры о поездах  {  string name;  short num;  int time\_h : 6;//ораничиваем границы переменных с помощью битовых полей  int time\_m : 7;  short stard\_day;  short end\_day;  void days\_of\_following() {  for (int i = stard\_day; i <= end\_day; i++) {  if (i == weekdays::Monday) { cout << "Monday "; }  if (i == weekdays::Tuesday) { cout << "Tuesday "; }  if (i == weekdays::Wednesday) { cout << "Wednesday "; }  if (i == weekdays::Thursday) { cout << "Thursday "; }  if (i == weekdays::Friday) { cout << "Friday "; }  if (i == weekdays::Saturday) { cout << "Saturday "; }  if (i == weekdays::Sunday) { cout << "Sunday "; }  }  }  };  train train\_list[size];//массив,позволяющий хранить данные о составах в рамках одной переменной  int current\_size = 0, choice; string place;  int main()//вызов менюс доступом ко всем функциям  {  system("color 70");  setlocale(LC\_CTYPE, "Russian");  cout << "Введите:" << endl;  cout << "1-для ввода поездов" << endl;  cout << "2-для вывода записи(ей) о поездах" << endl;  cout << "3-поиск по месту назначения" << endl;  cout << "4-для выхода" << endl;  cin >> choice;  while (true)//с помощью контекстного меню перемещаемся по возможностям программы  {  switch (choice)  {  case 1: enter\_new(); break;  case 2: out(); break;  case 3:  cout << endl << "Место назначения: " << endl;  cin >> place;  search(place);  case 4: return 0; break;  default: cout << "try again" << endl; cin >> choice;  }  }  }  void enter\_new()  {  int hr,min;  if (current\_size < size)//отслеживаем заполненнось расписания поездов  {  cout << endl << "Место назначения: " << endl;  getline(cin, train\_list[current\_size].name);  getline(cin, train\_list[current\_size].name);  cout << "Номер состава: " << endl;  cin >> train\_list[current\_size].num;  cout << "Время отправления(ЧЧ ММ): " << endl;  cin >> hr >> min;  train\_list[current\_size].time\_h = hr;  train\_list[current\_size].time\_m = min;  cout << "День начала следования: " << endl;  cin >> train\_list[current\_size].stard\_day;  cout << "День ококнчания следования: " << endl;  cin >> train\_list[current\_size].end\_day;  current\_size++;  }  else  cout << "Введено максимальное кол-во поездов";  cout << "Что дальше?" << endl;  cin >> choice;  }  void out()  {  cout << "Место назначения \t\t" << "Номер состава \t\t" << "Время отпрвления \t\t" << endl;  for (int i = 0; i < current\_size; i++)  {  cout << i + 1 << ". " << train\_list[i].name << " \t" << train\_list[i].num << " \t" << train\_list[i].time\_h << ':' << train\_list[i].time\_m << " \t";  train\_list[i].days\_of\_following();  cout << endl;  }  cout << "Что дальше?" << endl;  cin >> choice;  }  void search(string place) {  bool succes = false;  for (int i = 0; i < current\_size; i++) {//ищем совпадения фамилий и выводим все параметры структуры,если их находим  if (place == train\_list[i].name) {  succes = true;  cout << i + 1 << ". " << train\_list[i].name << " \t" << train\_list[i].num << " \t" << train\_list[i].time\_h << ':' << train\_list[i].time\_m << " \t";  train\_list[i].days\_of\_following();  cout << endl;  }  }  if (!succes) { cout << "Совпадений не найдено." << endl; }  } |
| **Результат программы** |
|  |

| **№ варианта** | **Условие задачи** |
| --- | --- |
| **9** | **Ученики**. Ф.И.О., класс (цифра+буква) предметы, оценки, средний балл. Выбор по фамилии. |

