**ЗАДАНИЕ №1**

**Программа: Классы. Полиморфизм на языке С++**

|  |  |  |
| --- | --- | --- |
| |  | | --- | |  |   **Задание**  На основе прежде разработанного класса управления двонапрямленим списком обеспечить возможность хранения разнотипных данных.  **Выбор алгоритма решения задачи**  Для решения этой задачи надо использовать базовый класс и 2 класса-потомка в которых все данні сделать закрытыми (для того чтобы при работе случайно не повредить их) и методы обработки этих данных. Уместным я считаю сделать еще один класс с методами: добавление, удаление одного или всех элементов, вывод списка на экран, а также [универсальную](http://click01.price.ru/click.jsp?url=41foRxstD*CBszUbM*LOtSHz7CYcZXEXhAqqIhukzzQHlgfrqUn5swwf2YgNnDpccAV2JS6CaYzLg4wFUThEnz82FhHprGpEIq95wy4lpI7lQIMytLgf-2fk6l1RYSTN0S9AICA80L6WcvKnG3Gs0FVDse6Amw*0K5wwPAbnLwd14HCSnamsD3atCsf1QLsDLlb0BfQxGMnDgsCU*CkfASqY2akMdWhTmpDCD3uKMgpO6xgVTrnCW2-r-4SDWAPLoVqG1bLhp9DKtJv5wtojJASp5RUHrTCs8oe*iBX4*tPiacwb1qS9owraO1CMcMdhPjw66UKR8eSH-K5DuwsmFkoHjk3HON-*0R4z*fSdTQmbPGdAmjeQZL9HG8U9vnB5vRoFTMTzg58POX4Kf*0zxYQUHPLhVohZ1GEdA0wjuOQ02Bay6et5lbEhzOANydLK1WgQXbcL6s81s-NNX77wUK-0X62LAyM9Pq399oiN07zBvvBjOzM7X5uC1bi4*R91A7Pyy2AGehICImkEua7arrXlMeFkPQdSAlIIEchTP3Z0FDg5T2Hsx14jWm6zRGyhIGQ3xIIM93iJdMkEEk39R2h7aEsOzamSV9**SMW-01uUY52IK6*5jCZ5rb-bfIF*WeRXiU63KoB0AdoSQ2Ekw2VSSLZudBo567bOFm8FM3JCZEsxsq2NWYRyiU1d29IPIIYajkBIpVta8HEM16jcI9v6jznBxnGERZ*uYqwCPqzqWylVP0pzVFVMtTS-RUlN97ZmoNwgqmRgaf1QyAXJ24hD3RU9OL70dJY223SkCq-K0sCgNeo-XNO7MJUBhT3aA6I6vNkL6O1ar0pYOxzbdeTdi5oOQViE6ZTRHBDeNSAO0WxTo3oKTHZDKCDI49uoT0tsNn8LXSdDLhYsJW0q0TDtu9OBsAwme63H*atMEJj4EYUuPCKv453i89mr*qwHf2VqHHcIAITkdECJ-pauXYA-u0XsNDW*l8o3**iUxiIU88wpIL6CVIzkDr4t9HZKDSlGqaRznhZ5YMgjMW7o4G9P9M1SL2d*h1B2Bnuq0dpG0Jd6gI3DehbtIYpTDIjNAxGMiIglwfi8U0pZcWJizmMALAO4dwAwjagXI7lt4gEMiWFWed8WpUzCWd3ETeSmNze22mTEwx1abYE41oSakM*waiqQyhYWO4bqNwL7cyICAAAAuty8CgAAAAA&eurl%5B%5D=oQo5pzerWyl8gH-5YyBcDiqj9RfixUvUrWfgRTko8eT3oKEIIxmHmROJpwlZ*ajmYMJ4W3y-t9hGeinajHvrrgIAAAC63LwKAAAAAA) функцию для добавления и удаление: функция поиска.  **Описание алгоритма решения задачи**  Основа программы это функция main, поскольку в ней находится меню пользователя из которого можно задействовать любую из функций. Меню всегда находится на экране как в начале работы так и после окончания работы одной из функций, завершение программы происходит при нажатии клавиши Esc.  Удаление всего списка происходит благодаря поочередному переходу с начала до конца списка с одновременным удалением активного элемента.  Добавление/удаление отдельного элемента происходит в два этапа, после того как пользователь выберет место добавления/удаление элемента, функция поиска возвратит показатель на тот элемент который забав пользователь, а потом функция добавления/удаление разорвет цепь и прибавит/удалит элемент. Функция вывода таблицы поочередно перебирает все элементы списка и выводит список на экран.  **Схему алгоритма**[см. здесь](http://life-prog.ru/view_programs.php?id=32&language=c&page=1)  Функции tabl, find, add и del\_all полностью аналогичные функциям с лабораторной работе №7.  **Описание программы -**[см. здесь](http://life-prog.ru/view_programs.php?id=33&language=c&page=1)  Функция tabl: выводит таблицу на экран монитора в которой отображаются все элементы списка.  **Текст программы**  **Head.h**  #ifndef \_HEAD\_H\_ #define \_HEAD\_H\_ #include <stdio.h> #include <iostream> #include <conio.h> #include <time.h>  using namespace std;  const int hr=8; const int NAME=19;  class Base           //базовый класс { protected:           //защищенные данные char fio[NAME+1]; unsigned int year; public:              //открытые данные Base \*prev, \*next;      //указатель на следующий и предыдущий virtual void view(){}   //виртуальная функция вывода };  class Stud : public Base   //класс Stud наследник от Base { char grup[10];                     //поля группа unsigned int stip;                 //и стипендия public: void add(Stud \*p, Base \*find);     // функция добавления add  void view();                       // функция вывода };  class Prep : public Base               //класс Prep наследник от Base { unsigned int stag, zarp;           //поля stag и zarp  public: void add(Prep \*p, Base \*find);     // функция добавления add void view();                       // функция вывода  };  class mtd                              // класс mtd, который не имеет наследования { public:                                //открытые функции Base \*Find(int n);                 //поиск void tabl();                       //вывод таблицы void del(Base \*find);              //удаление void del\_all();                    //удалить все }; #endif    **6.2 Func.cpp**  #include "Head.h"                      //подключение файла Head.h  int KEY, COUNT; Base \*FIRST;  Base \* mtd::Find(int n)                //реализация функции mtd для поиска  {                                      //нужного места в списке Base \*temp; temp=FIRST; int i=0; Base \*p; if (n>0) { if (temp==NULL)              //если список еще не создан  { KEY=1; return temp; } else { if (n==1)               //если введена первая позиция  { KEY=1; return temp; } else        //иначе введено число больше чем 1  { while (i<n-1)     //пока не достигнут конец списка { p=temp; if (p->next==NULL)   //если уже конец списка  {                                   i=n; KEY=0; return p;   //возврат указателя на посдедний элемент списка  } else    //если не конец списка  { temp=p->next; i++; } } KEY=1; return temp; } } } else       //иначе введено n<1 список не может быть создан  { KEY=-1; cout<<"Zadanoe chislo ne vhodit v predel spiska...\a\n"; } return FIRST; }  //////////////////////////////////////////////////////////////////////// void Stud::add(Stud \*p, Base \*find)    //функция добавления в класе Stud  { if (KEY>=0) // если key>=0 тогда можна добавить список  { if (find==NULL)   //если списка еще нет  { p->prev=NULL; FIRST=p; p->next=NULL; } else { if (KEY==0)       //если список достигнут конца { p->prev=find; p->next=NULL; find->next=p; } else              //иначе позиция не достигла конца { if (find==FIRST!=NULL) //если fin==first!=0 то это первый список  {  p->prev=NULL; p->next=FIRST; find->prev=p; FIRST=p; } else { if (find!=FIRST) //это не первый список. Добавлять нужно в середине  {                 //списка  p->prev=find->prev; p->next=find; p->prev->next=p; find->prev=p; } } } } } srand(unsigned(time(NULL))); int y,s,flag=0,key=0; char ch; cout<<"1. Vvesti FIO\n"; cout<<"2. Sgenerirovat' FIO\n\n"; while (!key) { key=0; ch=getch(); switch(ch)  //выбор ввода данных: вручную или сгенерировать { case '1': cout<<"Vvedite FIO "; cin>>fio; cout<<endl; key=1; break; case '2': for (s=0;s<NAME;s++) fio[s]=char((rand()%26)+64); fio[s]='\0'; key=1; break; } } year=(rand()%10)+1980; for (s=0;s<10;s++) grup[s]=char((rand()%26)+64); grup[s]='\0'; stip=(rand()%300)+500; }  //////////////////////////////////////////////////////////////////////// void Prep::add(Prep \*p, Base \*find)    //функция добавления в класе Prep {                                      // аналогичная функция добавления в класе Stud  if (KEY>=0) { if (find==NULL) { p->prev=NULL; FIRST=p; p->next=NULL; } else { if (KEY==0) { p->prev=find; p->next=NULL; find->next=p; } else { if (find==FIRST!=NULL) { p->prev=NULL; p->next=FIRST; find->prev=p; FIRST=p; } else { if (find!=FIRST) { p->prev=find->prev; p->next=find; p->prev->next=p; find->prev=p; } } } } } srand(unsigned(time(NULL))); int y,s,flag=0,key=0; char ch; cout<<"1. Vvesti FIO\n"; cout<<"2. Sgenerirovat' FIO\n\n"; while (!key) { key=0; ch=getch(); switch(ch) { case '1': cout<<"Vvedite FIO "; cin>>fio; cout<<endl; key=1; break; case '2': for (s=0;s<NAME;s++) fio[s]=char((rand()%26)+64); fio[s]='\0'; key=1; break; } } year=(rand()%45)+1940; stag=2009-year-25; zarp=(rand()%300)+500; }    //////////////////////////////////////////////////////////////////////// void mtd::tabl()     // вывод таблицы на экран { Base \*temp; temp=FIRST; int key=1,x=0; if (FIRST!=NULL) { printf(" -------------------------------------------------------------\n"); printf("|     |       |      |      |          |          |           |\n"); printf("|  No |   FIO  | Year | Stag |  Gryppa  | Zarplata | Stipendia |\n"); printf("|-----|-------|------|------|----------|----------|-----------|\n"); while (key) { ++x; printf("|%5d|",x); temp->view(); cout<<endl; if (temp->next==NULL) key=0; else temp=temp->next; } printf(" ----------------------------------------------------------\n\n"); } else cout<<"\nNe zadano kolichestvo rabochih\n"; }  //////////////////////////////////////////////////////////////////////// void Stud::view()    //функция выводу информации о студентах { printf("%20s|%6d|      |%10s|          |%11d|",fio,year,grup,stip); }  //////////////////////////////////////////////////////////////////////// void Prep::view()    //функция выводу информации о преподавателях { printf("%20s|%6d|%6d|          |%10d|           |",fio,year,stag,zarp);            }  //////////////////////////////////////////////////////////////////////// void mtd::del\_all()  //функция удалить все { Base \*p; int flag=1; if (FIRST!=NULL) { if (flag!=0) { p=FIRST; if (p->next==0) flag=0; else FIRST=p->next; delete p; } } FIRST=NULL; }  //////////////////////////////////////////////////////////////////////// void mtd::del(Base \*find)  //функция удались 1 элемент списка { Base \*p, \*temp; if (KEY>=0) { if (find==0)      //если список пуст, то нечего удалять  { cout<<"Spisok pyst!\n"; getch(); } else { if (KEY==0)    //если в списке нет такого элемента  { cout<<"Takogo elementa nety v spiske!"; getch(); } else { if (find==FIRST!=0)    //если это первый элемент  { if (FIRST->next==0)   //и если он последний  { p=FIRST; delete p; FIRST=0; } else    //иначе он не последний { p=FIRST; FIRST=p->next; FIRST->prev=0; delete p; } } else        //если это не первый элемент  { if (find!=FIRST)   { p=find; if (p->next==0)  //если он последний  { p->prev->next=0; delete p; } else    //иначе он не последний  { find->prev->next=p->next; p->next->prev=find->prev; delete p; } } } } } } }    **6.3 Main.cpp**  #include "Head.h"  extern Base \*FIRST; mtd \*fun;  void main(void) { mtd \*p; p=new mtd; int end=0; char ch; while(!end) //пока не нажата клавиша ESC { system("cls"); cout<<"1.   Dobavit' v spisok\n"; cout<<"2.   Pokazat' spisok\n"; cout<<"3.   Ydalit' [**element**](http://life-prog.ru/view_programs.php?id=34&language=c&page=1)\n"; cout<<"4.   Ydalit' spisok\n"; cout<<"Esc. Exit\n"; ch=getch(); switch (ch) // оператор выбора { case '1': system("cls"); cout<<"1. Stydent\n"; cout<<"2. Prepodavatel'\n"; int n; char chr; chr=getch(); switch (chr) { case '1': cout<<"\nVvedite piziciy - "; cin>>n; Stud \*s; s=new Stud; s->add(s, p->Find(n)); break; case '2': cout<<"\nVvedite piziciy - "; cin>>n; Prep \*pr; pr=new Prep; pr->add(pr, p->Find(n)); break; } break; case '2': system("cls"); if (FIRST) p->tabl(); else cout<<"Spisok pyst...\n"; getch(); break; case '3': system("cls"); if (FIRST) { cout<<"\nVvedite nomer elementa - "; cin>>n; p->del(p->Find(n)); } else { cout<<"Spisok pyst...\n"; getch(); } break; case '4': system("cls"); if (FIRST) p->del\_all(); else { cout<<"Spisok pyst...\n"; getch(); } break; case 27:    //выход из программы end=1; break; } } p->del\_all();     //освобождение динамической памяти delete p; }    **Результат работы программы**  **http://life-prog.ru/programs/10c_clip_image002.jpg**    **http://life-prog.ru/programs/10c_clip_image004.jpg**   |  | | --- | | Просмотров: 9592 | |

**ЗАДАНИЕ №2**

**Готовая программа работа с массивами в С++**

|  |  |
| --- | --- |
| |  | | --- | |  |   **Задание**  В двухмерном массиве сделать перестановку элементов первой четверти квадратной матрицы с элементами четвертой четверти той же матрицы.  **Выбор алгоритма решения задачи**  В этой задаче надо применять двухмерный массив. И с помощью цикла, вносить числа. Потом сделать условие так, чтобы программа знала, где первая четверть, а где четвертая четверть и переставить значение. Для этого нужен будет еще один двухмерный массив в котором будет храниться копия масива.  **Описание решения задачи**  Данную задачу можно развязать следующим способом: делим значение массива на 2 (массив должен быть четным). Потом выбираем переменные x, в которых будет записано данное число. Это нам дает разбить двухмерный массив на четверти. Дальше мы делаем копию массива для того, чтобы переставить значение матрицы. И с помощью cout выводим значение массива.  **Схема алгоритма решения задачи**  http://life-prog.ru/programs/2c_clip_image002.jpg  **Текст програми**  *#include<iostream.h>* *#include<conio.h>*  *const int m = 4;*  *void printArray ( int [][m] );*  *int main ()* *{* *int array1[m][m] = {1,1,5,6,3,3,4,4,5,5,6,6};* *int array2[m][m] = {1,1,5,6,3,3,4,4,5,5,6,6};* *int x;* *x = (m)/2;*  *printArray (array1);*  *for (int i = 0; i<m; i++)* *for (int j = 0; j<m; j++) {* *if  (i<x && j<x)* *array1 [i + x][j + x] = array1[i][j];* *if (i >= x && j >= x)* *array1 [i - x][j - x] = array2[i][j];* *}* *printArray(array1);*  *getch();* *return 0;* *}*  *void printArray (int a[][m])* *{* *for (int m1 = 0; m1<m; m1++) {* *for (int m2 = 0; m2<m; m2++)* *cout << a[m1][m2] << ' ';*  *cout << endl;* *}* *cout <<"\n";* *}*  **Результати роботи програми**  **http://life-prog.ru/programs/2c_clip_image002_0000.jpg** |

**ЗАДАНИЕ №3**

|  |
| --- |
|  |

**Задание**

Дано 20 вещественных чисел. Найти разницу между минимальным и максимальным из них.

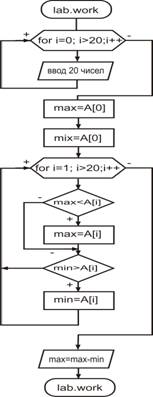
**Выбор алгоритма решения задачи**

В этой задаче надо применять массив. И с помощью цикла, вносить 20 чисел. Потом сравнивать их и найти максимальное и минимальное число. И отнять максимальное от минимального.

