Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЁТ**

**По**

**индивидуальному заданию №8**

Дисциплина/Профессиональный модуль: Операционные системы

Выполнил студент: Гуцалов

Группы: 2ОИБАС-1322

Преподаватель

Сибирев И.В.

Оценка за работу:\_\_\_\_\_\_\_

**Москва – 2024г.**

**ЛАБОРАТОРНАЯ РАБОТА №7**

**ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ СТРУКТУР**

**Задание**

В магазине сформирован список постоянных клиентов, который вклю-

чает ФИО, домашний адрес покупателя и размер предоставляемой скидки.

Удалить из этого списка все повторяющиеся записи, проверив ФИО и домаш-

ний адрес покупателя.

|  |
| --- |
| #include <iostream>  #include <string>  using namespace std;  int main()  {  setlocale(LC\_ALL,"RU");  struct mspok  {  char fio[40];  char adr[40];  int skidka;  }spok[100];  int j, i, n, k;  cout << "Введите кол-во клиентов:";  cin >> n;  cout << endl;  for (i = 0; i < n; i++)  {  cout << "Введите ФИО клиента:";  cin >> spok[i].fio;  cout << "Введите адрес клиента:";  cin >> spok[i].adr;  cout << "Введите скидку клиента:";  cin >> spok[i].skidka;  cout << endl;  }  for (i = 0; i < n - 1; i++)  for (j = i + 1; j < n; j++)  {  if ((strcmp(spok[i].fio, spok[j].fio) == 0)) (strcmp(spok[i].adr, spok[j].adr) == 0)  ; {  for (k = j; k < n; k++)  {  spok[k] = spok[k + 1];  n--;  }  }  ;  }  for (i = 0; i < n; i++)  cout << spok[i].fio << "" << spok[i].adr << "" << spok[i].skidka << endl;  return 0;  } |

**ЛАБОРАТОРНАЯ РАБОТА №8**

**ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ ФУНКЦИЙ**

Задание

|  |
| --- |
| #include <iostream>  #include <cmath>  #include <iomanip>  using namespace std;  typedef double (\*uf)(double, double, int&);  void tabl(double, double, double, uf);  double y(double, double, int&);  double s(double, double, int&);  int main()  {  cout << setw(8) << "x" << setw(15) << "y(x)" << setw(10) << "k" << endl;  tabl(-1.5,1.5,0.001,y);  cout << endl;  cout << setw(8) << "x" << setw(15) << "s(x)" << setw(10) << "k" << endl;  tabl(-1.5, 1.5, 0.001, s);  return 0;  }  void tabl(double a, double b, double h, double eps, uf fun)  {  int k = 0;  double sum;  for (double x = a; x < b + h / 2; x += h)  {  sum = fun(x, eps, k);  cout << setw(8) << x << setw(15) << sum << setw(10) << k << endl;  }  }  double y(double x, double eps, int& k)  {  return sin(x);  }  double s(double x, double eps, int& k)  {  double a, c, sum;  sum = a = c = x;  k = 1;  while (fabs(c) > eps)  {  c = cos(k \* x) / k;  a \*= c;  sum += a;  k++;  }  return sum;  } |