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <fstream>  #include <string>  using namespace std;  union Grade  {  char charecter;  short number;  }inputgrade;  struct student//создаем структуру,в которую добавляем все параметры,которые хотим знать о студентах  {  string LastName;  string FirstName;  string FatherName;  string major;  string grade;  string grades;  string gpa;  };  student student\_list[30];//массив,позволяющий хранить данные о группе студентов в рамках одной переменной  int currentsize();//данная функция нужна для нахождения кол-ва уже записаных студентов в файле до начала программы  void input();  void output();  void del();  void fill();//данная функция нужна,чтобы заполнить наш массив данными из файла  void find\_lastname(string);  int current\_size = currentsize(), size = 30;  int main()  {  system("color 70");  setlocale(LC\_ALL, "Russian");  int choice, count = 1; string fio;  do  {  cout << "\n1.Ввод данных с клавиатуры и запись в файл\n";  cout << "2.Вывод данных из файла\n";  cout << "3.Поиск по фамилии\n";  cout << "4.Удалить строку \n";  cout << "0.Выход из программы\n\n";  cout << "Введите номер операции: ";  cin >> choice;  switch (choice)  {  case 1: cout << "Введите количество студентов: ";  input(); break;  case 2: output(); break;  case 3: { cout << "Введите фамилию: ";  cin >> fio;  find\_lastname(fio); break;  }  case 4: del(); break;  case 0: exit(0); break;  }  } while (choice != 0);  }  void input() {  int number;  char buffer[200]; string buf;  ofstream file("database.txt", ios::app);//открываем файл для записи,не удаляя содержимое  if (file.fail())  {  cout << "\n Ошибка открытия файла";  exit(1);  }  cout << "Введите кол-во учеников,которые вы хотите добавить в реестр, на данный момент записано " << current\_size << " (макс.30): "; cin >> number;  if (current\_size + number >= 30) { cout << "Превышен лимит учеников."; exit(1); }  for (int p = current\_size; p < current\_size + number; p++)//записываем значения параметров структуры и записываем их в файл  {  cout << endl << "Фамилия: "; cin >> student\_list[p].LastName; file << student\_list[p].LastName << ' ';  cout << "Имя: "; cin >> student\_list[p].FirstName; file << student\_list[p].FirstName << ' ';  cout << "Отчество: "; cin >> student\_list[p].FatherName; file << student\_list[p].FatherName << ' ';  cout << "Год обучения: "; cin >> inputgrade.number;  student\_list[p].grade+=(char)inputgrade.number;  cout << "Класс: "; cin >> inputgrade.charecter; student\_list[p].grade+=inputgrade.charecter;  file << student\_list[p].grade << ' ';  cout << "Профильные предметы: "; cin >> student\_list[p].major; file << student\_list[p].major << ' ';  cout << "Оценки "; cin >> student\_list[p].grades; file << student\_list[p].grades << ' ';  cout << "Средний балл: "; cin >> student\_list[p].gpa; file << student\_list[p].gpa << ' ' << endl;  }  current\_size += number;  file.close();  }  void output() {  string buf;  ifstream file("database.txt");//открываем файл в режиме чтения  if (file.fail())  {  cout << "\n Ошибка открытия файла";  exit(1);  }  for (int i = 0; i < current\_size; i++) {//выводим содержимое файла  getline(file, buf);  cout << i + 1 << ". " << buf << endl;  }  file.close();  }  void find\_lastname(string fio) {  bool succes = false;  fill();//заполняем массив точными данными из файла  for (int i = 0; i < current\_size; i++) {//ищем совпадения фамилий и выводим все параметры структуры,если их находим  if (fio == student\_list[i].LastName) { succes = true; cout << student\_list[i].LastName << ' ' << student\_list[i].FirstName << ' ' << student\_list[i].FatherName << ' ' << student\_list[i].grade << ' ' << student\_list[i].major << ' ' << student\_list[i].grades << ' ' << student\_list[i].gpa << endl; }  }  if (!succes) { cout << "Совпадений не найдено." << endl; }  }  int currentsize() {  int count = 0; string buf;  ifstream file("database.txt");  if (file.fail())  {  cout << "\n Ошибка открытия файла";  exit(1);  }  while (getline(file, buf)) {//считаем сколько строк мы смогли взять из файла,чтобы знать сколько студентов там есть  count++;  }  file.close();  return count;  }  void del() {  fill();//заполняем массив точными данными из файла  int number;  cout << "Введите номер студента,запись которого вы хотите удалить из реестра, на данный момент записано " << current\_size << " : "; cin >> number;  if (current\_size < number) { cout << "Данной записи не найдено."; exit(1); }  for (int i = number - 1; i < current\_size - 1; i++) {//удаляем нужную строку из массива и сторчки выше спускаем на 1 позицую вниз  student\_list[i] = student\_list[i + 1];  }  student\_list[current\_size - 1] = {};  current\_size--;//корректируем размеры массива после всех манипуляций с ней  ofstream file("database.txt");//удаляем все прошлое содержимое,чтобы заменить его на актульные данные из массива  if (file.fail())  {  cout << "\n Ошибка открытия файла";  exit(1);  }  for (int p = 0; p < current\_size; p++)  {  file << student\_list[p].LastName << ' ';  file << student\_list[p].FirstName << ' ';  file << student\_list[p].FatherName << ' ';  file << student\_list[p].grade << ' ';  file << student\_list[p].major << ' ';  file << student\_list[p].grades << ' ';  file << student\_list[p].gpa << ' ' << endl;  }  file.close();  }  void fill() {  int number, pos = 0; string buf;  ifstream file("database.txt");  if (file.fail())  {  cout << "\n Ошибка открытия файла";  exit(1);  }  for (int i = 0; i < current\_size; i++) {//данные о студенте представляют 1 строку с разделяющими пробелами, поэтому для получения данных в структуру мы должны извлекать все то, что находится между пробелами  getline(file, buf);  student\_list[i].LastName = buf.substr(pos, buf.find(' ')); pos = student\_list[i].LastName.size() + 1;  student\_list[i].FirstName = buf.substr(pos, buf.find(' ', pos) - pos); pos += student\_list[i].FirstName.size() + 1;  student\_list[i].FatherName = buf.substr(pos, buf.find(' ', pos) - pos); pos += student\_list[i].FatherName.size() + 1;  student\_list[i].grade = buf.substr(pos, buf.find(' ', pos) - pos); pos += student\_list[i].grade.size() + 1;  student\_list[i].major = buf.substr(pos, buf.find(' ', pos) - pos); pos += student\_list[i].major.size() + 1;  student\_list[i].grades = buf.substr(pos, buf.find(' ', pos) - pos); pos += student\_list[i].grades.size() + 1;  student\_list[i].gpa = buf.substr(pos, buf.find(' ', pos) - pos);  pos = 0;  }  file.close();  } |
| **Результат программы** |
|  |