**Описание решения задачи**

Создадим массив из вещественных чисел типа float. Вводим 20 чисел в этот массив. Проверяем на максимальное и минимальное число с помощью условия: если предыдущее число меньше существующего, тогда существующее записывается в max, аналогично наоборот: если существующее меньше предыдущего, тогда существующее записывается в min. Потом вычитаем max от min и получаем разницу.

**Схема алгоритма решения задачи**

****

**Текст программы**

// programa nahojdeniya raznici (max-min)  
#include<iostream.h>  
#include<conio.h>  
#include<stdlib.h>

void printArray ( int [] );

int main()  
{  
   clrscr();  
   int x[20],  
             max,  
             min;

   for (int i = 0; i<20; i++)  
              x[ i ] = rand() / 1000;

   max = x[ 0 ];  
   min = x[ 0 ];

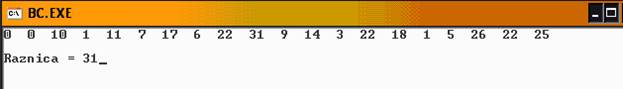
   for (int j = 1; j<20; j++) {  
              if (x[ j ] < min)  
                         min = x[ j ];  
              if (x[ j ] > max)  
                         max = x[ j ];  
   }  
   printArray( x );  
   cout << "\n" << "Raznica = " << max - min;

   getch();  
}

void printArray ( int a[] )  
{  
   for (int i = 0; i < 20; i++)  
              cout << a[ i ] <<"  ";  
   cout << endl;  
}

**Результаты работы программы**

В результате программа внесет в массив 20 чисел и найдет разницу между max и min числами данного массива.  
При компиляции получилось:



|  |
| --- |
| Просмотров: 11344 |

**Задание**

Вывести в левую часть экрана дисплея изображения прямоугольника и сохранить его в файл.

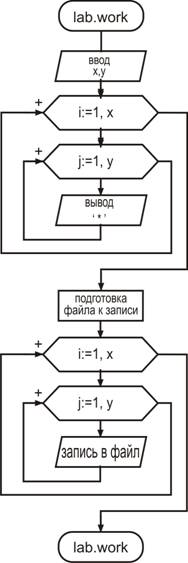
**Выбор алгоритма решения задачи**

В этой задаче нужны некоторые переменные, в которых мы будем вводить информацию на экран дисплея, потом создать файл, в который мы будем записывать информацию.

**Описание решения задачи**

Данную задачу можно решить следующим способом: объявляем две переменные, в которых мы будем записывать количество горизонтальных и вертикальных точек прямоугольника; символьную переменную. С помощью этих переменных можно сделать фигуру. Потом с помощью команды ofstream создаем файл и записываем у него символьную информацию.

**Схема алгоритма решения задачи**



**Текст программы**

**1. Head.h**

#ifndef \_\_lab3\_\_  
#define \_\_lab3\_\_

#include<iostream.h>  
#include<iomanip.h>  
#include<conio.h>  
#include<stdlib.h>  
#include<fstream.h>

#endif

**2. Main.cpp**

#include "head.h"

void printT (int, int );

int main (){  
   int x,y,  
                   b=1;

   cout<<"*\n\nVvedite kolich. x , y :* ";  
   cin>>x>>y;  
   while (b==1) {  
                  if (x>39 || y>22) {  
                                cout<<"*ERROR!!! Previshen limit!!!*";  
                                cout<<"*\nVvedite kolich. x , y :* ";  
                                cin>>x>>y;}  
                  else b = 0;  
                  }  
   printT(x,y);

   getch();  
   return 0;

}

**3. Func.cpp**

#include "head.h"

void printT (int X, int Y){

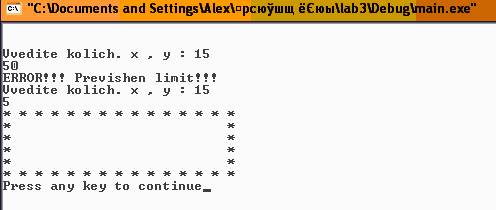
   ofstream outClientFile ("Files.dat" );  
   for (int i = 0; i < X; i++) {  
                  cout<<"\* ";  
                  outClientFile << "\* ";  
   }

   cout<<"\n";

   for (int j = 1; j < Y; j++) {  
                  cout<<"\*"<<setw(2\*X-1)<<"*\*\n*";  
                  outClientFile<<"\*"<<setw(2\*X-1)<<"\**\n*";  
   }

   for (int k = 0; k < X; k++) {  
                  cout<<"\* ";  
                  outClientFile << "\* ";  
   }  
   cout <<"*\n*";  
}

**Результат работы программы**

****

**ЗАДАНИЕ №3**

Разбить слово по слогам . Между слогами вставить тире.

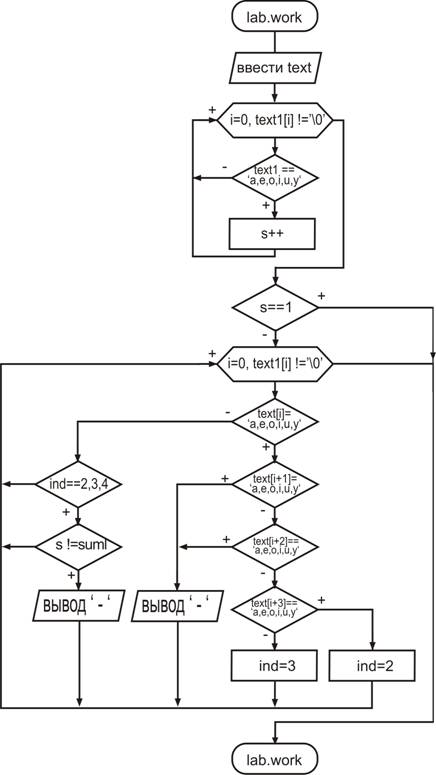
**Выбор алгоритма решения задачи**

Для решения этой задачи, необходимо оценить условие разбивания по слогам. Потом по этим условиям разбить слово.

**Описание решения задачи**

Данную задачу можно развязать следующим способом: объявляем переменную и вносим у нее слово. Потом делаем цикл и считаем сколько там гласных в переменную s. Если 1, тогда мы не разбиваем, а сразу переходим на конец. Если гласных больше, чем одна, тогда входим в цикл. Проверяем условие данная буква гласная, или нет. Если гласная, тогда мы увеличиваем переменную sum на 1 и проверяем есть ли следующая буква первой или второй гласной буквой, если так, тогда мы ставим '-', если нет, то проверяем есть ли гласная третья, четвертая, пятая, если так, тогда переменная ind = 2, 3 или 4. В следующем цикле мы проверяем есть ли данная буква гласной, если нет, тогда проверяем ind = 2, 3 или 4 и сумма гласных не равняется сумме у цикла, то есть s != sum, если так, тогда ставим '-'.

**Схема алгоритма решения задачи**

****

**Текст программы**

**1 Head.h**

// razbivka po slogas  
#ifndef TEXT\_H  
#define TEXT\_H

// class pazbivka texta po slogam  
class text {  
public:  
text ();  
void setText ( char \* );  
void setProb (char \*);  
void printText ();  
private:  
char \*slovo;  
};

#endif

**6.2 Prog.cpp**

#include <iostream.h>  
#include <conio.h>

#include "head.h"

int main ()  
{  
char \*sPtr = "alexander";  
text t;

cout << "Slovo : "<<sPtr<<"\n";  
cout << "Razbitoe po slogam slovo: ";  
t.setProb ( sPtr );  
t.printText();

            cout << endl;  
getch();  
return 0;  
}

**6.3 Func.cpp**

#include <iostream.h>

#include "head.h"

text::text() { slovo = 0;  }

void text::setText ( char \*s ) {  
slovo = s;  
}

void text::setProb (char \*text1) {  
int s = 0, sum = 0, ind = 1;  
for (int i=0; text1[i] != '\0'; i++)  
if ((text1[i] == 'a') || (text1[i] == 'o') ||(text1[i] == 'u')||   
(text1[i] == 'e') || (text1[i] == 'y') || (text1[i] == 'i'))  
s++;

if (s == 1)  
cout << text1;

            if (s > 1) {  
for (int i=0; text1[i] !='\0'; i++) {  
cout << text1[i];  
if ((text1[i] == 'a') || (text1[i] == 'o') ||(text1[i] == 'u')||   
(text1[i] == 'e') || (text1[i] == 'y') || (text1[i] == 'i')) {  
sum++;  
if ((text1[i+1] == 'a') || (text1[i+1] == 'o') ||(text1[i+1] == 'u')||   
(text1[i+1] == 'e') || (text1[i+1] == 'y') || (text1[i+1] == 'i'))  
cout <<"-";  
else if ((text1[i+2] == 'a') || (text1[i+2] == 'o') ||(text1[i+2] == 'u')||   
(text1[i+2] == 'e') || (text1[i+2] == 'y') || (text1[i+2] == 'i'))  
cout << "-";  
else if ((text1[i+3] == 'a') || (text1[i+3] == 'o') ||(text1[i+3] == 'u')||   
(text1[i+3] == 'e') || (text1[i+3] == 'y') || (text1[i+3] == 'i'))  
ind = 2;  
else if ((text1[i+4] == 'a') || (text1[i+4] == 'o') ||(text1[i+4] == 'u')||   
(text1[i+4] == 'e') || (text1[i+4] == 'y') || (text1[i+4] == 'i'))  
ind = 3;  
else ind = 4;  
}  
else if ((ind == 2)|| (ind == 3) || (ind == 4)) {  
if (s != sum) cout <<"-";  
ind = 0;  
}  
}  
}  
}

void text::printText() {  
cout << slovo;  
}

**Результати роботи програми**

alexander

a-le-xan-der

**адание**

Построить строку простых чисел, на которые можно делить трехзначительное целое число. Выяснить сумму строки.

**Выбор алгоритма решения задачи**

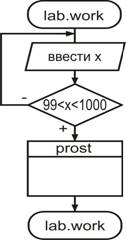
Для решения данной задачи, необходимо сделать цикл от 1 к трехзначительному целому числу, и в этом циклу сделать цикл, в котором будем проверять условие "простое данное число или нет". И потом найти сумму этих простых чисел.

**Описание решения задачи**

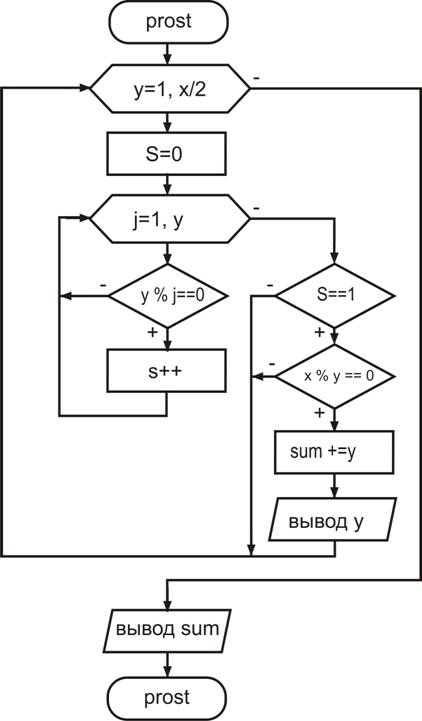
Для данной задачи необходимо переменные типа int. Мы вводим трехзначительное число, делаем цикл в цикле. В первом цикле проходим числа от 1 (в=1) к x/2 (то есть трехзначительное целое число). И каждое из этих чисел делим от 1 к y/2 и проверяем есть ли это число простым. Если это число является простым, мы трехзначительное число делим на это простое число, и если остаток от деления равняется нулю, тогда мы выводим это число на экран и множим это число на такие же числа, которые делятся на трехзначительное число.

**Схема алгоритма решения задачи**

**1 Схема главной программы**



**2 Схема подпрограммы**

****

**Текст программы**

#include <iostream.h>  
#include <conio.h>

void prost (int );

int main ()  
{  
int x = 1;

// ocnovnaya prog  
while (x<100 || x>999){  
cout <<"vvedite trehznachnoe chislo";  
cin >> x;  
}  
  
prost(x);  
getch();  
  
return 0;  
}

void prost(int ch){

   int s, sum = 0;

   cout <<"Prostuye chisla: 1\t";  
int usl = ch/2;  
int usl\_2;

   for (int y=2; y<usl; y++) {  
s=0;  
usl\_2 = y;

       for (int j=1; j<usl\_2; j++)   
if (y % j == 0) s++;

       if (s==1 && ch % y == 0) {  
cout <<y<<"\t";  
sum += y;  
}     
}  
cout <<"\nSumma = "<<sum+1;  
}

**Результат работы программы**

vvedite trehznachnoe chislo  
912  
1 2 3 19  
sum = 25

|  |
| --- |
| Просмотров: 10690 |

**ЗАДАНИЕ №4**

**Программа на тему функции в языке Си**

|  |  |  |
| --- | --- | --- |
| |  | | --- | |  |   **Задание**  Построить строку простых чисел, на которые можно делить трехзначительное целое число. Выяснить сумму строки.  **Выбор алгоритма решения задачи**  Для решения данной задачи, необходимо сделать цикл от 1 к трехзначительному целому числу, и в этом циклу сделать цикл, в котором будем проверять условие "простое данное число или нет". И потом найти сумму этих простых чисел.  **Описание решения задачи**  Для данной задачи необходимо переменные типа int. Мы вводим трехзначительное число, делаем цикл в цикле. В первом цикле проходим числа от 1 (в=1) к x/2 (то есть трехзначительное целое число). И каждое из этих чисел делим от 1 к y/2 и проверяем есть ли это число простым. Если это число является простым, мы трехзначительное число делим на это простое число, и если остаток от деления равняется нулю, тогда мы выводим это число на экран и множим это число на такие же числа, которые делятся на трехзначительное число.  **Схема алгоритма решения задачи**  **1 Схема главной программы**  http://life-prog.ru/programs/5c_clip_image002.jpg  **2 Схема подпрограммы**  **http://life-prog.ru/programs/5c_clip_image002_0000.jpg**    **Текст программы**  #include <iostream.h> #include <conio.h>  void prost (int );  int main () { int x = 1;  // ocnovnaya prog while (x<100 || x>999){ cout <<"vvedite trehznachnoe chislo"; cin >> x; }  prost(x); getch();  return 0; }  void prost(int ch){     int s, sum = 0;     cout <<"Prostuye chisla: 1\t"; int usl = ch/2; int usl\_2;     for (int y=2; y<usl; y++) { s=0; usl\_2 = y;         for (int j=1; j<usl\_2; j++)  if (y % j == 0) s++;         if (s==1 && ch % y == 0) { cout <<y<<"\t"; sum += y; }    } cout <<"\nSumma = "<<sum+1; }  **Результат работы программы**  vvedite trehznachnoe chislo 912 1 2 3 19 sum = 25   |  | | --- | | Просмотров: 10691 | |