**ЛАБОРАТОРНАЯ РАБОТА №10**

**ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ ФАЙЛОВ**

|  |
| --- |
| #include <iostream>  #include <string>  #include <iomanip>  #include <ctime>  using namespace std;  struct date {  int day;  int month;  int year;  };  struct goods {  wstring name;  unsigned quantity;  float price;  date dt;  };  typedef goods\* PGOODS;  void ru();  PGOODS create\_list(PGOODS, const size\_t);  void show\_list(PGOODS, const size\_t);  void print\_record(PGOODS);  void select\_list(PGOODS, const size\_t, const int \_month = 1, const float \_price = 100000.0f);  void sort\_list(PGOODS, const size\_t, const size\_t);  void uswap(goods&, goods&);  int main() {  ru();  wcout << L" Введите размер списка: ";  size\_t size;  cin >> size;  PGOODS ptr = NULL;  if ((ptr = create\_list(ptr, size)) != NULL) {  system("cls");  show\_list(ptr, size);  select\_list(ptr, size);  delete[] ptr;  ptr = NULL;  }  cin.get(); cin.get();  return 0;  }  PGOODS create\_list(PGOODS \_goods, const size\_t \_size) {  \_goods = new goods[\_size];  if (\_goods) {  for (size\_t n = 0; n < \_size; n++) {  fflush(stdin);  wcout << L"\n Введите наименование товара: ";  getline(wcin, \_goods[n].name);  wcout << L" Введите количество товара: ";  cin >> \_goods[n].quantity;  wcout << L" Введите цену товара: ";  cin >> \_goods[n].price;  wcout << L" Введите число поступления товара: ";  cin >> \_goods[n].dt.day;  wcout << L" Введите месяц поступления товара: ";  cin >> \_goods[n].dt.month;  wcout << L" Введите год поступления товара: ";  cin >> \_goods[n].dt.year;  }  }  else wcout << L" Не удалось создать список!\n";  return \_goods;  }  void show\_list(PGOODS \_goods, const size\_t \_size) {  if (\_goods) {  PGOODS next = \_goods;  for (size\_t n = 0; n < \_size; n++) print\_record(next++);  }  else wcout << L" Список не существует!\n";  cout << endl;  }  void print\_record(PGOODS \_goods) {  streamsize sign = 2, name = 36, quantity = 12, price = 14, date = 4;  wcout << setw(name) << left << \_goods->name;  cout << setw(quantity) << right << \_goods->quantity  << setw(price) << fixed << setprecision(sign) << \_goods->price;  cout << setw(date);  if (\_goods->dt.day < 10) cout << '0';  cout << \_goods->dt.day << '.';  if (\_goods->dt.month < 10) cout << '0';  cout << \_goods->dt.month << '.'  << \_goods->dt.year << endl;  }  void sort\_list(PGOODS \_goods, const size\_t \_beg, const size\_t \_end) {  int n = (int)\_beg, m = (int)\_end;  goods mid;  mid.name = \_goods[(\_beg + \_end) >> 1].name;  do {  while (\_goods[n].name < mid.name && n < (int)\_end) ++n;  while (\_goods[m].name > mid.name && m > (int)\_beg) --m;  if (n <= m) {  if (n < m) uswap(\_goods[n], \_goods[m]);  ++n;  --m;  }  } while (n <= m);  if ((int)\_beg < m) sort\_list(\_goods, \_beg, m);  if (n < (int)\_end) sort\_list(\_goods, n, \_end);  }  void uswap(goods& \_a, goods& \_b) {  goods temp = \_a;  \_a = \_b;  \_b = temp;  }  void select\_list(PGOODS \_goods, const size\_t \_size, const int \_month, const float \_price) {  wcout << L"\n\t\t\t\tОтсортированный запрос:\n\n";  bool empty = true;  if (\_goods && \_size) {  const time\_t term = \_month \* 31 \* 24 \* 60 \* 60;  if (\_size > 1) sort\_list(\_goods, 0, \_size - 1);  time\_t t;  tm\* ctm;  for (size\_t n = 0; n < \_size; n++) {  tm current = { 0, 0, 0, \_goods[n].dt.day, \_goods[n].dt.month - 1, \_goods[n].dt.year - 1900 };  time\_t beg = mktime(&current);  time(&t);  ctm = localtime(&t);  time\_t end = mktime(ctm);  if (end - beg > term && \_price < \_goods[n].price) {  print\_record(&\_goods[n]);  if (empty) empty = false;  }  }  }  if (empty) wcout << L"\n\tДанные отсутствуют!\n";  }  void ru() {  wcout.imbue(locale("rus\_rus.866"));  wcin.imbue(locale("rus\_rus.866"));  } |