Дополнительные задания

| **№ варианта** | **Условие задачи** |
| --- | --- |
| **7** | **Склад.** Наименование товара, цена, количество, процент торговой надбавки (5, 7, 11, 20, 25, 30). Выбор по цене. Процент торговой надбавки реализовать с помощью перечисления. |

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <string>  #include <Windows.h>  #define size 50  using namespace std;  void InputInfo();//функция для ввода  void OutputInfo();//функция для вывода  void DeleteInfo();//функция для удаления элементов  void SearchInfo();//функция для поиска  double ChangePercent();//функция для смены ставки  enum Percent//список перечисления,где каждому значению соответствует число  {  percent\_1 = 5,  percent\_2 = 10,  percent\_3 = 15,  percent\_4 = 20,  percent\_5 = 25,  percent\_6 = 30  };  struct Store//структура склада  {  string name\_product;  double price\_product;  int quantity;  };  Store list\_of\_store[size];  int choice, current\_size = 0, number\_product; double stavka = 5.0;  int main()  {  system("color 70");  setlocale(LC\_ALL, "RU");  SetConsoleCP(1251);  do {  cout << "Выберите операцию: " << endl;  cout << "1 - Ввод данных." << endl;  cout << "2 - Вывод данных." << endl;  cout << "3 - Поиск информации." << endl;  cout << "4 - Удаление информации." << endl;  cout << "5 - Работа с надбавкой." << endl;  cout << "6 - Выxод." << endl;  cout << endl;  cin >> choice;  switch (choice)  {  case 1:  InputInfo();  break;  case 2:  OutputInfo();  break;  case 3:  SearchInfo();  break;  case 4:  DeleteInfo();  break;  case 5:  stavka = ChangePercent();  break;  }  } while (choice != 6);  }  void InputInfo()//вводим необходимые параметры  {  cout << endl;  cout << "Ввод информации: " << endl;  cout << "Введите количество товаров: ";  cin >> number\_product;  cout << endl;  if (current\_size < size) {  for (int i = 0; i < number\_product; i++) {  cout << "Введите информацию для " << i + 1 << " товара: " << endl;  cout << "Введите название товара: ";  cin.ignore();  getline(cin, list\_of\_store[current\_size].name\_product);  cout << "Введите цену товара: ";  cin >> list\_of\_store[current\_size].price\_product;  cout << "Введите, в каком количестве данный товар находится на складе: ";  cin >> list\_of\_store[current\_size].quantity;  current\_size++;  cout << endl;  }  }  }  void OutputInfo()//выводим учитывая выбранную процентную ставку  {  cout << endl;  cout << "Вывод информации: " << endl;  for (int i = 0; i < current\_size; i++) {  cout << "Информация о " << i + 1 << " товаре: " << endl;  cout << "Название товара: ";  cout << list\_of\_store[i].name\_product << endl;  cout << "Цена товара с учётом надбавки: ";  cout << list\_of\_store[i].price\_product \* (1 + (stavka / 100)) << endl;  cout << "Количество данного товара на складе: ";  cout << list\_of\_store[i].quantity;  cout << endl;  }  }  void SearchInfo()//поиск информации по названию товара  {  cout << endl;  string name;  cout << "Введите название товара: ";  cin.ignore();  getline(cin, name);  for (int i = 0; i < current\_size; i++) {  if (name == list\_of\_store[i].name\_product) {  cout << "Информация о " << i + 1 << " товаре: " << endl;  cout << "Название товара: ";  cout << list\_of\_store[i].name\_product << endl;  cout << "Цена товара с учетом надбавки: ";  cout << list\_of\_store[i].price\_product \* (1 + stavka / 100) << endl;  cout << "Количество данного товара на складе: ";  cout << list\_of\_store[i].quantity;  cout << endl;  }  else {  cout << "Ничего не найдено.";  }  }  }  void DeleteInfo()//удаление строки по номеру в структуре  {  cout << endl;  int NumberLine;  cout << "Введите номер строки, которую хотите удалить: ";  cin >> NumberLine;  if (NumberLine > 0 && NumberLine <= current\_size) {  for (int del = NumberLine - 1; del < current\_size - 1; del++) {  list\_of\_store[del] = list\_of\_store[del + 1];  }  current\_size = current\_size - 1;  }  else {  cout << "Неверный номер строки." << endl;  }  }  double ChangePercent()//изменяем глобальную ставку на товары склада  {  cout << endl;  cout << "Выберите процент надбавки: " << endl;  cout << "1 - 5%" << endl;  cout << "2 - 10%" << endl;  cout << "3 - 15%" << endl;  cout << "4 - 20%" << endl;  cout << "5 - 25%" << endl;  cout << "6 - 30%" << endl;  int choice;  cin >> choice;  switch (choice)  {  case 1:  return Percent::percent\_1;//выбираем нужное значение из перечисления, а это возвращает подразумевающееся под значением число  break;  case 2:  return Percent::percent\_2;  break;  case 3:  return Percent::percent\_3;  break;  case 4:  return Percent::percent\_4;  break;  case 5:  return Percent::percent\_5;  break;  case 6:  return Percent::percent\_6;  break;  default:  cout << "Нет такого варианта." << endl;  break;  }  } |
| **Результат программы** |
|  |