**ЗАДАНИЕ №5**

**Программа: Обработка символов и ввод-вывод данных на языке C++**

|  |  |
| --- | --- |
| |  | | --- | |  |   **Задача**  Выяснить наиболее часто встречаемую начальную букву слов. Построить график расположения начальных букв слов. Вывести таблицу начальных букв слов в файл.  **Выбор алгоритма решения задачи**  Выбор алгоритма решения задачи был именно таким поскольку, я считаю что он пусть лучше подходит к условию задачи и есть последовательным и универсальным. Также есть легким для чтения и выполнение компилятором.  **Описание алгоритма решения задачи**  Программа должна считывать из текстового файла все символы (путь к файлу прописывается в самой программе без возможности изменения его пользователем), программа определяет первую букву слов и изменяет соответствующий счетчик. После определения всех начальных букв выводит график по количеству найденных букв, также в программе мается сортировка для более удобного просмотра результата. После окончания всех операций, программа записывает таблицу в файл.    **Схема алгоритма решения задачи**  **Функция scan**  http://life-prog.ru/programs/6c_clip_image001.gif    **Функция main**  http://life-prog.ru/programs/6c_clip_image003.gifhttp://life-prog.ru/programs/6c_clip_image004.gif    **Описание программы**  **Функция scan**  Эта функция открывает файл для чтения и по строкам считывает символы, после считывания каждого строке она проверяет и находит первые буквы в словах и изменяет соответствующий счетчик на 1. Для того чтобы программа смогла увеличить соответствующий счетчик используется два массива, первый заполненный всеми буквами алфавита (как малыми так и большими) и второй для хранения информации в найденных букв (т.е. счетчиков).  **Функция sort**  Для более удобного просмотра графику, используется сортировка по трем критериям, а именно: по алфавиту, по росту и по спаданию количества найденных букв.  **Функция graf**  Эта функция выводит график по количеству найденных букв, она используется после сортировки и дает возможность наглядно увидеть изменение количества букв, а для более детального просмотра слева выводит после соответствующей буквы сам счетчик (т.е. точное количество букв).    **Текст программы**  **stdafx.h**  #include <stdio.h> #include <conio.h> #include <stdlib.h> #include <iostream>  using namespace std;  const int SYMB=219;  void add(int \*sm); void scan(int \*sm, int \*kl); int maxx(int max, int \*kl); void sort(int \*sm, int \*kl); void graf(int max, int \*sm, int \*kl); void print (int \*sm, int \*kl);  **Func.cpp**  #include "stdafx.h"  void add(int \*sm) { int x; for (x=0;x<26;x++) { sm[x]=65+x; sm[x+26]=97+x; } }  void scan(int \*sm, int \*kl) { char ch[80]; int n,x; FILE \*fp; if((fp=fopen("D:\TEXT.txt", "r"))==NULL) { printf("Nevozmogno otkrit\nRazmestite fail TEXT.txt на диске D:\\"); getch(); exit(1); }         rewind(fp); while (!feof(fp)) { fgets(ch,79,fp); printf(ch); for (n=0;n<79;n++) { if (n==0 | ch[n-1]==' ') { for (x=0;x<52;x++) { if (ch[n]==sm[x]) kl[x]++; } } } for (n=0;n<79;n++) ch[n]=0; } cout<<endl; fclose(fp); }  int maxx(int max, int \*kl) { int x; for (x=0;x<52;x++) { if (kl[x]>max) max=kl[x]; } return max; }  void sort(int \*sm, int \*kl) { int i=0,x,a,b; char ch; cout<<"Sortirovka po:\n"; cout<<"1. Vozrostaniu\n"; cout<<"2. Ybuvaniu\n"; cout<<"3. Alfavity\n"; ch=getch(); switch (ch) { case '1': a=0; b=1; i=1; break; case '2': a=1; b=0; i=1; break; case '3': a=b=i=0; } while (i==1) { i=0; for (x=0;x<51;x++) { if (kl[x+a]>kl[x+b]) { int y,z; y=kl[x]; z=sm[x]; kl[x]=kl[x+1]; sm[x]=sm[x+1]; kl[x+1]=y; sm[x+1]=z; i=1; } } } }  void graf(int max, int \*sm, int \*kl) { int x,i=1; while (max>67) { max/=2; i\*=2; }         for (x=0;x<52;x++) { if (kl[x]!=0) { int a,b; printf("%2c = %4d - ",sm[x],kl[x]); for (a=-1;a<(kl[x]/i);a++) printf("%c",SYMB); cout<<endl; } } }  void print (int \*sm, int \*kl) { FILE \*fp; int x; if((fp=fopen("D:\TEXT - TABL.txt", "w"))==NULL) { printf("Nevozmogno otkrit"); getch(); exit(1); }  fprintf(fp," ------------------\n"); fprintf(fp,"|Simvol|Kolichestvo|\n"); fprintf(fp," ------------------\n"); for (x=1;x<52;x++) { if (kl[x]!=0) { fprintf(fp,"|%5c | %-9d |\n",sm[x],kl[x]); } } fprintf(fp," ------------------\n");         fclose(fp); }  **Laba4.1.cpp**  #include "stdafx.h"  void main() { int x,max=0; int sm[52]={0}, kl[52]={0}; add(sm); scan(sm,kl); max=maxx(max,kl); sort(sm,kl); cout<<"\n================================================================================\n"; graf(max, sm,kl); cout<<"\n================================================================================\n"; cout<<"Maksimal'noe kolichestvo simvolov = "; for (x=0;x<52;x++) { if (kl[x]==max) printf("%c - %d, ",sm[x],kl[x]); } print(sm,kl); getch(); }    **Результат работы программы**    http://life-prog.ru/programs/6c_clip_image008.jpg  http://life-prog.ru/programs/6c_clip_image006.jpg  http://life-prog.ru/programs/6c_clip_image010.jpg |

**ЗАДАНИЕ №6**

**Программа: Использование и обработка структурированных данных на языке С++**

|  |  |
| --- | --- |
| |  | | --- | |  |   **Задание**  Для сведения составить программу построения графику изменения зарплаты работников за  неделю.  http://life-prog.ru/programs/table.gif    **Выбор алгоритма решения задачи**  Выбор алгоритма решения задачи был именно таким поскольку, я считаю что он лучше подходит к условию задачи и есть последовательным и универсальным. Также есть легким для чтения и выполнения.  **Схема алгоритма решения задачи**  **Функция add**  **http://life-prog.ru/programs/7c_clip_image002.jpg**  **Функция viev**  **http://life-prog.ru/programs/7c_clip_image004.jpg**  **Функция graf**  **http://life-prog.ru/programs/7c_clip_image006.jpg**  **Функция main**  **http://life-prog.ru/programs/7c_clip_image008.jpg**    **Текст программы**    **Текст Laba5.cpp**  #Include "stdafx.h"  Void main(void) { int n=0;    //переменная для количества рабочих cout<<"vvedite kolichestvo robochih - "; cin>>n;     //здесь мы вводим количество рабочих cout<<endl; add(n);//функция заполнения структуры  cout<<"press any key to continue...\n"; getch(); view(n); //функция вывода таблицы на экран      cout<<"press any key to continue...\n"; getch(); graf(n);// функция рисования графику  getch(); }    **Текст stdafx.cpp**  #include "stdafx.h"  void add(int n)    //функция ввода данных { int x,y,z;   //переменные для работы программы for (x=0;x<n;x++) { A[x].no=x+1; cout<<"Vvedite fio "<<x+1<<"-go robochego - "; cin>>A[x].fio; // Заполнение имени рабочего  cout<<"Vvedite professiu "<<x+1<<"-go robochego - "; cin>>A[x].pro; // Заполнение профессии рабочего  z=0; for (y=0;y<5;y++) { z++; cout<<"Vvedite kolichestvo chasov dlia "<<x+1 <<"-go robochego za "<<z<<"-iu den - "; cin>>A[x].B[y].ch;   //ввод количества часов A[x].B[y].zar=A[x].B[y].ch\*56; } } } ////////////////////////////////////////////////////////////////////////// void view(int n)// Функция вывода таблицы на экран  { if(n>1) { int x,y; printf("-------------------------------------------------------------\n"); printf("|   |      |          | Mo|   Tu  |   We  |   Th  |   Fr    |\n"); printf("| # | FIO  |Profession| Ch|$  | Ch| $ | Ch| $ | Ch|$ | Ch| $ |\n"); printf("|---|---------|--------|------|------|------|------|---------|\n"); for(x=0;x<n;x++)  //цикл вывода рабочих  { printf("|%2d | %9s | %8s |",A[x].no,A[x].fio,A[x].pro); for (y=0;y<5;y++) printf("%2d |%4d |",A[x].B[y].ch,A[x].B[y].zar); cout<<endl; } printf(" -----------------------------------------------------------\n\n"); } else cout<<"Ne zadano kolichestvo rabochih\n"; } ////////////////////////////////////////////////////////////////////////// void graf(int n)// Функция построения графику  { int x,y,a,b,p=1,max=0,s[][1]={0,0}; if(n>1) { for (x=0;x<n;x++) { s[x][0]=0; for (y=0;y<5;y++)// Нахождение суммы зарплаты за неделю  s[x][0]=s[x][0]+A[x].B[y].zar; } for (x=0;x<n;x++) { if (max<s[x][0])// Нахождение наибольшей зарплаты за неделю  max=s[x][0]; } while (max>20)// Нахождение делителя для получения числа меньше 20  { max=max/2; p=p\*2; } for (x=0;x<n;x++) s[x][0]=s[x][0]/p;// Пропорциональное деление зарплаты всех работников  b=max; for (a=0;a<b;a++) { cout<<"| "; for (x=0;x<n;x++) { if (s[x][0]<max)//если зарплата работника меньше максимальной cout<<"       ";// Вывод если меньше от максимальной                          else cout<<"\*\*\*\*\*\* ";// Вывод если больше от максимальной  } cout<<endl; max--;// Уменьшение максимального значения  } cout<<" --------------------------------------------------------->\n   "; for (x=1;x<n+1;x++) cout<<x<<"-iu   "; cout<<" - Rabotniki"; } else cout<<"Ne zadano kolichestvo rabochih\n"; }    **Текст stdafx.h**  #pragma once  #define WIN32\_LEAN\_AND\_MEAN        // Exclude rarely-used stuff from Windows headers #include <stdio.h>// Подключение библиотек  #include <tchar.h> #include <iostream> #include <conio.h>  using namespace std;  struct    // Объявление структуры как без измеримого массива  { int no; char fio[9],pro[8]; struct { int ch,zar; }B[5]; }A[]={0}; // Прототипы функций  void add(int n); void view(int n); void graf(int n);    **Результаты работы программы**  **http://life-prog.ru/programs/7c_clip_image010.jpg**  **http://life-prog.ru/programs/7c_clip_image012.jpg**  **http://life-prog.ru/programs/7c_clip_image014.jpg** |

**ЗАДАНИЕ №7**

# Программа: Использование и обработка динамических списков на языке С++

|  |  |
| --- | --- |
| |  | | --- | |  |   **Задание**  Для сведения составить программу построения графику изменения зарплаты работников по каждому дню.  http://life-prog.ru/programs/table.gif  **Выбор алгоритма решения задачи**  Выбор алгоритма решения задачи был аналогичен предыдущей программе -> [см. здесь](http://life-prog.ru/view_programs.php?id=31&language=c&page=1).  **Схема алгоритма решения задачи**  **Основная программа**  **http://life-prog.ru/programs/8c_clip_image002.jpg**  **Функция del**  **http://life-prog.ru/programs/8c_clip_image004.jpg**  **Функция SRAND**  **http://life-prog.ru/programs/8c_clip_image006.jpg**  **Функция name**  **http://life-prog.ru/programs/8c_clip_image008.jpg**  **Функция add**  **http://life-prog.ru/programs/8c_clip_image010.jpg**  **Функции view и graf см.**[здесь ->>](http://life-prog.ru/view_programs.php?id=31&language=c&page=1)    **Описание программы**  **Функция add**  Эта функция заполняет массив структур (размер массива отмечается в константе), заполнение происходит автоматически с помощью произвольных чисел и имен, пользователь не берет ни которой участия в заполнении.  **Функции view и graf см.**[здесь ->>](http://life-prog.ru/view_programs.php?id=31&language=c&page=1)  **Функция del**  Эта функция удаляет весь список с помощью поочередного удаления каждого элемента.    **Текст программы**  **stdafx.h**  #include <stdio.h> #include <iostream> #include <conio.h>  using namespace std;  const int ch=8;  struct A                 //структура рабочего { A \*prev;                               //указатель на предыдущий int no; char fio[9]; char pro[9]; struct B { int ch,zar; }B[5]; A \*next;                //указатель на следующий }; //прототипы функций int SRAND(); int name(int y); A \*add(int n, A \*first); void view(int n, A \*first); void graf(int n, A \*first); void del(A \*first);  **Func.cpp**  #include "stdafx.h" //подключение файла stdafx.h  int SRAND()         // функция случайное число в диапазоне от 0 .. 1000 { int i=0; cout<<"Press any key to start..."; while (!kbhit())     //если клавиша не нажата { if (i>1000) i=0; i++; } cout<<endl; return(i);    //возвращение полученного числа }  int name(int y)     //функция получение буквы { int x,m[26]; for (x=0;x<26;x++) m[x]=x+65; return (m[y]); }  A \*add(int n, A \*first)       //функция добавление списка и заполнение случайным числом. { A \*p, \*temp=0; int x,y,z,s,numb; char nam[9]; srand(SRAND());   //функция для хаотичного случайного числа p=new A;           //выделение памяти под список for (x=0;x<n;x++)   //цикл добавления { if (x==0)    //если список первый first=p;    //указатель на первый элемент else p->prev->next=p;                                 p->prev=temp;  //указатель на предыдущий temp=p; p->no=x+1; for (s=0;s<8;s++)     { p->fio[s]=name(rand()%26); //случайное заполнение фамилии из 8 букв p->pro[s]=name(rand()%26); //случайное заполнение профессии из 8 букв } p->fio[s]='\0';  //присвоение конца строки p->pro[s]='\0';  //присвоение конца строки for (y=0;y<5;y++) {                                              p->B[y].ch=rand()%(ch+1); p->B[y].zar=p->B[y].ch\*56; } if(x!=n-1)   //если список не последний { p->next=new A; p=p->next; } else p->next=0;  //иначе он последний.  Указатель на следующий принимает значение 0 } return first;    //возврат указателя на первый элемент }  void view(int n, A \*first)       //вывод таблицы работников { A \*p; if(n>0) { int x,y; printf(" ----------------------------------------------------------------------------\n"); printf("|   |           |          |   Mo    |   Tu    |   We    |   Th    |   Fr    |\n"); printf("| # |     FIO   |Profession| Ch|  $  | Ch|  $  | Ch|  $  | Ch|  $  | Ch|  $  |\n"); printf("|---|-----------|----------|---------|---------|---------|---------|---------|\n"); for(x=0;x<n;x++) { p=first; printf("|%2d | %9s | %8s |",p->no,p->fio,p->pro); for (y=0;y<5;y++) printf("%2d |%4d |",p->B[y].ch,p->B[y].zar); cout<<endl; first=p->next; } printf(" ----------------------------------------------------------------------------\n\n"); } else cout<<"Ne zadano kolichestvo rabochih\n"; }  void graf(int n, A \*first)                    //функция построения графика { A \*p, \*temp; int x,y,a,b,k=1,max=0,s[5]; if(n>0) { for (y=0;y<5;y++) { temp=first; s[y]=0; for (x=0;x<n;x++) { p=temp; s[y]=s[y]+p->B[y].zar;   // Нахождение суммы зарплаты за неделю  temp=p->next; } } for (y=0;y<5;y++) { if (max<s[y])      // Нахождение наибольшей зарплаты за неделю  max=s[y]; } while (max>20)    // Нахождение делителя для получения числа меньше 20  { max=max/2; k=k\*2; } for (y=0;y<5;y++) s[y]=s[y]/k;   // Пропорциональное деление зарплаты всех работников  b=max; for (a=0;a<b;a++)         { printf("%3d| ",a+1); for (y=0;y<5;y++) { if (s[y]<max)    //если зарплата работника меньше максимальной  cout<<"              ";  // Вывод если меньше от максимальной  else cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\* ";  // Вывод если больше от максимальной  } cout<<endl; max--;   // Уменьшение максимального значения  } cout<<" ----------------------------------------------------------------------------->\n   "; for (y=1;y<6;y++) { cout<<"   "<<y<<"-iu"; cout<<" - Den'"; } } else cout<<"Ne zadano kolichestvo rabochih\n"; }  void del(A \*first)    //функция удаления списка { A \*p, \*temp; for(p=first, temp=p->next; temp; p=temp, temp=p->next)   //пока не конец списка  delete p;                                                                                        //удаление текущего списка }    **Laba5.cpp**  #include "stdafx.h"                            //подключение файла stdafx.h  void main(void) { A \*first=0; int n; cout<<"Vvedite kolichestvo robochih "; cin>>n; first=add(n, first);   //функция заполнения структуры  getch(); view(n, first);         //функция вывода таблицы на экран      cout<<"Press any key to continue...\n\n"; getch(); graf(n, first);          // функция рисования графику  del(first);              // функция удаления списка  getch(); }    **Результат работы программы**  **http://life-prog.ru/programs/8c_clip_image012.jpg** |

**ЗАДАНИЕ №8**

# Готовая программа: Классы. Инкапсуляция на языке Си

|  |  |
| --- | --- |
| |  | | --- | |  |   **Задание**  Для сведения составить программу построения графику изменения зарплаты работников по каждому дню.  http://life-prog.ru/programs/table.gif  **Выбор алгоритма решения задачи**  Для решения этой задачи надо использовать класс в каком все данные нужно сделать закрытыми (для того чтобы при работе случайно не повредить их) и методы обработки этих данных. Уместным я считаю сделать методы: добавление, удаление одного или всех элементов, вывод графику и списка на экран, а также несколько универсальных функций которые помогут в работе как: функция поиска, заполнение, и индексирование элементов.  **Описание алгоритма решения задачи**  Основа программы это функция main, поскольку в ней находится меню пользователя из которого можно задействовать любую из функций. Меню всегда находится на экране как в начале работы так и после окончания работы одной из функций, завершение программы происходит при нажатии клавиши Esc.  Удаление всего списка происходит благодаря поочередному переходу с начала до конца списка с одновременным удалением активного элемента.  Добавление/удаление отдельного элемента происходит в два этапа, после того как пользователь выберет место добавления/удаление элемента, функция поиска возвратит показатель на тот элемент который забав пользователь, а потом функция добавления/удаление разорвет цепь и прибавит/удалит элемент. Функция вывода таблицы/графику поочередно перебирает все элементы списка и выводит список/график (лишь производительность в виде шкалы) на экран.    **Схему алгоритма**[см. здесь](http://life-prog.ru/view_programs.php?id=32&language=c&page=1)    **Описание программы**  1 **Функция inf**: После добавления нового элемента в список эта функция заполняет все его поля произвольными значениями давая лишь возможность пользователю самостоятельно выбрать как будет заполняться: из клавиатуры или произвольным значением.  2 **Функция index**: Эта функция индексирует все элементы списка в порядке роста.  3 **Функция find**: Эта функция перед добавлением нового элемента в список, ищет тот элемент на место которого должен прибавиться новый. И возвращает показатель на него, в случае если заданная позиция большая чем мается в списке элементов, то новой элемент добавится в конец списка.  4 **Функция add**: Прибавляет новой элемент в список учитывающие те данные которые возвратит функция find.  5 **Функция view**: Выводит таблицу на экран монитора в которой отображаются все элементы списка.  6 **Функция graf**: Выводит на экран график производительности работы всех рабочих в списке по каждому дню за неделю.  7 **Функция del**: Удаляет тот  элемент списка который укажет пользователь, функция удаляет элемент в соответствии с тем какое значение возвратит функция find.  8 **Функция del\_all**: Удаляет весь список и возвращает показателю на первый элемент нулевое значение.    **Текст программы**  **Head.h**  #ifndef \_HEAD\_H\_ #define \_HEAD\_H\_ #include <stdio.h> #include <iostream> #include <conio.h> #include <time.h>  using namespace std;  const int hr=8;  class A      //объявление класса А { private: //закрытые данные A \*prev; int no; char fio[9]; char pro[9]; struct B { int ch,zar; }B[5]; A \*next; public:  // открытые данные //прототипы функций для работы со списком void inf(A \*p);               //функция заполнения полей данными  void index(A \*first);        // функция индексирует все элементы списка в порядке роста  A \*find(A \*first, int n);    // функция ищет тот элемент, на место которого должен  //прибавиться новый  A \*add(A \*first,A \*fin);     // Прибавляет новой элемент в список учитывающие те  //данные которые  // возвратит функция find  void view(A \*first);         //Функция выводит данные таблицы на экран void graf(A \*first);         // Выводит на экран график производительности работы  //всех рабочих  A \*del(A \*first,A \*fin);     //удаление одного элемента A \*del\_all(A \*first);        //удаление всех элементов };  #endif    **Func.cpp**  #include "Head.h" //подключение файла Head.h  int key, count=0;  void A::inf(A \*p)       //реализация функции заполнения { srand(unsigned (time(NULL)));     //инициализация хаотичного случайного набора int y,s,flag=0; char ch; p->no=++count; cout<<"1. Vvesti FIO\n"; cout<<"2. Sgenerirovat' FIO\n\n"; ch=getch();      switch(ch)   //выбор действий { case '1':    //если нажата цифра 1 cout<<"Vvedite FIO "; cin>>p->fio; cout<<endl; break; case '2':    //если нажата цифра 2 flag=1; break; } for (s=0;s<8;s++)    { if (flag==1) //если нажата была цифра 2 p->fio[s]=char((rand()%26)+64);         //сгенерировать случайную ФИО p->pro[s]=char((rand()%26)+64);         //генерирования случайной профессии } p->fio[s]='\0';  //присвоение конца строки p->pro[s]='\0';  //присвоение конца строки      for (y=0;y<5;y++) {                    p->B[y].ch=rand()%(hr+1); p->B[y].zar=p->B[y].ch\*56; } }  //////////////////////////////////////////////////////////////////////////// void A::index(A \*first)     //реализации функции индексации { A \*p; int flag=1; count=0; while (flag) //пока не достигнут конец списка { p=first; p->no=++count; if (p->next==0) flag=0; else first=p->next; } }  //////////////////////////////////////////////////////////////////////////// A \* A::find(A \*first, int n)   //реализации функции позиции на добавление { int i=0; A \*p; if (n>0)     //если позиция больше нуля { if (first==0) //если список еще не создан { key=1; return first; } if (n==1)    //если введена первая позиция { key=1; return (first); } else         //иначе введено число больше чем 1 { while (i<n-1)       //пока не достигнут конец списка { p=first; if (p->next==0)//если уже конец списка   {                                 i=n; key=0; return (p);//возврат указателя на посдедний элемент списка } else   //если не конец списка first=p->next; i++;   //инктемент i } key=1; return (first);      } } else   //иначе введено n<1 список не может быть создан { key=-1; cout<<"Zadanoe chislo ne vhodit v predel spiska...\a\n"; } }  //////////////////////////////////////////////////////////////////////////// A \* A::add(A \*first,A \*fin)     //реализация функции прибавления нового элемента в список { A \*p; if (key>=0)  // если key>=0 тогда можна добавить список  { p=new A; if (fin==0)  //если списка еще нет { p->prev=0; first=p; p->next=0; p->inf(p); } else { if (key==0)  //если список достигнут конца { p->prev=fin; p->next=0; fin->next=p; p->inf(p); } else         //иначе позиция не достигла конца { if (fin==first!=0)  //если fin==first!=0 то это первый список { p->prev=0; p->next=first; first->prev=p; first=p; p->inf(p); } else { if (fin!=first)   //это не первый список. Добавлять нужно в середине {                   //списка p->prev=fin->prev; p->next=fin; fin->prev->next=p; fin->prev=p; p->inf(p); } } } } } return first; }  //////////////////////////////////////////////////////////////////////////// void A::view(A \*first)     //реализация функции вывода на экран { A \*p; int flag=1; if(first!=0) { int y; printf(" ----------------------------------------------------------------------------\n"); printf("|   |           |          |   Mo    |   Tu    |   We    |   Th    |   Fr    |\n"); printf("| # |     FIO   |Profession| Ch|  $  | Ch|  $  | Ch|  $  | Ch|  $  | Ch|  $  |\n"); printf("|---|-----------|----------|---------|---------|---------|---------|---------|\n"); p=first; while (flag!=0) { p=first; printf("|%2d | %9s | %8s |",p->no,p->fio,p->pro); for (y=0;y<5;y++) printf("%2d |%4d |",p->B[y].ch,p->B[y].zar); cout<<endl; first=p->next; if (p->next==0) flag=0; } printf(" ----------------------------------------------------------------------------\n\n"); } else cout<<"\nNe zadano kolichestvo rabochih\n"; }  //////////////////////////////////////////////////////////////////////////// void A::graf(A \*first)      //реализация функции построения графика { A \*p, \*temp; int y,a,b,k=1,max=0,s[5]; if(first!=0) { for (y=0;y<5;y++) { int flag=1; temp=first; s[y]=0; while (flag!=0) { p=temp; s[y]=s[y]+p->B[y].zar; temp=p->next; if (p->next==0) flag=0; } } for (y=0;y<5;y++) { if (max<s[y]) max=s[y]; } while (max>20) { max=max/2; k=k\*2; } for (y=0;y<5;y++) s[y]=s[y]/k; b=max; for (a=0;a<b;a++) { printf("%3d| ",a+1); for (y=0;y<5;y++) { if (s[y]<max) cout<<"              "; else cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\* "; } cout<<endl; max--; } cout<<" ----------------------------------------------------------------------------->\n   "; for (y=1;y<6;y++) { cout<<"   "<<y<<"-iu"; cout<<" - Den'"; } cout<<endl; } else cout<<"Ne zadano kolichestvo rabochih\n"; }  //////////////////////////////////////////////////////////////////////////// A \*A::del(A \*first,A \*fin)    //функция удаления одного элемента { A \*p, \*temp; if (key>=0) { if (fin==0)    //если список пуст, то нечего удалять { cout<<"Spisok pyst!\n"; } else { if (key==0)  //если в списке нет такого элемента { //if () cout<<"Takogo elementa nety v spiske!"; } else { if (fin==first!=0)//если это первый элемент { if (first->next==0)  //и он последний { p=first; delete p; first=0; return first; } else                 //если он не последний { p=first; first=p->next; first->prev=0; delete p; return first; } } else                //если это не первый элемент { if (fin!=first)     { p=fin; if (p->next==0)//если он последний { p->prev->next=0; delete p; return first; } else         //иначе он не последний { fin->prev->next=p->next; p->next->prev=fin->prev; delete p; return first; } } } } } } return first; }  //////////////////////////////////////////////////////////////////////////// A \* A::del\_all(A \*first)      //реализация удаления всего списка { A \*p; int flag=1; if (first!=0) { if (flag!=0) { p=first; if (p->next==0) flag=0; else first=p->next; delete p; } } first=0; return first; }    **Main.cpp**  #include "Head.h"  void main(void) { A a, \*first=0, \*p; int n,x,end=1; char ch; while (end!=0)             //пока не нажата клавиша ESC { system("cls"); cout<<"===========Menu===========\n"; cout<<"  1. Dobavit'\n"; cout<<"  2. Tablica\n"; cout<<"  3. Grafik\n"; cout<<"  4. Indexirovanie\n"; cout<<"  5. Ydalit' element\n"; cout<<"  6. Ydalit' spisok\n"; cout<<"Esc. Exit\n"; cout<<"==========================\n"; ch=getch(); switch (ch)   // выбор от нажатия { case '1':     //если нажата клавиша 1, то добавление элемента system("cls"); cout<<"\nVvedite poziciy elementa "; cin>>n; first=first->add(first,first->find(first,n)); break; case '2':    //если нажата клавиша 2, то вывод на экран списка system("cls"); first->view(first); getch(); break; case '3': system("cls"); first->graf(first); getch(); break; case '4': system("cls"); first->index(first); first->view(first); getch(); break; case '5': system("cls"); cout<<"\nVvedite ydaliaemeu element "; cin>>n; first=first->del(first,first->find(first,n)); break; case '6': system("cls"); first=first->del\_all(first); break; case 27:   //если нажата клавиша ESC, значит выход end=0; break; } } first->del\_all(first);    //удаление всего списка }    **Результат работы программы**  **http://life-prog.ru/programs/9c_clip_image002.jpg**  **http://life-prog.ru/programs/9c_clip_image004.jpg**  **http://life-prog.ru/programs/9c_clip_image006.jpg** |

**ЗАДАНИЕ №9**

Составить программу, которая сортирует таблицу студентов по максимальном или минимальном значении средней оценки. Тип сортировки задается во время программы (по максимальному или по минимальному).

**Выбор алгоритма решения задачи**

Для решения данной задачи необходимо сделать структуру, в котором есть имя, группа и средняя оценка студента. Необходимы будут функции вводу, обработки и вывода сортировочной таблицы.

**Описание решения задачи**

Объявляем три функции:

Void vvod(student );   
Void sort(student );   
Void vivod(student );

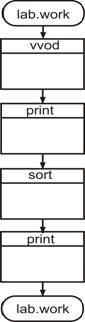
Функция ввода является стандартной, то есть через cin>> мы вносим значение в программу.

В функции sort мы объявляем переменную типу short или просто int, в которую во время работы программы будем заносить +/- 1. Если мы внесем 1 тогда программа будет сортировать по большему среднему баллу, а если - 1, тогда наоборот.

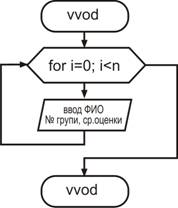
Потом делаем функцию, которая выводит студентов в заполняемом и в сортированном виду.

**Схема алгоритма решения задачи**

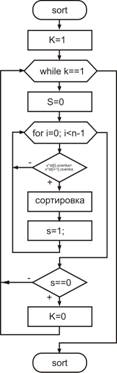
**Основная программа**

****

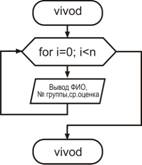
**Фукнция ввода**

****

**Фукнция сортировки**

****

***Ф*укнция вывода на экран таблици**

****

**Текст программы**

**1 Header.h**

#ifndef TEXT\_H  
#define TEXT\_H

#include<iostream.h>  
#include<conio.h>  
#include<string.h>

const n = 2;

struct student {  
char name[20];  
int group;  
float ocenka;  
};

void vvod(student \*);  
void sort(student \*);  
void print(student \*);

#endif

**2 Program.cpp**

#include "header.h"

int main(){

            student stud[n];

            vvod(&stud[n]);  
sort(&stud[n]);  
print(&stud[n]);  
  
getch();  
return 0;  
}

**3 Function.cpp**

#include "header.h"

void vvod(student \*st){

      cout <<"Vvedite name, group, srednuyu ocenky: ";  
for (int i=0; i<n; i++)   
cin>>st[i].name>>st[i].group>>st[i].ocenka;  
}

void sort(student \*st){  
short x, k;  
char name1[20];  
int group1, s;  
float ocenka1;  
cout <<"Vvedyte po vozrastaniu ili po ubivaniy 1, -1 ? ";  
cin >> x;

      k=1;  
while (k==1) {  
s = 0;  
for (int i=0; i<n-1; i++)  
if (x\*st[i].ocenka < x\*st[i+1].ocenka) {  
s = 1;  
strcpy (name1 , st[i].name);   
group1 = st[i].group;  
ocenka1 = st[i].ocenka;  
st[i] = st[i+1];  
strcpy (st[i+1].name, name1);  
st[i+1].group = group1;  
st[i+1].ocenka = ocenka1;  
}  
if (s == 0) k = 0;  
}  
}

void print(student \*st){

      cout <<"------------------------------------";  
for (int i=0; i<n; i++)   
cout <<"|"<< st[i].name <<" | "<< st[i].group <<" | "<< st[i].ocenka<<" |\n";  
cout <<"------------------------------------";