| **№ варианта** | **Условие задачи** |
| --- | --- |
| **7** | **Склад.** Наименование товара, цена, количество, процент торговой надбавки (5, 10, 15, 20, 35, 30). Выбор по наименованию. |

|  |
| --- |
| **Код программы** |
| #include <string>  #include <iostream>  #include <fstream>  #define size 100  using namespace std;  void input();//функция ввода данных в структуру  void out();//функция вывода структуры  void search();//функция поиска по названию элемента  void ofile();//функция записи в файл  void ifile();//функция прочтения из файла  union ForStore//union содержащий в себе переменные вида char,int,double  {  char name\_product[50];  int quantity\_product;  double price\_product;  };  struct Store//структура склада  {  ForStore name\_of\_product;  ForStore cost;  ForStore many;  ForStore percent;  };  Store list\_of\_store[size];  int current\_size = 0, choice, number\_product;  int main()  {  setlocale(LC\_ALL, "Russian");  do {//вызов меню и функций  cout << "Выберите операцию:" << endl;  cout << "1 - Ввод информации." << endl;  cout << "2 - Вывод информации." << endl;  cout << "3 - Поиск информации." << endl;  cout << "4 - Запись в файл." << endl;  cout << "5 - Чтение из файла." << endl;  cout << "6 - Выход." << endl;  cin >> choice;  switch (choice)  {  case 1:  input();  break;  case 2:  out();  break;  case 3:  search();  break;  case 4:  ofile();  break;  case 5:  ifile();  break;  }  } while (choice != 6);  }  void input()//добавляем новый элемент в структуру  {  cout << endl;  cout << "Ввод информации: " << endl;  cout << "Введите количество товаров: ";  cin >> number\_product;  cout << endl;  if (current\_size < size) {  for (int i = 0; i < number\_product; i++) {  cout << "Введите информацию для " << i + 1 << " товара: " << endl;  cout << "Введите название товара: ";  cin.ignore();  cin.getline(list\_of\_store[current\_size].name\_of\_product.name\_product, 40);  cout << "Введите цену товара: ";  cin >> list\_of\_store[current\_size].cost.price\_product;  cout << "Введите, в каком количестве данный товар находится на складе: ";  cin >> list\_of\_store[current\_size].many.quantity\_product;  cout << "Введите процентную надбавку: ";  cin >> list\_of\_store[current\_size].percent.quantity\_product;  current\_size++;  cout << endl;  }  }  }  void out()//выводим всю информацию об структуре  {  cout << endl;  cout << "Вывод информации: " << endl;  for (int i = 0; i < current\_size; i++) {  cout << "Информация о " << i + 1 << " товаре: " << endl;  cout << "Название товара: ";  cout << list\_of\_store[i].name\_of\_product.name\_product << endl;  cout << "Цена товара: ";  cout << list\_of\_store[i].cost.price\_product << endl;  cout << "Количество данного товара на складе: ";  cout << list\_of\_store[i].many.quantity\_product << endl;  cout << "Процентная надбавка: ";  cout << list\_of\_store[i].percent.quantity\_product;  cout << endl;  }  }  void search() {//ищем элемент с совпадающем названием  char first\_name[50];  cout << "Введите наименование товара: ";  cin.ignore();  cin.getline(first\_name, 40);  for (int i = 0; i < current\_size; i++) {  if (strcmp(first\_name, list\_of\_store[i].name\_of\_product.name\_product) == 0) {  cout << "Информация о " << i + 1 << " товаре: " << endl;  cout << "Название товара: ";  cout << list\_of\_store[i].name\_of\_product.name\_product << endl;  cout << "Цена товара: ";  cout << list\_of\_store[i].cost.price\_product << endl;  cout << "Количество данного товара на складе: ";  cout << list\_of\_store[i].many.quantity\_product << endl;  cout << "Процентная надбавка: ";  cout << list\_of\_store[i].percent.quantity\_product;  cout << endl;  }  }  }  void ofile()//запись в файл  {  ofstream file1("store.txt", ios::app);  if (!file1) {  cout << "Не удалось открыть файл" << endl;  }  for (int i = 0; i < current\_size; i++) {  file1 << "Информация о " << i + 1 << " товаре: " << endl;  file1 << "Название товара: " << list\_of\_store[i].name\_of\_product.name\_product << endl;  file1 << "Цена товара: " << list\_of\_store[i].cost.price\_product << endl;  file1 << "Количество товара: " << list\_of\_store[i].many.quantity\_product << endl;  file1 << "Процентная надбавка: " << list\_of\_store[i].percent.quantity\_product << endl;  }  file1.close();  cout << "Данные успешно записаны в файл." << endl;  }  void ifile()//чтение из файла  {  ifstream file2("store.txt");  if (!file2) {  cout << "Не удалось открыть файл." << endl;  }  string line;  while (getline(file2, line)) {  cout << line << endl;  }  file2.close();  cout << "Данные успешно прочтены из файла." << endl;  } |
| **Результат программы** |
|  |

| **№ варианта** | **Условие задачи** |
| --- | --- |
| **12** | **Записная книжка.** Ф.И.О, дата рождения, адрес, телефон . Поиск по фамилии. Дату рождения реализовать с помощью битового поля. |