ЗАДАНИЕ №10

**Программа: Управление динамическим распределением памятью в языке Си**

|  |  |
| --- | --- |
| |  | | --- | |  |   **Задание**  Разработать программу, которая выполняет обработку данных об объектах файловой системы (их атрибуты: имя, расширение, дата создания, размер, атрибуты). Даны хранить в виде однонаправленного списка в динамически распределяемой памяти. Программа должна поддерживать диалоговый интерфейс с пользователем и обеспечивать следующие:  1. добавление объектов в список;  2. удаление объектов из списка;  3. поиск и отображение объектов по заданному признаку;  4. отображение информационных полей всех элементов списка.  Выполнить контроль и мониторинг использования динамически распределяемой памяти. Для общения с пользователем использовать простое меню или общаться с помощью команд.    **Выбор алгоритма решения задачи**  Для решения данной задачи, необходимо использовать динамическую память. Разбить программу на функции (функции создания, удаления записи и т. д.).  В данном алгоритме используются связанные списки. В них существует шесть полей с показателями, один из которых указывает на следующий элемент списка, а другие на данные которые содержат нужную информацию. При выделении памяти сначала выделяется память для записи, а потом для структуры данных. При удалении все происходит в обратном порядке, поскольку иначе не будет освобождаться объем задействованной. При выделении памяти сначала выделяется память для записи, а потом для структуры данных. При удалении все происходит в обратном порядке, поскольку иначе не будет освобождаться объем задействованной памяти.    **Описание решения задачи**  Функция Add создает элемент списка, память в куче выделяется с помощью функции new. Функция Del удаляет выбранный элемент списка по его номеру (номер элемента на отображается на экране). Память освобождается функцией delete, сначала освобождается данные используются структурой данных, а потом память тронута самой записью. Функция Show выводит на экран созданный список. Функция Find находит искомый элемент (поиск возможен по части имени). Функция Freemem осуществляет очистку памяти занятой списком.  В программе пользователю предоставляется выбор конкретных действий после выполнения этих действий осуществляется возвращение в главное меню. Выход из программы осуществляется при нажатии клавиши Esc, после этого осуществляется очистка памяти и проверка, освободилась ли вся память. Если нет, то выводится предупреждение.    **Текст программы**  **1. Head.h**  #ifndef TEXT\_H  #define TEXT\_H  #include <iostream.h> #include <string.h> #include <alloc.h> #include <conio.h> #include <stdlib.h>    struct FDat {               //структура данных char Name[20];          //имя char r[4];              // расширение char d[10];             //       дата  int size;               // размер char attr[4];           //  атрибут FDat \*Next; };  FDat \*Add(FDat \*, FDat \*); FDat \*Del(FDat \*, int &); void Show (FDat \*); void Find (FDat \*); void FreeRam (FDat \*);  #endif    **2 PROGRAM**  #include "head.h"  FDat \*Add(FDat \*posl) { if (posl == NULL){ posl = new FDat; } else { posl->Next = new FDat; posl = posl->Next; }                  cout << "Vvedite - Name, Racsherenie, Daty, Razmer and Attribut: \n"; cin >> posl->Name >> posl->r >> posl->d >> posl->size >> posl->attr; posl->Next = NULL; return posl; }  FDat \*Del(FDat \*begin, int &identif              ) { FDat \*prom, \*begin\_new, \*next; int x = 0, j =  0, k = 0, y = 0; char name[20];                  cout << "Mi udalaem!!! \n";                  cout <<"Vvedite NAME : "; cin >> name; begin\_new = prom = next = begin; while ( begin != NULL) { for (int i = 0; i<strlen(name); i++) if (name[i] == begin->Name[i]) j++; if (j == strlen(name)) { if (prom == begin) { begin\_new = begin->Next; prom = begin->Next; delete begin; x = 1; y = 1; begin = prom; } else { while ( k == 0) { if (next->Next == begin){ if (next->Next->Next == NULL) { k = 1; x = 1; delete begin; next->Next = NULL; begin = NULL; identif = 1; begin\_new = next; } else { prom = begin->Next; delete begin; x = 1; k = 1; y = 1; next->Next = begin = prom; } } next = next->Next; } } } if (y == 1) begin = NULL; if (begin != NULL)            begin = begin->Next; j = 0; } if (x == 0) cout << "NAME nety ... "; getch(); return begin\_new;  }  void Show (FDat \*begin) { cout << "Prosmotr spiska!!! \n"; cout <<"Name, Racsherenie, Data, Razmer and Attribut\n"; while ( begin != NULL) { cout <<"\n"<< begin->Name <<"."<<begin->r <<" "<<begin->d <<"  " << begin->size <<" byte"<<begin->attr <<" \n "; begin = begin->Next; } getch();  }  void Find (FDat \*begin) { int x = 0, j =  0; char name[20]; cout << "poisk!!! \n"; cout <<"Vvedite NAME : "; cin >> name; while ( begin != NULL) { for (int i = 0; i<strlen(name); i++) if (name[i] == begin->Name[i]) j++;                                   if (j == strlen(name)) { cout <<"\n"<< begin->Name <<"."<<begin->r <<" "<<begin->d <<"  " << begin->size <<" byte"<<begin->attr <<" \n "; x = 1; j = 0; }                                 begin = begin->Next; } if (x == 0) cout << "NAME nety ... "; getch(); }  void FreeRam (FDat \*begin) { FDat \*prom; cout << "ochistka!!! \n"; prom = begin; while ( prom != NULL) { begin = begin->Next; delete prom; prom = begin; } }  ////////////////////////////////////////////////////  int main () { long int mem; FDat \*Begin = NULL, \*Posl = NULL, \*Dopol; int ch, iden = 0;               //    {наж. клавиша}                  mem = farcoreleft(); //     {Размер своб. памяти} cout <<"Memory - "<<mem<<"\n"; do { cout<<" <1> - DOBAVIT ZAPIS \n" <<" <2> - UDALIT ZAPIS \n" <<" <3> - PROSMOTR SPISKA \n" <<" <4> - POISK \n" <<" <5> - EXIT \n";                    cin>>ch;       //  {реализация меню}                    switch (ch) { case 1 : cout <<"adress = " <<Begin; cout <<"adress = " <<Posl; if (Begin == NULL) { Begin = Add(Posl); Posl = Begin; } else Posl = Add( Posl); cout <<"adress = " <<Begin; break; case 2 : Dopol = Del(Begin, iden); if (iden == 0) Begin = Dopol; else Posl = Dopol; break; case 3 : Show(Begin); break; case 4 : Find(Begin); break; } } while (ch != 5);                  cout <<"\n Memory - " <<farcoreleft()<<"\n";                  FreeRam(Begin);           //   {очистка памяти} getch();                  cout <<"\n Memory - " <<farcoreleft()<<"\n"; if (mem == farcoreleft()) cout <<"Pamat' ne poterana ... "; else cout << "ERROR!!! Potera Memory!!!";                  cout<<" Press key..."<<endl; getch();                  return 0; }    **Результат роботи програми**  http://life-prog.ru/programs/12c_clip_image002.jpg  http://life-prog.ru/programs/12c_clip_image004.jpg  http://life-prog.ru/programs/12c_clip_image006.jpg  http://life-prog.ru/programs/12c_clip_image008.jpg |

ЗАДАНИЕ №11

**Перегрузка операций в Си. Абстракция данных на языке C++**

|  |  |
| --- | --- |
| |  | | --- | |  |   **Задание**  На основе абстрактного базового класса, реализовать контейнер динамических объектов, выполнить перегрузки основных операций для металлической сетки, таким образом, чтобы они вели себя подобно встроенных стандартных типов. Основные операции (=, отношение, арифметические операции).  **Выбор алгоритма решения задачи**  Для решения этой задачи необходимо создать абстрактный базовый класс. И производный от него класс управления данными, не обязательно динамических объектов, так как алгоритм и структура одинакова. Перегруженные операторы, такие как отношение, арифметические и так далее, необходимо указать, а затем описать, что они будут делать.  **Описание решения задачи**  Для этого выберем абстрактный базовый класс: class Employee. В нем опишем перегрузки операторов, например «+»: friend Employee &amp; operator + (Employee &amp;, Employee &amp;. В походном классе опишем действие перегруженного оператора «+».  friend complex  operator+ (complex &i, complex &\_i)          {complex R(0,0);                    R.re = i.re + \_i.re; R.im = i.im + \_i.im;                    return R;          }    **Текст программы**  **1. Base.h**  #ifndef BASE\_H #include <iostream.h> #include <string.h> #include <assert.h> #define BASE\_H  class Employee { public: Employee (void) {};              friend   Employee & operator+ (Employee &, Employee &) ; friend bool operator== (Employee &,Employee &); friend Employee & operator- (Employee &, Employee &) ; friend ostream & operator<< (ostream & o, Employee &) ; friend istream & operator>> (istream & o, Employee &) ; };  #endif    **2. Cint.h**  #include <iostream.h> #include "base.h"  class complex : public Employee { int re; int im; public: complex (int r, int j) { re = r; im = j; }              friend ostream  &operator<< (ostream & o, complex &i) { return o <<  i.re << "  " << i.im<<endl; }              friend istream &operator>> (istream & o, complex &i) {return o >> i.re >>i.im;}              friend complex  operator+ (complex &i, complex &\_i) {complex R(0,0); R.re = i.re + \_i.re; R.im = i.im + \_i.im; return R; }              friend bool operator== (complex &i,complex &\_i) { if (i.re == \_i.re && i.im == \_i.im)  return true; else return false; }              friend complex  operator- (complex &i, complex &\_i) { complex R1(0,0); R1.re = i.re - \_i.re; R1.im = i.im - \_i.im; return R1; } };    **3. Main.cpp**  #include <iostream.h> #include <iomanip.h> #include <conio.h> #include "BASE.h" #include "Cint.h"  int main() { complex A1(0,0); complex A2(0,0);  cout << "Vvedite - 2 raza 2 chisla: \n";  cin >> A1; cin >> A2;  cout <<"Perviy masiv А1= " << A1<<'\n'; cout <<"Vtoroy masiv А2= " << A2<<'\n';              cout << "a1 + a2: " << A1 + A2 << endl; cout << "a1 - a2: " << A1 - A2 << endl;              A1 = A2; cout <<"A1 = А2: "<< A1 <<endl;  if (A1 == A2) cout << "A1 == A2 "<<endl;              cout<< A1 << endl;              getch();              return 0; }    **Результат работы программы**  http://life-prog.ru/programs/14c_clip_image002.jpg |

ЗАДАНИЕ №12

# Разработка собственных типов в С. Полиморфизм в языке Си

|  |  |
| --- | --- |
| |  | | --- | |  |   **Задание**  Разработать собственный класс управления строками. Объекты которого использовать как элементы ранее разработанного контейнером (см. [здесь](http://life-prog.ru/view_programs.php?id=39&language=c&page=2)). Расширить функциональность контейнерного класса, обеспечив обработку данных, хранящихся, например: поиск. Продемонстрировать возможность однотипной обработки разнотипных элементов контейнера (Объектов класса строки и каких-либо других).  **Примечания**: предусмотреть возможность ввода и вывода данных с использованием стандартных cin, cout.    **Выбор алгоритма решения задачи**  Для реализации данной задачи необходимо разработать собственный класс управления строками.Т.е. перегружать операторов += для присвоения к существующему строки, операция == для сравнения, операция < или > и т.д. Я использовал ранее разработанный абстрактный базовый класс. И объявил производный от него класс в котором описал действия управления строками.    **Описание решения задачи**  Для решения этой задачи необходимо объявить абстрактный базовый класс class Employee. В нем я перегружает основные перегружены функции - это, например <<: friend ostream & operator<< (ostream & про, Employee &). После этого объявляется производный от абстрактного - это class String.Записывается он так: class String: public Employee. Так же в этом классе реализован поиск в строке.    **Схема алгоритма решения задачи**  http://life-prog.ru/programs/15c_clip_image0022.jpg    **Текст программы**  **1. Base.h**  #ifndef BASE\_H #include <iostream.h> #include <string.h> #include <assert.h> #define BASE\_H  class Employee { public: Employee (void) {};           friend ostream & operator<< (ostream & o, Employee &) ; friend istream & operator>> (istream & o, Employee &) ; };  #endif    **2. Cint.h**  #ifndef CINT\_H #define CINT\_H  #include <iostream.h> #include <string.h> #include "base.h"  class String : public Employee {  private: int length; char \*sPtr;              void setString (const char \*);  public: String (const char \* = ""); String (const String &); ~String(void);              void Poisk ();              const String &operator= (const String &i) {          if ( &i != this) { delete [] sPtr; length = i.length; setString( i.sPtr ); } return \*this; }              bool operator ==(const String &i) const {return            strcmp (sPtr, i.sPtr) == 0;}              const String &operator+= (const String &);              friend ostream &operator<< (ostream &o, const String &s) { o <<s.sPtr;   return o;} friend istream &operator>> (istream &i, String &s) {          char temp[50]; i >> setw(50)>>temp; s = temp; return i; }              bool operator!= ( const String &right ) const   { return !( \*this == right ); } // проверка s1 > s2                                                               bool operator<( const String &i ) const      { return strcmp (sPtr, i.sPtr) < 0;}  bool operator> ( const String &i ) const      { return strcmp (sPtr, i.sPtr) > 0;}              virtual bool operator<=( const String &right ) const  { return !( right < \*this ) ; }              bool operator>=( const String &right)  { return !( \*this < right ); } }; const String &String::operator +=( const String &right) {  char \*tempPtr = sPtr; length += right.length; sPtr = new char [length +1]; assert ( sPtr != 0); strcpy ( sPtr, tempPtr); strcat ( sPtr, right.sPtr); delete [] tempPtr; return \*this; }  void String::setString(const char \*st) { sPtr = new char [length + 1 ]; assert( sPtr != 0); strcpy( sPtr, st); } void String::Poisk () { char st1[12]; cout << "Vvedite poisk: "; cin >>st1;  if (strstr(sPtr, st1) != NULL) cout <<"Takoe est' v stroke";  }              String::String(const char \*s) : length ( strlen ( s ) ) { cout<<"Conversion constructor: "<< s << '\n'; setString ( s ); }              String::String(const String &copy) : length ( copy.length ) { cout<<"Copy constructor: "<< copy.sPtr << '\n'; setString ( copy.sPtr ); }              String::~String() { delete [] sPtr; }  #endif    **3. Main.cpp**  #include <iostream.h> #include <iomanip.h> #include <conio.h> #include <string.h> #include "BASE.h" #include "Cint.h"    int main() { int i; String A1,A2,A3; String A4[4];  cout << "Vvedite - 2 raza 2 slova: \n";              cin >> A1; cin >> A2;              cout <<"Vvedite masiv strok (4): "; for (i = 0; i<4; i++){ cin>>A4[i];}  cout <<"Perviy masiv = " << A1<<'\n'; cout <<"Vtoroy masiv = " << A2<<'\n';              if (A1 == A2) cout << "A1 == A2 \n";              A3 = A1; A1 = A2;  cout<<"Posle A1 = A2, A1 : "<< A1 << endl;              if (A1 > A3) cout << "A1 > A3 \n";              if (A1 != A3) cout << "A1 != A3 "<<endl;              cout<<"Posle A1 += A3 = ";              A1 += A3;              cout<<A1<<endl; cout << "Masiv = ";              for (i = 0; i<4; i++){ cout<<A4[i]<<'\n';}              A1.Poisk(); cout<<endl;              getch();              return 0; }    **Результати работы программы**  http://life-prog.ru/programs/15c_clip_image002.jpg |

ЗАДАНИЕ №13

# ООП C++. Дружественные функции в Си

|  |  |
| --- | --- |
| |  | | --- | |  |   **Ззадание**  Используя раннее разработанный контейнер, расширить его функциональность за счет разработки дружественных классов для реализации доступа к его личным данным. Методами дружного класса обеспечить поиск.    **Выбор алгоритма решения задачи**  Чтобы расширить функциональность ранее разработанного контейнера (см. [здесь](http://life-prog.ru/view_programs.php?id=40&language=c&page=2)), включим в него дружеские функции.    **Описание решения задачи**  Для этого мы запишем в производном классе class String дружный класс, который мы назовем Drug. Объявления его записывается внутри класса String. И записывается он вот так: friend class Drug;    **Текст программы**  **1. Base.h**  #ifndef BASE\_H #include <iostream.h> #include <string.h> #include <assert.h> #define BASE\_H  class Employee { public: Employee (void) {};  friend ostream & operator<< (ostream & o, Employee &) ; friend istream & operator>> (istream & o, Employee &) ; };  #endif    **2. Cint.h**  #ifndef CINT\_H #define CINT\_H  #include <iostream.h> #include <string.h> #include "base.h"  class String : public Employee {  private: int length; char \*sPtr;  void setString (const char \*);  public: String (const char \* = ""); String (const String &); ~String(void);  void Poisk (); friend class Drug;  const String &operator= (const String &i) {          if ( &i != this) { delete [] sPtr; length = i.length; setString( i.sPtr ); } return \*this; }  bool operator ==(const String &i) const {return  strcmp (sPtr, i.sPtr) == 0;}  const String &operator+= (const String &);  friend ostream &operator<< (ostream &o, const String &s) { o <<s.sPtr;   return o;} friend istream &operator>> (istream &i, String &s) {          char temp[50]; i >> setw(50)>>temp; s = temp; return i; }  bool operator!= ( const String &right ) const   { return !( \*this == right ); } // проверка s1 > s2                                                               bool operator<( const String &i ) const      { return strcmp (sPtr, i.sPtr) < 0;}  bool operator> ( const String &i ) const      { return strcmp (sPtr, i.sPtr) > 0;}  virtual bool operator<=( const String &right ) const  { return !( right < \*this ) ; }  bool operator>=( const String &right)  { return !( \*this < right ); } }; const String &String::operator +=( const String &right) {  char \*tempPtr = sPtr; length += right.length; sPtr = new char [length +1]; assert ( sPtr != 0); strcpy ( sPtr, tempPtr); strcat ( sPtr, right.sPtr); delete [] tempPtr; return \*this; }  void String::setString(const char \*st) { sPtr = new char [length + 1 ]; assert( sPtr != 0); strcpy( sPtr, st); } void String::Poisk () { char st1[12]; cout << "Vvedite poisk: "; cin >>st1;  if (strstr(sPtr, st1) != NULL) cout <<"Takoe est' v stroke";  }  String::String(const char \*s) : length ( strlen ( s ) ) { cout<<"Conversion constructor: "<< s << '\n'; setString ( s ); }  String::String(const String &copy) : length ( copy.length ) { cout<<"Copy constructor: "<< copy.sPtr << '\n'; setString ( copy.sPtr ); }  String::~String() { delete [] sPtr; }  class Drug { public: String drug(String &); char \* p(String &); void Poisk(); };          String Drug::drug(String &t) {return t;}  char \* Drug::p(String &t) {          char \*temp; temp = t.sPtr; return temp;}  void Drug::Poisk () { char st2[12]; char st1[50]; Drug x; cout << "Vvedite poisk: "; cin >>st2; strcpy(st1, x.p); if (strstr(st1, st2) != NULL) cout <<"Takoe est' v stroke";  }  #endif    **3. Main.cpp**  #include <iostream.h> #include <iomanip.h> #include <conio.h> #include <string.h> #include "BASE.h" #include "Cint.h"    int main() { int i; String A1,A2,A3; String A4[4];  Drug tt;  cout << "Vvedite - 2 raza 2 slova: \n";  cin >> A1; cin >> A2;  cout <<"Experiment = "<<tt.drug(A1)<<" "<<tt.p(A1)<<'\n';  cout <<"Vvedite masiv strok (4): "; for (i = 0; i<4; i++){ cin>>A4[i];}  cout <<"Perviy masiv = " << A1<<'\n'; cout <<"Vtoroy masiv = " << A2<<'\n';  if (A1 == A2) cout << "A1 == A2 \n";  A3 = A1; A1 = A2;  cout<<"Posle A1 = A2, A1 : "<< A1 << endl;  if (A1 > A3) cout << "A1 > A3 \n";  if (A1 != A3) cout << "A1 != A3 "<<endl;  cout<<"Posle A1 += A3 = ";  A1 += A3;  cout<<A1<<endl; cout << "Masiv = ";  for (i = 0; i<4; i++){ cout<<A4[i]<<'\n';}  A1.Poisk(); cout<<"\nPoisk mojet delat' i friend class!!! \n"<<tt.p(A1); //          tt.poisk(); cout<<endl; getch();  return 0; }    **Результат работы программы**  http://life-prog.ru/programs/16c_clip_image002.jpg |