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <Windows.h>  #include <string>  #define size 100  using namespace std;  void InputInfo();//функция для ввода  void OutputInfo();//функция для вывода  void DeleteInfo();//функция для удаления элементов  void SearchInfo();//функция для поиска  struct Date//структура даты в записной книге  {  unsigned int day : 5;//с помщью битовых полей обозначаем необходимое число битов для записи информации  unsigned int month : 5;  unsigned int year : 12;  };  struct book//структура записной книжки  {  Date date;  string last\_name;  string name;  string dadname;  string address;  string number\_phone;  };  book list\_of\_book[size];  int choice, current\_size = 0, number\_persons, d;  int main()  {  system("color 70");  setlocale(LC\_ALL, "RU");  SetConsoleCP(1251);  do {  cout << "Выберите операцию: " << endl;  cout << "1 - Ввод данных." << endl;  cout << "2 - Вывод данных." << endl;  cout << "3 - Поиск информации." << endl;  cout << "4 - Удаление информации." << endl;  cout << "5 - Выxод." << endl;  cout << endl;  cin >> choice;  switch (choice)  {  case 1:  InputInfo();  break;  case 2:  OutputInfo();  break;  case 3:  SearchInfo();  break;  case 4:  DeleteInfo();  break;  }  } while (choice != 5);  }  void InputInfo()//вводим необходимые параметры  {  cout << endl;  cout << "Ввод информации: " << endl;  cout << "Введите количество персон: ";  cin >> number\_persons;  cout << endl;  if (current\_size < size) {  for (int i = 0; i < number\_persons; i++) {  cout << "Информация о " << i + 1 << " персоне: " << endl;  cout << "Введите имя: ";  cin.ignore();  getline(cin, list\_of\_book[current\_size].name);  cout << "Введите фамилию: ";  getline(cin, list\_of\_book[current\_size].last\_name);  cout << "Введите отчество: ";  getline(cin, list\_of\_book[current\_size].dadname);  cout << "Введите адрес: ";  getline(cin, list\_of\_book[current\_size].address);  cout << "Введите мобильный телефон: ";  getline(cin, list\_of\_book[current\_size].number\_phone);  cout << "Введите день рождения: ";  cin >> d;  list\_of\_book[current\_size].date.day = d;  cout << "Введите месяц рождения: ";  cin >> d;  list\_of\_book[current\_size].date.month = d;  cout << "Введите год рождения: ";  cin >> d;  list\_of\_book[current\_size].date.year = d;  current\_size++;  cout << endl;  }  }  }  void OutputInfo()//выводим учитывая выбранную процентную ставку  {  cout << endl;  cout << "Вывод информации: " << endl;  for (int i = 0; i < current\_size; i++) {  cout << "Информация о " << i + 1 << " персоне: " << endl;  cout << "Имя: ";  cout << list\_of\_book[i].name << endl;  cout << "Фамилия: ";  cout << list\_of\_book[i].last\_name << endl;  cout << "Отчество: ";  cout << list\_of\_book[i].dadname << endl;  cout << "Адрес: ";  cout << list\_of\_book[i].address << endl;  cout << "Мобильный телефон: ";  cout << list\_of\_book[i].number\_phone << endl;  cout << "Дата рождения: " << endl;  cout << "День: " << list\_of\_book[i].date.day << endl;  cout << "Месяц: " << list\_of\_book[i].date.month << endl;  cout << "Год: " << list\_of\_book[i].date.year << endl;  cout << endl;  }  }  void DeleteInfo()//удаление строки по номеру в структуре  {  cout << endl;  int NumberLine;  cout << "Введите номер строки, которую хотите удалить: ";  cin >> NumberLine;  if (NumberLine > 0 && NumberLine <= current\_size) {  for (int del = NumberLine - 1; del < current\_size - 1; del++) {  list\_of\_book[del] = list\_of\_book[del + 1];  }  current\_size = current\_size - 1;  }  else {  cout << "Неверный номер строки." << endl;  }  }  void SearchInfo()//поиск информации по фамилии  {  string surname;  cout << "Введите фамилию: ";  cin.ignore();  getline(cin, surname);  for (int i = 0; i < current\_size; i++) {  if (surname == list\_of\_book[i].last\_name) {  cout << "Информация о " << i + 1 << " персоне: " << endl;  cout << "Имя: ";  cout << list\_of\_book[i].name << endl;  cout << "Фамилия: ";  cout << list\_of\_book[i].last\_name << endl;  cout << "Отчество: ";  cout << list\_of\_book[i].dadname << endl;  cout << "Адрес: ";  cout << list\_of\_book[i].address << endl;  cout << "Мобильный телефон: ";  cout << list\_of\_book[i].number\_phone << endl;  cout << "Дата рождения: " << endl;  cout << "День: " << list\_of\_book[i].date.day << endl;  cout << "Месяц: " << list\_of\_book[i].date.month << endl;  cout << "Год: " << list\_of\_book[i].date.year << endl;  cout << endl;  }  }  } |
| **Результат программы** |
|  |

| **№ варианта** | **Условие задачи** |
| --- | --- |
| **12** | **Записная книжка.** Ф.И.О, дата рождения, адрес, телефон, место работы или учебы, должность. Поиск по фамилии. |

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <fstream>  #include <Windows.h>  #include <string>  #define size 100  using namespace std;  void input();//функция ввода элементов записной книги  void out();//функция вывода содержания записной книги  void search();//функция поиска по содержанию книги  void infile();//функция записи в файл  void onfile();//функция вывода содержимого файла  union ForBook//обьединение массиов char разной длины  {  char name[50];  char surname[50];  char dadname[50];  char address[50];  char number\_phone[20];  char place\_work\_or\_study[10];  char post[20];  };  struct Book  {  ForBook yourself\_name;  ForBook last\_name;  ForBook dad\_name;  ForBook direction;  ForBook phone;  ForBook place;  ForBook position;  string date;  };  Book list\_of\_book[size];  int choice, current\_size = 0, number\_persons;  int main()  {  system("color 70");  setlocale(LC\_ALL, "Russian");  SetConsoleCP(1251);  do {  cout << "Выберите операцию:" << endl;  cout << "1 - Ввод информации." << endl;  cout << "2 - Вывод информации." << endl;  cout << "3 - Поиск информации." << endl;  cout << "4 - Запись в файл." << endl;  cout << "5 - Чтение из файла." << endl;  cout << "6 - Выход." << endl;  cin >> choice;  switch (choice)  {  case 1:  input();  break;  case 2:  out();  break;  case 3:  search();  break;  case 4:  infile();  break;  case 5:  onfile();  break;  }  } while (choice != 6);  }  void input()  {  cout << endl;  cout << "Ввод информации: " << endl;  cout << "Введите количество человек: ";  cin >> number\_persons;  cout << endl;  if (current\_size < size) {  for (int i = 0; i < number\_persons; i++) {  cout << "Информация о " << i + 1 << " персоне: " << endl;  cout << "Введите имя: ";  cin.ignore();  cin.getline(list\_of\_book[current\_size].yourself\_name.name, 40);  cout << "Введите фамилию: ";  cin.getline(list\_of\_book[current\_size].last\_name.surname, 40);  cout << "Введите отчество: ";  cin.getline(list\_of\_book[current\_size].dad\_name.dadname, 40);  cout << "Введите адрес: ";  cin.getline(list\_of\_book[current\_size].direction.address, 40);  cout << "Введите номер телефона: ";  cin.getline(list\_of\_book[current\_size].phone.number\_phone, 15);  cout << "Введите место работы или учебы: ";  cin.getline(list\_of\_book[current\_size].place.place\_work\_or\_study, 10);  cout << "Введите должность: ";  cin.getline(list\_of\_book[current\_size].position.post, 20);  cout << "Введите дату рождения: ";  getline(cin, list\_of\_book[current\_size].date);  current\_size++;  cout << endl;  }  }  }  void out()  {  cout << endl;  cout << "Вывод информации: " << endl;  for (int i = 0; i < current\_size; i++) {  cout << "Информация о " << i + 1 << " персоне: " << endl;  cout << "Имя: ";  cout << list\_of\_book[i].yourself\_name.name << endl;  cout << "Фамилия: ";  cout << list\_of\_book[i].last\_name.surname << endl;  cout << "Отчество: ";  cout << list\_of\_book[i].dad\_name.dadname << endl;  cout << "Адрес: ";  cout << list\_of\_book[i].direction.address << endl;  cout << "Номер телефона: ";  cout << list\_of\_book[i].phone.number\_phone << endl;  cout << "Место работы или учёбы: ";  cout << list\_of\_book[i].place.place\_work\_or\_study << endl;  cout << "Должность: ";  cout << list\_of\_book[i].position.post << endl;  cout << "Дата рождения: ";  cout << list\_of\_book[i].date << endl;  cout << endl;  }  }  void search()  {  char lastname[50];  cout << "Введите фамилию: ";  cin.ignore();  cin.getline(lastname, 40);  for (int i = 0; i < current\_size; i++) {  if (strcmp(lastname, list\_of\_book[i].last\_name.surname) == 0) {  cout << "Информация о " << i + 1 << " персоне: " << endl;  cout << "Имя: ";  cout << list\_of\_book[i].yourself\_name.name << endl;  cout << "Фамилия: ";  cout << list\_of\_book[i].last\_name.surname << endl;  cout << "Отчество: ";  cout << list\_of\_book[i].dad\_name.dadname << endl;  cout << "Адрес: ";  cout << list\_of\_book[i].direction.address << endl;  cout << "Номер телефона: ";  cout << list\_of\_book[i].phone.number\_phone << endl;  cout << "Место работы или учёбы: ";  cout << list\_of\_book[i].place.place\_work\_or\_study << endl;  cout << "Должность: ";  cout << list\_of\_book[i].position.post << endl;  cout << "Дата рождения: ";  cout << list\_of\_book[i].date << endl;  cout << endl;  }  else  {  cout << "Информации не найдено." << endl;  }  }  }  void infile()  {  ofstream file1("book.txt", ios::app);  if (!file1) {  cout << "Не удалось открыть файл для записи." << endl;  }  for (int i = 0; i < current\_size; i++) {  file1 << "Информация о " << i + 1 << " персоне: " << endl;  file1 << "Имя: ";  file1 << list\_of\_book[i].yourself\_name.name << endl;  file1 << "Фамилия: ";  file1 << list\_of\_book[i].last\_name.surname << endl;  file1 << "Отчество: ";  file1 << list\_of\_book[i].dad\_name.dadname << endl;  file1 << "Адрес: ";  file1 << list\_of\_book[i].direction.address << endl;  file1 << "Номер телефона: ";  file1 << list\_of\_book[i].phone.number\_phone << endl;  file1 << "Место работы или учёбы: ";  file1 << list\_of\_book[i].place.place\_work\_or\_study << endl;  file1 << "Должность: ";  file1 << list\_of\_book[i].position.post << endl;  file1 << "Дата рождения: ";  file1 << list\_of\_book[i].date << endl;  }  file1.close();  cout << "Данные успешно записаны." << endl;  }  void onfile()  {  ifstream file2("book.txt");  if (!file2) {  cout << "Не удалось открыть файл для чтения." << endl;  }  string line;  while (getline(file2, line)) {  cout << line << endl;  }  file2.close();  cout << "Данные успешно прочтены из файлы." << endl;  } |
| **Результат программы** |
|  |