# ЗАДАНИЕ №14

# Разработка шаблонов функций в языке Си

|  |  |
| --- | --- |
| |  | | --- | |  |   **Задание**  Используя шаблоны, разработать семейство функций обработки разнотипных данных, размещаемых посредством ранее разработанного контейнерного класса (см. [здесь](http://life-prog.ru/view_programs.php?id=41&language=c&page=2)). Например, статические функции нахождения мах, мин, сортировка и т.д. Продемонстрировать на примере использованных шаблонов.    **Выбор алгоритма решения задачи**  Для реализации данной задачи необходимо использовать шаблоны. Шаблоны является одной из наиболее мощных возможностей С++ по созданию более универсального программного обеспечения, которое можно использовать повторно.    **Описание решения задачи**  Чтобы объявить шаблон, необходимо написать следующее: template < class T>. После этого написать необходимые параметры, которые нужно принять и написать, что должен делать тот или иной шаблон.    **Текст программы**  **1 base.h**  #ifndef base\_h #define base\_h #include<string.h> #include<iostream.h>  class Base { public: Base (){}; virtual ~Base(){}; virtual void show() = 0; };  #endif  **2 Cint.h**  #ifndef Cint\_h #define Cint\_h #include<string.h> #include<iostream.h> #include "Base.h"  struct text{ char n[50]; text \*Next; };  struct summa{ int n; summa \*Next; };  struct drob{ float n; drob \*Next; };    //////////////////////////////////////////////////  class Cint: public Base { public: Cint(); ~Cint();              void show() {cout << "S kakimi dannimy vi hotyte rabotat???" <<"\ntext - 1\n"<<"integer - 2\n"<<"drobniye - 3\n"<<"Exit - 4"<<endl;}; //////////////////////////////////////////////////              void Text();              text \*gettPtr(){return tPtr;};  //////////////////////////////////////////////////              void Summa();  summa \*getsPtr() {return sPtr;};  //////////////////////////////////////////////////              void Drob();  drob \*getdPtr() {return dPtr;};  //////////////////////////////////////////////////  private: text \*tPtr; summa \*sPtr; drob \*dPtr;  }; #endif    **3 Cint.cpp**  #include <iostream.h> #include <conio.h> #include "Cint.h"  template < class T> T \*Add(T \*posl, T \*begin);    template < class T> void Show(T \*begin);    template < class T> void FreeRam(T \*begin);  Cint::Cint(){ tPtr = new text; sPtr = new summa; dPtr = new drob; }  Cint::~Cint(){ tPtr = NULL; sPtr = NULL; dPtr = NULL; }  void Cint::Text(){ text \*Posl = NULL; int ch;  do{ cout<<" <1> - DOBAVIT ZAPIS \n" <<" <2> - PROSMOTR SPISKA \n" <<" <3> - EXIT \n";  cin>>ch;  switch (ch){  case(1): Posl = Add(Posl, gettPtr()); break; case(2): Show(gettPtr()); break; }  }while (ch != 3);  FreeRam(gettPtr()); getch(); }  ////////////////////////////////////////////////////////  void Cint::Drob(){ drob \*Posl = NULL; int ch;  do{ cout<<" <1> - DOBAVIT ZAPIS \n" <<" <2> - PROSMOTR SPISKA \n" <<" <3> - EXIT \n";  cin>>ch;  switch (ch){  case(1): Posl = Add(Posl, getdPtr()); break; case(2): Show(getdPtr()); break; }  }while (ch != 3);  FreeRam(getdPtr());           //   {очистка памяти} getch(); } ////////////////////////////////////////////////////////    void Cint::Summa(){ summa \*Posl = NULL; int ch;  do{ cout<<" <1> - DOBAVIT ZAPIS \n" <<" <2> - PROSMOTR SPISKA \n" <<" <3> - EXIT \n";  cin>>ch;  switch (ch){  case(1): Posl = Add(Posl, getsPtr()); break; case(2): Show(getsPtr()); break; }  }while (ch != 3);  FreeRam(getsPtr());           //   {очистка памяти} getch(); }  ////////////////////////////////////////////////////////  template < class T> T \*Add(T \*posl, T \*begin) { if (posl == NULL){ posl = begin; } else { posl->Next = new T; posl = posl->Next; }  cout << "Vvedite: \n"; cin >> posl->n; posl->Next = NULL;  return posl; }    template < class T> void Show(T \*begin) { cout << "Prosmotr spiska!!! \n"; cout <<"Name\n"; while (begin != NULL) { cout <<"\n"<< begin->n<<"\n"; begin = begin->Next; } getch(); }    template < class T> void FreeRam(T \*begin) { T \*prom; cout << "ochistka!!! \n"<<endl; prom = begin;  while ( prom != NULL) { begin = begin->Next; delete prom; prom = begin; }           }    **4. Main.cpp**  #include <string.h> #include <iostream.h> #include "Cint.h"  int main() { Cint F; int i; F.show(); cout<<"Vvedite cifry: "<<endl; cin >> i;    switch (i) { case 1 :  F.Text(); break;                          case 2 : F.Summa(); break;                          case 3 : F.Drob(); break; }  return 0; }    **Результат работы программы**  http://life-prog.ru/programs/17c_clip_image002.jpg  http://life-prog.ru/programs/17c_clip_image004.jpg |

ЗАДАНИЕ №15

**Программа: Шаблоны классов в языке Си**

|  |  |  |
| --- | --- | --- |
| |  | | --- | |  |   **Ззадание**  Разработать шаблон контейнерного класса для хранения и обработки разнотипных данных.  **Выбор алгоритма решения задачи**  Для решения этой задачи необходимо будет объявить шаблон контейнерного класса, т.е. контейнерный класс для разнотипных данных, которые выполняют одно и тоже действие, например, положить в стек, взять с стека.  **Описание решения задачи**  Для откровение шаблона контейнерного класса, необходимо написать так:  template< class T >  class Cint,  после чего писать производные шаблоны этого контейнерного класса, например, положить в стек:  template< class T >      void Cint<T>::Show (T \*begin) )    **Текст программы** **1. Base.h**  #ifndef base\_h #define base\_h #include<string.h> #include<iostream.h>  class Base { public: Base (){}; virtual ~Base(){}; virtual void show() = 0; };  #endif    **2. Cint.h**  #ifndef Cint\_h #define Cint\_h #include<string.h> #include<conio.h> #include<iostream.h> #include "Base.h"    //////////////////////////////////////////////////    template< class T >  class Cint: public Base { public: Cint(){tPtr = new T;}; ~Cint(){tPtr = NULL;};  void show() {cout << "S kakimi dannimy vi hotyte rabotat???" <<"\ntext - 1\n"<<"integer - 2\n"<<"drobniye - 3\n"<<"Exit - 4"<<endl;};  void Text(); T \*Add(T \*, T \*); void Show(T \*); void FreeRam(T \*);  T \*gettPtr(){return tPtr;};  //////////////////////////////////////////////////  private: T \*tPtr; };  /////////////////////////////////////////////////// ///////////////////////////////////////////////////  template < class T> T \*Add(T \*posl, T \*begin);  template < class T> void Show(T \*begin);  template < class T> void FreeRam(T \*begin);  //////////////////////////////////////////////////////  template< class T > void Cint<T>::Text(){  int ch; T \*Posl = NULL;        do{ cout<<" <1> - DOBAVIT ZAPIS \n" <<" <2> - PROSMOTR SPISKA \n" <<" <3> - EXIT \n";  cin>>ch;  switch (ch){  case(1): Posl = Add(Posl, gettPtr()); break; case(2): Show(gettPtr()); break; }  }while (ch != 3);  FreeRam(gettPtr()); getch(); }    ////////////////////////////////////////////////////////    template < class T> T \*Cint<T>::Add(T \*posl, T \*begin) { if (posl == NULL){ posl = begin; } else { posl->Next = new T; posl = posl->Next; }  cout << "Vvedite: \n"; cin >> posl->n; posl->Next = NULL;  return posl; }  //////////////////////////////////////////////////////////  template < class T> void Cint<T>::Show(T \*begin) { cout << "<Prosmotr spiska!!!> \n"; while (begin != NULL) { cout <<"\n"<< begin->n; begin = begin->Next; } cout<<"\n"<<endl; getch(); }  ////////////////////////////////////////////////////////////  template < class T> void Cint<T>::FreeRam(T \*begin) { T \*prom; cout << "ochistka!!! \n"<<endl; prom = begin;  while ( prom != NULL) { begin = begin->Next; delete prom; prom = begin; }           }  #endif    **3. Main.cpp**  #include <string.h> #include <iostream.h> #include "Cint.h"  struct text{  char n[50]; text \*Next; };  struct summa{ int n; summa \*Next; };  struct drob{ float n; drob \*Next; };  /////////////////////////////////////////////////////////  int main() {  Cint<text> F;  Cint<summa> G; Cint<drob> H; int i; F.show(); cout<<"Vvedite cifry: "<<endl; cin >> i;    switch (i) {     case 1 :  F.Text(); break;      case 2 : G.Text(); break;      case 3 : H.Text(); break; }  return 0; }    **Результат работы программы**  http://life-prog.ru/programs/18c_clip_image002.jpg  http://life-prog.ru/programs/18c_clip_image004.jpg   |  | | --- | | Просмотров: 8578 | |

# 6

# ЗАДАНИЕ №16

# Программа: использование стандартной библиотеки шаблонов на языке Си

|  |  |
| --- | --- |
| |  | | --- | |  |   **Задание**  Использовать ранее разработанный контейнер, методы поиска, сортировки функций. Контейнерные классы стандартные библиотеки шаблонов и стандартные алгоритмы.    **Выбор алгоритма решения задачи**  Для решения поставленной задачи, необходима программа находящаяся [здесь](http://life-prog.ru/view_programs.php?id=38&language=c&page=2) и программа, которая выполняет те же действия, но написана с использованием стандартных шаблонов (STL). Необходимо проверить время выполнения программы, написанной моим алгоритмом и проверить время выполнения программы, написанной с использованием **шаблонов STL**.    **Описание решения задачи**  Для откровение стандартного шаблона STL, необходимо написать следующее:  std::list< int > values;  Для внесения чисел до двунаправленного списка, необходимо написать:  values.push\_front ();  Для выполнения сортировки:  values.sort();    **Текст программы**  **1.1. telef.h**  #ifndef Telef\_h #define Telef\_h #include<string.h> #include<iostream.h>  /////////////////////////////////////////////////  struct Data{ int x; Data \*Next; Data \*Pred; };  //////////////////////////////////////////////////  class telef   { public: telef(){BeginPtr = new Data; EndPtr = NULL;}; ~telef(){BeginPtr = NULL;};  Data \*getBeginPtr(){return BeginPtr;}; Data \*getEndPtr(){return EndPtr;}; void setEndPtr(Data \*End){EndPtr = End;}; // тоже необходимо занести. void setBeginPtr(Data \*Begin){BeginPtr = Begin;};  void Add(Data \*,Data \*, int); void Del\_all(Data \*); void show(Data \*); void Poisk\_of\_chislo(Data \*); void Sort\_of\_chislo(Data \*); private: Data \*BeginPtr; Data \*EndPtr;  }; #endif    **1.2. telef.cpp**  #include<string.h> #include<iostream.h> #include<list> #include<algorithm> #include "telef.h"  void telef::Add(Data \*posl, Data \*begin,int temp) { if (posl == NULL){ posl = begin; posl->Pred = NULL; } else { posl->Next = new Data; posl->Next->Pred = posl; posl = posl->Next; }  posl->x = temp; posl->Next = NULL; setEndPtr(posl);  }  ////////////////////////////////////////////////////////  void telef::Del\_all(Data \*begin) { Data \*prom; prom = begin; while ( prom != NULL) { begin = begin->Next;       //когда все удалено, то количество необходимо поставить в ноль!!! delete prom; prom = begin; } cout << "DELETE ALL"<<endl; setBeginPtr(new Data);  setEndPtr(NULL); }  ////////////////////////////////////////////////////////  void telef::show(Data \*begin) { cout << "Prosmotr spiska!!! \n\n"; while ( begin != NULL) { cout << begin->x <<"  "; begin = begin->Next; } cout<<endl; }  ////////////////////////////////////////////////////////  void telef::Poisk\_of\_chislo(Data \*begin)  { int x = 0, temp; cout << "<POISK PO NAME> \n"; cout <<"chislo = 3 "; temp = 3; while ( begin != NULL) { if ( begin->x == temp ){ cout << "\nTakoe chislo naydeno"; x = 1; } begin = begin->Next; } if (x == 0 ) cout<<"\ntakogo chisla nety"; cout << endl; }  ////////////////////////////////////////////////////////  void telef::Sort\_of\_chislo(Data \*begin)  { int x\_temp;              Data \*Work; int k = 1, \_k = 1;  Work = begin;  while (k == 1){ begin = Work; while ( begin->Next != NULL) { if (begin->x > begin->Next->x) { x\_temp = begin->x; begin->x = begin->Next->x; begin->Next->x = x\_temp;  \_k = 0; }  begin = begin->Next; } if (\_k == 1) k = 0; \_k = 1; }  cout <<"\nSortirovka osyshestvlena!!!\n";  }    **1.3. Main.cpp**  #include<stdlib.h> #include<conio.h> #include<iostream.h> #include<windows.h> #include "telef.h"  long getTime( int tm ) { return labs( GetTickCount() - tm );  } /\*getTime()\*/    int main () { long tm1 = getTime( 0L ); cout<<" - "<<GetTickCount()<<"\n";  telef T; int chislo;  for (int i = 0; i<1000;i++){ chislo = rand() % 150; T.Add(T.getEndPtr(),T.getBeginPtr(),chislo); }  T.show(T.getBeginPtr()); T.Poisk\_of\_chislo(T.getBeginPtr()); T.Sort\_of\_chislo(T.getBeginPtr()); T.show(T.getBeginPtr());  T.Del\_all(T.getBeginPtr());    //   {очистка памяти}  cout<<" - "<<GetTickCount()<<"\n";  tm1 = getTime( tm1 ); cout << "\nTime: " << tm1 << endl; cout<<" Press key..."<<endl; getch();  return 0; }    **2.1. Main.cpp**  #include<iostream.h> #include<list> #include<iterator> #include<string.h> #include<conio.h> #include<windows.h> #include <stdlib.h>  using namespace std;  long getTime( int tm ) { return labs( GetTickCount() - tm );  } /\*getTime()\*/    int main()  { long tm1 = getTime( 0L );  cout<<" - "<<GetTickCount()<<"\n";  std::list< int > values; int x = 2,y;  for (int i = 0 ; i < 1000; i++ ){ y = rand() % 150; values.push\_front ( y ); }  cout << "valuex soderjit : "; list<int>::iterator p = values.begin(); while(p != values.end()) { cout << \*p<<" "; p++; }  values.sort(); cout << "\nvalues posle SORT soderjit: "; p = values.begin(); while(p != values.end()) { cout << \*p<<" "; p++; }  cout<<"\nPoisk: "; p = values.begin(); while(p != values.end()) { if ( \*p == 2) cout<<"\n2 - vhodit v spisok!!!"; p++; }  cout<<" - "<<GetTickCount()<<"\n";  tm1 = getTime( tm1 ); cout << "\nTime: " << tm1 << endl; cout<<endl; getch();  return 0; }      **Результат работы программы**  http://life-prog.ru/programs/19c_clip_image002.jpg  http://life-prog.ru/programs/19c_clip_image004.jpg  Вывод: изучил возможности, предоставляемые стандартной библиотекой шаблонов. Замерил время выполнения моего алгоритма и уже существующего, и замерил время выполнения стандартной библиотеки шаблонов. Получил такие результаты: моя программа заняла 78 условных единиц, а программа на основе стандартной библиотеки шаблонов заняла 47 условных единиц. Итак, стандартная библиотека шаблонов выполняет те же действия в 1,66 раз быстрее. Она оптимизирована, стандартизирована, универсальная. И для корректности программного обеспечения следует ее применять. |