| **№ варианта** | **Условие задачи** |
| --- | --- |
| **10** | **Государство**. Наименование, столица, численность населения, площадь. Выбор государства по столице. Форму правления реализовать с помощью перечисления. |

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <string>  #include <Windows.h>  #define size 50  using namespace std;  void InputInfo();//функция для ввода  void OutputInfo();//функция для вывода  void DeleteInfo();//функция для удаления элементов  void SearchInfo();//функция для поиска  enum Forms  {  Monarchy,  Republic  };  struct Store  {  string name\_country;  string name\_center;  double quantity\_people;  double square\_country;  short government;  };  Store list\_of\_countries[size];  int choice, current\_size = 0, number\_countries;  Forms selected\_form = Monarchy;  int main()  {  system("color 70");  setlocale(LC\_ALL, "RU");  SetConsoleCP(1251);  do {  cout << "Выберите операцию: " << endl;  cout << "1 - Ввод данных." << endl;  cout << "2 - Вывод данных." << endl;  cout << "3 - Поиск информации." << endl;  cout << "4 - Удаление информации." << endl;  cout << "5 - Работа с формами правления." << endl;  cout << "6 - Выxод." << endl;  cout << endl;  cin >> choice;  switch (choice)  {  case 1:  InputInfo();  break;  case 2:  OutputInfo();  break;  case 3:  SearchInfo();  break;  case 4:  DeleteInfo();  break;  }  } while (choice != 6);  }  void InputInfo()//вводим необходимые параметры  {  cout << endl;  cout << "Ввод информации: " << endl;  cout << "Введите количество стран: ";  cin >> number\_countries;  cout << endl;  if (current\_size < size) {  for (int i = 0; i < number\_countries; i++) {  cout << "Введите информацию для " << i + 1 << " страны: " << endl;  cout << "Введите название страны: ";  cin.ignore();  getline(cin, list\_of\_countries[current\_size].name\_country);  cout << "Введите столицу страны: ";  getline(cin, list\_of\_countries[current\_size].name\_center);  cout << "Введите население страны: ";  cin >> list\_of\_countries[current\_size].quantity\_people;  cout << "Введите площадь страны: ";  cin >> list\_of\_countries[current\_size].square\_country;  cout << "Выберите форму правления для государства (0 - Монархия, 1 - Республика): ";  cin >> list\_of\_countries[current\_size].government;  current\_size++;  cout << endl;  }  }  }  void OutputInfo()//выводим учитывая выбранную процентную ставку  {  cout << endl;  cout << "Вывод информации: " << endl;  for (int i = 0; i < current\_size; i++) {  cout << "Информация о " << i + 1 << " стране: " << endl;  cout << "Название страны: ";  cout << list\_of\_countries[i].name\_country << endl;  cout << "Столица страны: ";  cout << list\_of\_countries[i].name\_center << endl;  cout << "Население страны: ";  cout << list\_of\_countries[i].quantity\_people << endl;  cout << "Площадь страны: ";  cout << list\_of\_countries[i].square\_country << endl;  cout << "Форма правления: ";  if (list\_of\_countries[i].government == Forms::Monarchy) { cout << "Монархия" << endl; }  else { cout << "Республика" << endl; }  cout << endl;  }  }  void SearchInfo()//поиск информации по столице  {  cout << endl;  string capital;  cout << "Введите столицу государства: ";  cin.ignore();  getline(cin, capital);  for (int i = 0; i < current\_size; i++) {  if (capital == list\_of\_countries[i].name\_center) {  cout << "Информация о " << i + 1 << " стране: " << endl;  cout << "Название страны: ";  cout << list\_of\_countries[i].name\_country << endl;  cout << "Столица страны: ";  cout << list\_of\_countries[i].name\_center << endl;  cout << "Население страны: ";  cout << list\_of\_countries[i].quantity\_people << endl;  cout << "Площадь страны: ";  cout << list\_of\_countries[i].square\_country << endl;  cout << "Форма правления: ";  if (list\_of\_countries[i].government == Forms::Monarchy) { cout << "Монархия" << endl; }  else { cout << "Республика" << endl; }  }  else {  cout << "Ничего не найдено." << endl;  }  }  }  void DeleteInfo()//удаление строки по номеру в структуре  {  cout << endl;  int NumberLine;  cout << "Введите номер строки, которую хотите удалить: ";  cin >> NumberLine;  if (NumberLine > 0 && NumberLine <= current\_size) {  for (int del = NumberLine - 1; del < current\_size - 1; del++) {  list\_of\_countries[del] = list\_of\_countries[del + 1];  }  current\_size = current\_size - 1;  }  else {  cout << "Неверный номер строки." << endl;  }  } |
| **Результат программы** |
|  |