ЗАДАНИЕ №17

**Программа на Visual C++ выводящяя в windows-окно эллипсы**

|  |  |
| --- | --- |
| |  | | --- | |  |   Создание программы на Visual C++, выводящей в windows-окно эллипсов с использованием всех возможных вариантов заливки.    В среде Visual C++ наберём текст программы :  // Устанавливаем режим строгой проверки типов // для Windows-программ #define STRICT #include<windows.h> #include<windowsx.h> #include<stdio.h> #include<string.h> #include<stdlib.h> #include<math.h> #define HNST HINSTANCE  WNDCLASS WCS;      // Структура класса окна int X,Y,          // Координаты для вывода текста функцией TextOut HatchStyle=0;      // Стандартный тип заливки char AName[]=" РГЗ №2  ";  // Обработчик сообщения WM\_DESTROY : сообщает Windows о завершении работы void OnDestroy (HWND) { PostQuitMessage(0L); } void OnMouseMove(HWND hW, int x, int y, UINT Flags)  {  if(Flags&MK\_LBUTTON) ::X=x,::Y=y,InvalidateRect(hW,0L,TRUE); } // Обработчик сообщения WM\_CREATE  : // загружает ресурс растрового изображения  BOOL OnCreat   (HWND hW,CREATESTRUCT FAR\*) { return TRUE; }  // Обработчик сообщения WM\_PAINT  : // Перерисовывает необходимые фрагменты окна программы void OnPaint   (HWND hW) { PAINTSTRUCT PS;RECT RC; RC.left=X-200,RC.top=Y-30,RC.right=X+200,RC.bottom=Y+50; HDC DC=BeginPaint(hW,&PS); HBRUSH             OLDB; UINT B[]={HS\_BDIAGONAL,HS\_CROSS,HS\_DIAGCROSS,HS\_FDIAGONAL,HS\_HORIZONTAL,HS\_VERTICAL}; char\*X[]={"HS\_BDIAGONAL","HS\_CROSS","HS\_DIAGCROSS","HS\_FDIAGONAL","HS\_HORIZONTAL","HS\_VERTICAL"};  OLDB=(HBRUSH)SelectObject(DC,CreateHatchBrush(B[::HatchStyle],RGB(0,0,0))); COLORREF R=SetBkColor(DC,RGB(255,200,100)),T=SetTextColor(DC,RGB(0,0,255)); Ellipse(DC,::X-150,::Y-80,::X+150,::Y+80),   DrawText(DC,"Заливка ",-1,&RC,DT\_CENTER), RC.bottom+=20,RC.top+=20,DrawText(DC,X[::HatchStyle],-1,&RC,DT\_CENTER), SetBkColor(DC,R),SetTextColor(DC,T),SelectObject(DC,OLDB); EndPaint(hW,&PS); }  // Обработчики сообщений мыши и клавиатуры : // формируют строку символов и передают её на контекст устройства void OnLeftUp  (HWND hW,int x,int y,UINT fl) { ::X=x,::Y=y,InvalidateRect(hW,0L,TRUE); }  // Обработчик сообщения WM\_LBUTTONDBLCLICK // ( двойной щелчок правой кнопкой мыши ) void OnLDBLCLICK(HWND hW, BOOL fDoubleClick, int x, int y, UINT /\*keyFlags\*/)  { if(fDoubleClick)::X=x,::Y=y,++HatchStyle%=6,InvalidateRect(hW,0L,TRUE);  }  // Оконная процедура WProc : обрабатывает все сообщения , // посылаемые окну hW // Устанавливаем WProc в качестве оконной процедуры главного окна программы  LRESULT CALLBACK WProc(HWND hW,UINT msg,WPARAM wParam,LPARAM lParam) {  switch(msg) // Для обработки сообщений можно использовать // макросы HANDLE\_MSG из <windowsx.h> // связывающие конкретные сообщения с соответствующими функциями-обработчиками  {  HANDLE\_MSG(hW,WM\_CREATE,OnCreat); HANDLE\_MSG(hW,WM\_DESTROY,OnDestroy); HANDLE\_MSG(hW,WM\_LBUTTONDBLCLK,OnLDBLCLICK); HANDLE\_MSG(hW,WM\_LBUTTONUP,OnLeftUp); HANDLE\_MSG(hW,WM\_PAINT,OnPaint); HANDLE\_MSG(hW,WM\_MOUSEMOVE,OnMouseMove); default:return DefWindowProc(hW,msg,wParam,lParam); // Для необрабатываемых сообщений // следует вызывать API-функцию DefWindowProc, // обеспечивающую стандартную обработку соответствующих сообщений } }  // Функция RG - заполняет поля структуры класса окна (WCS) // и регистрирует класс окна , // после чего можно создавать окна данного класса окна BOOL RG(HNST h) { WCS.style=CS\_HREDRAW|CS\_VREDRAW|CS\_DBLCLKS; WCS.lpfnWndProc=WProc; WCS.hInstance=h; WCS.cbClsExtra=WCS.cbWndExtra=0; WCS.hIcon=LoadIcon(0L,IDI\_APPLICATION); WCS.hCursor=LoadCursor(0L,IDC\_CROSS); WCS.hbrBackground=GetStockBrush(LTGRAY\_BRUSH); WCS.lpszMenuName=0L;WCS.lpszClassName=AName; return RegisterClass(&WCS); } #define def CW\_USEDEFAULT #define WStyle WS\_OVERLAPPEDWINDOW  // Функция Creat - создаёт окно и показывает его  HWND Creat(HNST h,int Cmd) { auto HWND hW=CreateWindow(AName,AName,WStyle,def,def,def,def,0L,0L,h,0L); if (hW) ShowWindow(hW,Cmd),UpdateWindow(hW); return  hW; }  // Функция WinMain - главная функция Windows-приложения  // вызываемая при запуске программы  int PASCAL WinMain(HNST h,HNST Prev,LPSTR ,int Cmd) { MSG M; register MSG\*P=&M; if(!Prev&&!RG(h)||!Creat(h,Cmd)) return FALSE; while(GetMessage(P,0,0,0))TranslateMessage(P),DispatchMessage(P); return P->wParam; }    **Результат работы программы:**  http://life-prog.ru/programs/20c_clip_image002.jpg  http://life-prog.ru/programs/20c_clip_image004.jpg |

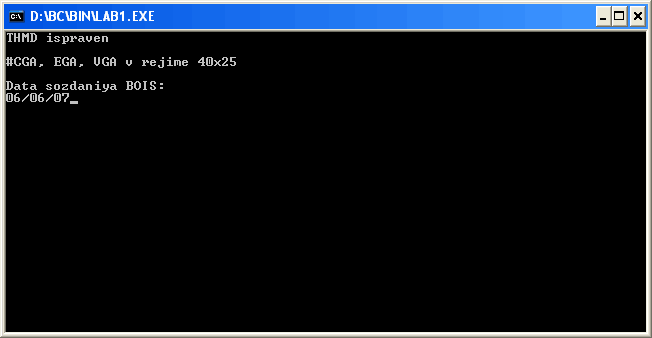
ЗАДАНИЕ №18

Написать программу, которая может: определить тип видеоконтроллера,  его режим и дату создания BIOS; проверить исправность НМД.

#include <iostream.h>  
#include<conio.h>  
void main()  
{  
            clrscr();  
            char far b, \*b1, \*b2;  
            int B, B1;

            outportb(0x70,0xE0);  
b = intportb(0x70);  
B = (int) \*b&8;  
if (B == 0) cout<<"HMD ispraven";  
else cout<<"HMD ne ispraven";  
cout<<"\n\n";  
b1 = ( char far \* ) 0x00400010;  
cout<<\*b1;  
B1 = (int) \*b&48;  
  
switch (B1) {  
case 0: cout<<"Ne ispol'zuetsa ili EGA"; break;         
case 16: cout<<"CGA, EGA, VGA v rejime 40x25";break;  
case 32: cout<<"CGA, EGA, VGA v rejime 80x25";break;  
case 48: cout<<"Monohromniy kontroler"; break;  
}  
cout<<"\n\n";  
  
b2 = ( char far \* ) 0xF000FFF5;  
cout<<"Data sozdaniya BOIS:\n";  
for(int i = 0;i<8; i++) cout<<\*(b2++);  
getch();  
}

**Результат работы программы:**



ЗАДАНИЕ №19

**Работа с графиков в Си**

|  |  |
| --- | --- |
| |  | | --- | |  |   Написать программу, которая на экране рисует линии с заданным цветом и байтом вывода. Чтобы тушилася первая линия, получался сдвиг вывода. Программа должна делать аппаратный сдвиг графики. Окно должно переходить из графического режима в текстовый.    #include<iostream.h> #include<conio.h> #include<dos.h> void main (){  asm mov ax, 0x10  asm int 0x10  /////////////////////////////////////////////  //Fon  inportb(0x3DA);  outportb(0x3C0,0);  outportb(0x3C0,0x35);  outportb(0x3C0,0x20);  int addr = 150\*80;  int i,j,q=0,x2,y2=0;   char far \*vmem = (char far \*)0xA0000000; char far \*vmem\_temp; vmem += addr; ////////////////////////////////////////////////// // 16 line for(i = 0;i<16; i++){ for(int j = 0;j<80; j++){   outportb(0x3CE,0x5); outportb(0x3CF,0x0); outportb(0x3CE,0x8); outportb(0x3CF,0xFF); outportb(0x3C4,0x2); outportb(0x3C5,5+7\*i ); \*vmem++ = 0x8E; } } ////////////////////////////////////////////////// //16 line + sdvig vmem\_temp = vmem;   for(i = 0;i<16; i++){ for(int j = 0;j<80; j++){ outportb(0x3CE,0x5); outportb(0x3CF,0x0); outportb(0x3CE,0x3); outportb(0x3CF,0x6);    //sdvig dannih outportb(0x3C4,0x2); outportb(0x3C5,5+7\*i ); //prirajenie tsveta \*vmem++ = 0x8E; } } //////////////////////////////////////////////////// //16 line XOR + sdvig for(i = 0;i<16; i++){ for(int j = 0;j<80; j++){ outportb(0x3CE,0x5); outportb(0x3CF,0x3); outportb(0x3CE,0x3); outportb(0x3CF,0x6);    //sdvig dannih outportb(0x3C4,0x2); outportb(0x3C5,5+7\*i ); //prirajenie tsveta \*vmem++ = 0x8E; } } getch(); /////////////////////////////////////////////// //potusheniye pervoy stroki char far \*vmem1 = (char far \*)0xA0000000; vmem1 += addr; for(j = 0;j<80; j++){          outportb(0x3CF,0x3); outportb(0x3C5,0x35); \*vmem1++ = 0xFF; } getch(); /////////////////////////////////////////////// //rejim 1 zapolneniye baytom outportb(0x3CE,0x5); outportb(0x3CF,1); char c;    for(i = 0;i<16; i++){ delay(30); for(int j = 0;j<760; j++){ c=\*(vmem + 5\*80 + 5 + j); \*(vmem + (y2\*80 + x2 + j))=c; outportb(0x3D4,0xC); outportb(0x3D5,0x2); outportb(0x3D4,0xD); outportb(0x3D5,0x20); inportb(0x3DA); outportb(0x3C0,0x0); outportb(0x3C0,0x35); outportb(0x3C0,0x20); } } ///////////////////////////////////////////////// //aparatniy sdvig  while (int(c) != 27){ c = getch(); vmem = vmem\_temp; if (int(c) == 77) q++; if (int(c) == 75) q--;;   for(i = 0;i<16; i++){ for(int j = 0;j<80; j++){ outportb(0x3CE,0x5); outportb(0x3CF,0x0); outportb(0x3CE,0x3); outportb(0x3CF,q);    //sdvig dannih outportb(0x3C4,0x2); outportb(0x3C5,5+7\*i ); //prirajenie tsveta \*vmem++ = 0x8E; } } } inportb(0x3DA); outportb(0x3C0,2); outportb(0x3C0,0); outportb(0x3C0,0x20); asm mov ax,3; asm int 0x10; getch(); } |

ЗАДАНИЕ №20

**Натуральные числа в языке СИ или СИ++**

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | |  |   Дано натуральное число эн. Каждую цифру числа заменить на дополнение ее до десяти. Подсчитать количество цифр в изображении числа.  Написать программу на языке C или C++.  Здесь я покажу разницу между языком СИ и языком СИ++.    Мы можем использовать для переменных тип unsigned int - это для целых неотрицательных числах. Т.е.  unsigned int x;  Так как в задании сказано натуральные числа. А можем просто  int x;    Пример программы на языке СИ написанный на Visual Studio 2010:  #include "stdafx.h" #include "conio.h"  int \_tmain(int argc, \_TCHAR\* argv[]) { unsigned int x; int inc = 1; // переменная для степени int sum = 0; // подсчет суммы чисел int temp; // переменная для вычислений int x2 = 0; // переменная куда будет записываться новое число (к примеру: 999). bool b = true; // переменная для выхода из цикла while  printf("Vvedite tseloe chislo: "); scanf ("%d", &x); // вводим число while (b) { temp = x/inc; if (x/inc < 10) // ищем максимальную разрядность b = false; else inc = inc\*10; // разрядность числа sum++; // подсчет суммы количества чисел } for(int i = 0; i<sum; i++) // цикл изменения цисла к прим. 111 на 999 { temp = x/inc;  x = x - temp\*inc; temp = 10 - temp; // изменяем 10 - число x2 = x2 + temp\*inc;  inc = inc/10; // уменьшаем разрядность } printf("Poluchenoe chislo: %d\n",x2); printf("Kolichestvo chisel: %d",sum); \_getch(); return 0; }  Результат работы программы:  http://life-prog.ru/programs/untitled65465.jpg   ЗАДАНИЕ №20.1  Пример программы на языке СИ++ тоже написанный на Visual Studio 2010:    #include "stdafx.h" #include "conio.h" #include <iostream>  int \_tmain(int argc, \_TCHAR\* argv[]) { unsigned int x; int inc = 1; // переменная для степени int sum = 0; // подсчет суммы чисел int temp; // переменная для вычислений int x2 = 0; // переменная куда будет записываться новое число (к примеру: 999). bool b = true; // переменная для выхода из цикла while  //readline(x); std::cout<<"Vvedite tseloe chislo: "; std::cin>>x; // вводим число while (b) { temp = x/inc; if (x/inc < 10) // ищем максимальную разрядность b = false; else inc = inc\*10; // разрядность числа sum++; // подсчет суммы количества чисел }  for(int i = 0; i<sum; i++) // цикл изменения цисла к прим. 111 на 999 { temp = x/inc;  x = x - temp\*inc; temp = 10 - temp; // изменяем 10 - число x2 = x2 + temp\*inc;  inc = inc/10; // уменьшаем разрядность } std::cout<<"Poluchenoe chislo: "<<x2<<"\n"; std::cout<<"Kolichestvo chisel: "<<sum<<std::endl; \_getch(); return 0; }  Результат работы программы:  http://life-prog.ru/programs/untitled654651.jpg  Итак, мы видим, что отличия почти нету.  ЗАДАНИЕ №21 | | |
| 2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255  256  257  258  259  260  261  262  263  264  265  266  267  268  269  270  271  272  273  274  275  276  277  278  279  280  281  282  283  284  285  286  287  288  289  290  291  292  293  294  295  296  297  298  299  300  301  302  303  304  305  306  307  308  309  310  311  312  313  314  315  316  317  318  319  320  321  322  323  324  325  326  327  328  329  330  331  332  333  334  335  336  337  338  339  340  341  342  343  344  345  346  347 | | Блочная сортировка чисел массива.работает запись /чтение файла  #include <iostream>  #include <conio.h>  #include <locale>  #include <time.h>  #include <fstream>  #include <math.h>  #include <ctype.h>  #include <ctime>  #include <algorithm>  #define D 4  #define P 10  using namespace std;  void zahita()  {      while(cin.fail())    {     cin.clear();          cin.ignore(256,'**\n**');          cout<<"Такого пункта нет в меню программы**\n**Повторите попытку**\n\n\a**";      }    }      int main()    {    setlocale(LC\_ALL, "Russian");      time\_t t;      time(&t);      tm\* local = localtime(&t);      char Hours[10] = {0};      char Minutes[10] = {0};      char Date [20] = {0};      strftime(Hours, sizeof(Hours)/sizeof(Hours[0]), "%H", local);      strftime(Minutes, sizeof(Minutes)/sizeof(Minutes[0]), "%M", local);      strftime(Date, sizeof(Date)/sizeof(Date[0]), "%d %B %Y", local);      cout << "Текущее время и дата: " << Hours << ":" << Minutes << " " << Date <<"**\n**"<< endl;  cout << "Молодой Алексей.Группа БД-212.**\n\n\t**" ;  cout << "==========================**\n\t**БЛОЧНАЯ СОРТИРОВКА МАССИВА**\n\t**==========================**\n\n**" ;  cout << "Убедитесь в наличии файлов:**\n**\ massivG.txt - сгенерированный массив**\n** massivS.txt - сортированный массив № 1**\n** massivS2.txt - сортированный массив № 2 **\n\n**" ;  while (true)    {  Metka:  cout << " ==============================================================================**\n**";  cout << " \*  1.Сгенерировать массив чисел, сортировать и записать его в файл           \***\n**";  cout << " \*  2.Считать массив чисел из файла и отобразить его на консоли               \***\n**";  cout << " \*  3.Запустить процесс сортировки из файла                                   \***\n**";  cout << " \*  0.Выход                                                                   \***\n**";  cout << " ==============================================================================**\n\n**";  cout << "Меню - ";  int dv;  cin >> dv;  cout << "**\n**";  while(cin.fail())    {     cin.clear();          cin.ignore(256,'**\n**');                  cout<<"Такого пункта нет в меню программы!**\n**Повторите попытку**\n\n\a**";                  goto Metka;    }     if (dv < 0|dv > 3)    {          cout <<"Такого пункта нет в меню программы!**\n**Повторите попытку**\n\n\a**";    }  else  switch (dv)    {  case 1:    {      int N, b[P], i, j, k, m, n,sort,zap;      float x, v;      cout << "Введите размер массива... ";      cin >> N;      int \*a = new int[N];      if (N==0|N>100000000)     {         cout <<"Размер массива нулевой или очень большого размера!**\n**Повторите попытку**\n\n\a**";      break;     }      else      srand(time(NULL)|clock());      for (int i = 0; i < N; i++)     {        a[i]=rand()%10000;     }      cout << "Полученный массив:**\n\n**";      for (int i = 0; i < N; i++)     {      cout << a[i] << " " <<"**\t**";     }      cout << "**\n**";     ofstream massivg("massivG.txt");      massivg.is\_open();     for (int i = 0; i < N; i++)     {      massivg << a[i]<< " "<<"**\n**";     }        massivg.close();      cout <<"Массив сгенерирован и записан в файл massivG.txt **\n\n**";      cout<<"Запустить процесс сортировки?**\n**";      cout<<"1.Да**\n**2.Нет**\n**3.Выход в главное меню**\n\n**Меню - ";      cin>>sort;      zahita;      cout<<"**\n**";      if(sort==3)      {          goto Metka;      }      if(sort==1)      {          time\_t time1,time2;          int vrem;          time1=clock();              for (m = 0; m<D; m++)              {              for (i = 0; i<P; i++)              {              b[i] = 0;              }              for (i = 0; i<N; i++)              {              v = a[i];              for (n = 0; n<m; n++)              {                  v = v / P;              }              k = ((int)floor(v)) % P;              x = a[i];              for (j = i; j>b[k]; j--)              {                  a[j] = a[j - 1];              }              a[b[k]] = x;              for (j = k; j<P; j++)              {                  b[j] = b[j] + 1;              }                }                  }        cout <<"Сортированный массив:**\n\n**";      for (i = 0; i<N; i++)      {          cout << a[i] << " **\t**";      }          time2=clock();          vrem =(time2-time1)/60;          cout << "**\n**";          cout<<"На выполнение операции затрачено: " << vrem << " миллисекунд.**\n\n**";      }      cout<<"Записать полученный массив в файл?**\n**";      cout<<"1.Да**\n**2.Нет**\n**3.Выход в главное меню**\n\n**Меню - ";      cin>>zap;      cout<<"**\n**";      if(zap==3)      {          goto Metka;      }      if(zap==1)      {      ofstream massivS("massivS.txt");      massivS.is\_open();      for (int i = 0; i < N; i++)      {      massivS << a[i]<< " ";      massivS << endl;      }      massivS.close();      cout <<"Массив записан в файл massivS.txt **\n\n**";      break;      }      else      {      break;      }       }  case 2:      {      int file;           cout<<"Какой файл считать ?**\n\n**";           cout<<"1.Сгенерированный массив**\n**2.Сортированный массив**\n**3.Выход в главное меню**\n\n**Меню - ";      cin>>file;      cout <<"**\n**";      if(file==3)      {          goto Metka;      }      if(file==1)      {      ifstream massivG("massivG.txt");      string line;      if (!massivG)      {          cout << "Файл не найден!**\n\a**" << endl;          break;      }          cout << "Массив считан из файла!**\n\n**";      while (!massivG.eof())      {          getline(massivG, line);          int size = line.size();          cout << line<<"**\t**";      }          cout <<"**\n**";      massivG.close();      break;      }      if(file==2)      {      ifstream massivS("massivS.txt");      string line;      if (!massivS)      {          cout << "Файл не найден!**\n\a**" << endl;          break;      }          cout << "Массив считан из файла!**\n\n**";      while (!massivS.eof())      {          getline(massivS, line);          int size = line.size();          cout << line<<"**\t**";      }          cout <<"**\n**";      massivS.close();      }      break;      }  case 3:      {  float x,v;  int  b[P], i, j, k, m, n,sort, zap;  int K=-1;//счетчик      ifstream massivG("massivG.txt");      if (!massivG){          cout << "Файл не найден!**\n\a**" << endl;          break;      }      string line;      cout << "Массив считан из файла!**\n\n**";      while (!massivG.eof())      {          getline(massivG, line);          int size = line.size();          cout << line<<"**\t**";      K++;      }      massivG.close();          int \*d = new int[K]; // Выделение памяти для массива          massivG.open("massivG.txt");                for (int i = 0; !massivG.eof(); i++)                {                    // Заполнение массива и вывод значений его элементов          massivG >> d[i];                }          massivG.close();          cout <<"**\n**";        cout << "Элементов в массиве : " << K <<'**\n**';         cout << "**\n**";        cout<<"Запустить процесс сортировки?**\n**";      cout<<"1.Да**\n**2.Нет**\n**3.Выход в главное меню**\n\n**Меню - ";      cin>>sort;      cout <<"**\n**";      if(sort==3)      {          goto Metka;      }      if(sort==1)      {          time\_t time1,time2;          int vrem;          time1=clock();              for (m = 0; m<D; m++)              {              for (i = 0; i<P; i++)              {              b[i] = 0;              }              for (i = 0; i<K; i++)              {              v = d[i];              for (n = 0; n<m; n++)              {                  v = v / P;              }              k = ((int)floor(v)) % P;              x = d[i];              for (j = i; j>b[k]; j--)              {                  d[j] = d[j - 1];              }              d[b[k]] = x;              for (j = k; j<P; j++)              {                  b[j] = b[j] + 1;              }               }                }        cout <<"Сортированный массив**\n\n**";      for (i = 0; i<K; i++)      {          cout << d[i] << " **\t**";      }          time2=clock();          vrem =(time2-time1)/60;          cout << "**\n**";          cout<<"На выполнение операции затрачено: " << vrem << " миллисекунд.**\n\n**";      }      cout<<"Записать полученный массив в файл?**\n**";      cout<<"1.Да**\n**2.Нет**\n**3.Выход в главное меню**\n\n**Меню - ";      cin>>zap;      cout <<"**\n**";      if(zap==3)      {          goto Metka;      }      if(zap==1)      {      ofstream massivS("massivS2.txt");      massivS.is\_open();      for (int i = 0; i < K; i++)      {      massivS << d[i]<< " ";      massivS <<endl;      }      massivS.close();      cout <<"Массив записан в файл massivS2.txt **\n\n**";      break;      }      else      {      break;      }       }      case 0:      {          break;        }         }         if(dv==0)             break;  }    } |  |

Задание № 22

Найти среднее геометрическое из набора случайных чисел.

**Кликните здесь для просмотра всего текста**

| **C++**[Выделить код](http://www.cyberforum.ru/) | |
| --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36 | #include <iostream>  #include <math.h>  #include <locale>  using namespace std;    void main(){      while (true){      setlocale(0,"");           cout << "Дан набор целых чисел. Найти среднее геометрическое.**\n\n**";      int number;      double res = 1;           cout << "Введите количество чисел...  ";      while (!(cin >> number)) {           cin.clear();           cin.ignore(numeric\_limits<streamsize>::max(), '**\n**');           cout << "Можно вводить только числа! ";}      if (number == 0){           cout << "Ноль не допустимое число. **\n**";}      else          if (number >75){          cout << "Очень большое число, я не смогу посчитать.**\n**";}      else{      if (number <0){          number = number\*(-1);}      int \*mass = new int[number];          srand(time(0));      for(int i = 0; i < number; i++){          mass[i] = rand();          cout <<  mass[i] << " **\n**";}      for (int i = 0; i < number; i++){          res = res \*mass[i];}          res = pow(res, 1.0/number);          cout << "Среднее геометрическое: "<< res <<endl; }          cout << "--------**\n**";          cout << "Молодой Алексей, гр. БД-212**\n**Информатика и программирование **\n**Нажмите Enter... **\n\n\n**";  }} | |  |

Шифрование текста. Чтение из файла, шифровка, дешифровка и запись в файл.

**Кликните здесь для просмотра всего текста**

| **C++**[Выделить код](http://www.cyberforum.ru/) | |
| --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133 | #include <iostream>  #include <conio.h>  #include <locale>  #include <time.h>  #include <fstream>          int main()  {  setlocale(LC\_ALL, "Russian");  using namespace std;  cout << "Молодой Алексей.Группа БД-212.**\n**" ;  cout << "Программа для шифровки и дешифровки текста из файла.**\n\n**" ;  cout << "Убедитесь в наличии файлов:**\n** f.txt(чтение текста)**\n** g.txt(запись зашифр. текста)**\n** h.txt(запись расшифр. текста)**\n\n**" ;  while (true)  {  cout << "Выберите действие :**\n\n**";  cout << "1.Шифровать текст **\n**";  cout << "2.Дешифровать текст **\n**";  cout << "3.Отобразить на консоли зашифрованный и расшифрованный текст **\n**";  cout << "0.Выход **\n\n**";      int dv;  while (!(cin >> dv)) {           cin.clear();           cin.ignore(numeric\_limits<streamsize>::max(),'**\n**' );           cout << "Можно вводить только число! Выберите действие :**\n\n**";           cout << "1.Шифровать текст **\n**";           cout << "2.Дешифровать текст **\n**";           cout << "3.Отобразить на консоли зашифрованный и расшифрованный текст **\n**";           cout << "0.Выход **\n\n**";           cin >>dv;    }    switch (dv)  {  case 1:      {cout << "ШИФРОВАНИЕ ТЕКСТА **\n\n**";      cout << "Текст будет считан из файла f.txt **\n\n**";      system ("pause");      ifstream f("f.txt");      ofstream g("g.txt");      string line;      if (!f || !g)      {   cout << "Файлы не найдены!**\n\n\n**" << endl;          system("pause");          return 1;}      while (!f.eof())      {          getline(f, line);          int size = line.size();            for (int i = 0; i < size; i++)          {              if (line[i] == 'я') line[i] = 'а';              else if (line[i] == 'Я') line[i] = 'А';              else line[i] = line[i] + 1;          }          g << line;      }      f.close();      g.close();      cout << "Текст зашифрован и помещен в файл g.txt **\n\n**";      system("pause");      break;      }  case 2:      {cout << "ДЕШИФРОВАНИЕ ТЕКСТА **\n\n**";      cout << "Текст будет считан из файла g.txt **\n\n**";      system ("pause");      ifstream g("g.txt");      ofstream h("h.txt");      string line;      if (!g || !h)      {   cout << "Файлы не найдены!**\n\n\n**" << endl;          system("pause");          return 1;}      while (!g.eof())      {          getline(g, line);          int size = line.size();            for (int i = 0; i < size; i++)          {              if (line[i] == 'а') line[i] = 'я';              else if (line[i] == 'А') line[i] = 'Я';              else line[i] = line[i] - 1;          }          h << line;      }      g.close();      h.close();      cout << "Текст расшифрован и помещен в файл h.txt **\n\n**";      system("pause");      break;      }  case 3:{             ifstream g("g.txt");             ifstream h("h.txt");             string line;             if (!g || !h)             {                 cout << "Файл не найдены!**\n\n\n**" << endl;                 system("pause");                 return 1;             }             cout << "Зашифрованный текст**\n\n**";             while (!g.eof())             {                 getline(g, line);                 int size = line.size();                 cout << line << "**\n**";             }             cout << "Расшифрованный текст**\n\n**";             while (!h.eof())             {                 getline(h, line);                 int size = line.size();                 cout << line << "**\n\n**";             }             g.close();             h.close();             break;}    case 0:      {break;}  }if(dv==0)break;}    } | |  |

Даны координаты точек Х и У. Найти крайнюю верхнюю точку и крайнюю нижнюю точки.

**Кликните здесь для просмотра всего текста**

| **C++**[Выделить код](http://www.cyberforum.ru/) | |
| --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80 | #include <iostream>  #include <locale>  #include <time.h>  #include <fstream>  #include <vector>  #include <string>  #include <conio.h>    using namespace std;    struct coord\_t {      double x;      double y;  };  int main()   {  setlocale(LC\_ALL, "Russian");  cout << "Молодой Алексей.Группа БД-212.**\n\n\t**" ;  cout << "===========================================================**\n\t**Поиск левого верхнего и правого нижнего угла по координатам**\n\t**===========================================================**\n\n**" ;  cout << "Убедитесь в наличии файлов!**\n\n**" ;  cout << "Записать координаты точек в файл**\n**";  int x, y, zap;  Metka:  cout << "Введите координату Х: ";  cin >> x;  cout << "Введите координату Y: ";  cin >> y;  ofstream points("points.txt",ios\_base::app);      points.is\_open();      points << x <<" "<< y<<"**\n**";      points.close();  cout << "Добавить еще координаты?**\n\n**";  cout << "1. Да**\n**";  cout << "2. Нет**\n\n**";  cout << "Выбор -  ";  cin >>zap;  if(zap==1)  {      goto Metka;  }  else  {      vector<struct coord\_t> coord;      ifstream points("points.txt");      points.is\_open();      struct coord\_t xy;      for (int i=1;!points.eof(); ++i) {              points >> xy.x;              points >> xy.y;              coord.push\_back(xy);              cout << "Координата " << i <<" **\t**"<< " X: " << xy.x <<"**\t**"<< " Y: " << xy.y << endl;        }      cout<<"**\n**";        double maxx=-1000000, maxy=-1000000, minx=1000000, miny=1000000;        for (int i=0; i<coord.size(); i++)  {  if (coord[i].x>maxx)     maxx=coord[i].x;  if (coord[i].x<minx)     minx=coord[i].x;  if (coord[i].y>maxy)     maxy=coord[i].y;  if (coord[i].y<miny)     miny=coord[i].y;    }            cout <<"Левый верхний угол имеет координаты  " << minx << " "<< maxy<<endl;      cout <<"Правый верхний угол имеет координаты " << maxx << " "<< miny<<endl;        \_getch();  }   } | |  |