| **№ варианта** | **Условие задачи** |
| --- | --- |
| **10** | **Горожанин.** Ф.И.О., дата рождения, адрес, пол (м, ж). Реализовать выборку по году рождения. |

|  |
| --- |
| **Код программы** |
| #include <string>  #include <iostream>  #include <fstream>  #include <Windows.h>  #define size 300  using namespace std;  void input();//функция ввода элементов записной книги  void out();//функция вывода содержания записной книги  void search();//функция поиска по содержанию книги  void infile();//функция записи в файл  void onfile();//функция вывода содержимого файла  union Person//обьединение переменных типа int и char  {  int year\_birthday;  char address\_and\_gender\_and\_name[200];  };  struct ForPerson  {  Person name;  Person birthday;  Person point;  Person gender\_person;  };  ForPerson list\_of\_person[size];  int choice, current\_size = 0, number\_persons;  int main()  {  system("color 70");  setlocale(LC\_ALL, "Russian");  SetConsoleCP(1251);  do {  cout << "Выберите операцию:" << endl;  cout << "1 - Ввод информации." << endl;  cout << "2 - Вывод информации." << endl;  cout << "3 - Поиск информации." << endl;  cout << "4 - Запись в файл." << endl;  cout << "5 - Чтение из файла." << endl;  cout << "6 - Выход." << endl;  cin >> choice;  switch (choice)  {  case 1:  input();  break;  case 2:  out();  break;  case 3:  search();  break;  case 4:  infile();  break;  case 5:  onfile();  break;  }  } while (choice != 6);  }  void input()  {  cout << endl;  cout << "Ввод информации: " << endl;  cout << "Введите количество граждан: ";  cin >> number\_persons;  cout << endl;  if (current\_size < size) {  for (int i = 0; i < number\_persons; i++) {  cout << "Введите информацию для " << i + 1 << " гражданина: " << endl;  cout << "Введите ФИО гражданина: ";  cin.ignore();  cin.getline(list\_of\_person[current\_size].name.address\_and\_gender\_and\_name, 190);  cout << "Введите адрес гражданина: ";  cin.getline(list\_of\_person[current\_size].point.address\_and\_gender\_and\_name, 190);  cout << "Введите пол гражданина (м/ж): ";  cin.getline(list\_of\_person[current\_size].gender\_person.address\_and\_gender\_and\_name, 190);  cout << "Введите год рождения гражданина: ";  cin >> list\_of\_person[current\_size].birthday.year\_birthday;  current\_size++;  cout << endl;  }  }  }  void out()  {  cout << endl;  cout << "Вывод информации: " << endl;  for (int i = 0; i < current\_size; i++) {  cout << "Информацию о " << i + 1 << " гражданине: " << endl;  cout << "ФИО гражданина: ";  cout << list\_of\_person[i].name.address\_and\_gender\_and\_name << endl;  cout << "Адрес гражданина: ";  cout << list\_of\_person[i].point.address\_and\_gender\_and\_name << endl;  cout << "Пол гражданина (м/ж): ";  cout << list\_of\_person[i].gender\_person.address\_and\_gender\_and\_name << endl;  cout << "Год рождения гражданина: ";  cout << list\_of\_person[i].birthday.year\_birthday;  cout << endl;  }  }  void search()  {  int year;  cout << "Введите год рождения гражданина: ";  cin >> year;  for (int i = 0; i < current\_size; i++) {  if (year == list\_of\_person[i].birthday.year\_birthday) {  cout << "Информацию о " << i + 1 << " гражданине: " << endl;  cout << "ФИО гражданина: ";  cout << list\_of\_person[i].name.address\_and\_gender\_and\_name << endl;  cout << "Адрес гражданина: ";  cout << list\_of\_person[i].point.address\_and\_gender\_and\_name << endl;  cout << "Пол гражданина (м/ж): ";  cout << list\_of\_person[i].gender\_person.address\_and\_gender\_and\_name << endl;  cout << "Год рождения гражданина: ";  cout << list\_of\_person[i].birthday.year\_birthday;  cout << endl;  }  else {  cout << "Информация не найдена." << endl;  }  }  }  void infile()  {  ofstream file1("person.txt");  if (!file1) {  cout << "Не удалось открыть файл." << endl;  }  for (int i = 0; i < current\_size; i++) {  file1 << "Информацию о " << i + 1 << " гражданине: " << endl;  file1 << "ФИО гражданина: " << list\_of\_person[i].name.address\_and\_gender\_and\_name << endl;  file1 << "Адрес гражданина: " << list\_of\_person[i].point.address\_and\_gender\_and\_name << endl;  file1 << "Пол гражданина (м/ж): " << list\_of\_person[i].gender\_person.address\_and\_gender\_and\_name << endl;  file1 << "Год рождения гражданина: " << list\_of\_person[i].birthday.year\_birthday;  }  file1.close();  cout << "Данные успешно записаны." << endl;  }  void onfile()  {  ifstream file2("person.txt");  if (!file2) {  cout << "Не удалось открыть файл." << endl;  }  string line;  while (getline(file2, line)) {  cout << line << endl;  }  file2.close();  cout << "Данные успешно прочтены из файла." << endl;  } |
| **Результат программы** |
|  |