**Лабораторна робота №10**

**Виконав студент групи ТІ-92 1-го курсу**

**П.І.Б Черноусов Денис Ігорович**

Завдання :

Задача 11.1. Создайте манипулятор для вывода чисел в научной нотации с символом Е в верхнем регистре.

Задача 11.2. Напишите программу для копирования текстового файла. В процессе копирования преобразуйте каждый символ табуляции в соответствующее число пробелов.

Задача 11.3. Напишите программу для поиска в текстовом файле слова, заданного в командной строке. После выполнения программы на экране должно появиться число, обозначающее, сколько раз данное слово найдено в файле. Для простоты считайте следующее: все, что с обеих сторон окружено пробелами, есть слово.

Задача 11.4. Напишите инструкцию, которая устанавливает указатель записи на 81-й байт в файле, связанном с потоком out.

Задача 11.5. Напишите программу создания базового класса Num. В этом классе должно храниться целое и определяться виртуальная функция shownum(). Создайте два производных класса outhex и outoct, которые наследуют класс Num. Функция shownum() должна быть переопределена в производных классах так, чтобы осуществлять вывод на экран значений, в шестнадцатеричной и восьмеричной системах счисления соответственно.

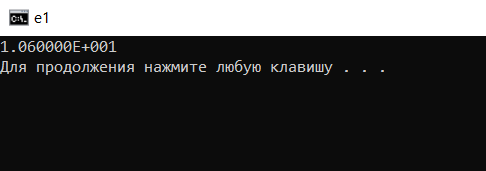
Задача 11.6. Напишите программу, в которой базовый класс Dist используется для хранения в переменной типа double расстояния между двумя точками. В классе Dist создайте виртуальную функцию trav\_time(), которая выводит на экран время, необходимое для прохождения этого расстояния с учетом того, что расстояние задано в милях, а скорость равна 60 миль в час. В производном классе Metric переопределите функцию trav\_time() так, чтобы она выводила на экран время, необходимое для прохождения этого расстояния, считая теперь, что расстояние задано в километрах, а скорость равна 100 километров в час.

Задача 11.7. Проведите эксперимент с двумя программами из Example 11.4 и Example 11.5. Попытайтесь создать объект, используя класс area из Example 11.4, и проанализируйте сообщение об ошибке. В Example 11.5 попытайтесь удалить переопределение функции func() внутри класса derived2. Убедитесь, что тогда действительно будет использоваться та версия функции func(), переопределение которой находится в классе derived1. Что произойдет в Example 11.5 при удалении переопределения функции func() из класса derived1? Будет ли при этом программа компилироваться и запускаться? Если да, то почему?

Задача 11.8. Программа из Example 11.6 при компиляции при помощи команды g++ -Wall ex116.cpp выдает предупреждения. Объясните причину появления этих предупреждений и для их устранения внесите необходимые правки в код данного примера. Совет 1: Функция main() в программе со списками (см. Example 11.6 ) только иллюстрирует работу классов. Для изучения динамического полиморфизма попробуйте использовать в этой программе следующую функцию main(): int main() { list \*p; queue q\_ob; stack s\_ob; char ch; for (int i=0; i<< "Stack or Queue? (S/Q):"; cin >> ch; ch = tolower(ch); if (ch == 'q') p = &q\_ob; else p = &s\_ob; p->store(i); } cout << "Enter T to terminate\n"; for (;;) { cout << "Remove from Stack or Queue? (S/Q):"; cin >> ch; ch = tolower(ch); if (ch == 't') break; if (ch == 'q') p = &q\_ob; else p = &s\_ob; cout << p->retrieve() << '\n'; } cout << '\n'; return 0; } Добавьте список другого типа к программе из Example 11.6 . Эта версия должна поддерживать отсортированный (в порядке возрастания) список. Назовите список sorted. Совет 2: Обдумайте случаи, в которых следует использовать динамический полиморфизм, чтобы упростить решение разного рода проблем.

Задача 11.9. Расширьте пример со списком, (см. Example 11.6 ) так, чтобы в нем перегружались операторы + и -. Используйте оператор + для внесения элемента в список, а оператор - для выборки элемента из списка..

Завдання 11.1



Код :

#include <iostream>

#include <iomanip>

using namespace std;

ostream & ss (ostream & s){

s << scientific ;

s << uppercase ;

return s ;

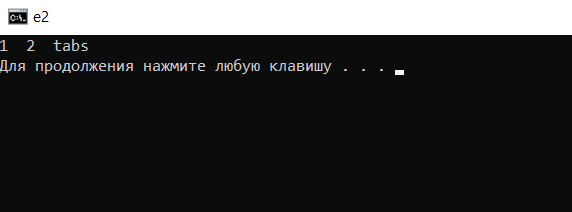
}

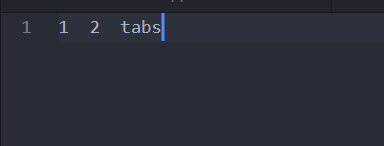
int main(){

cout << ss << 10.60 ;

}

Завдання 11.2





Код:

#include <iostream>

#include <fstream>

using namespace std ;

int main(){

ofstream fo ("e2.txt");

fo << "1 2 tabs" ;

fo.close();

ifstream fi ("e2.txt");

char ch[20] ;

while (!fi.eof()){

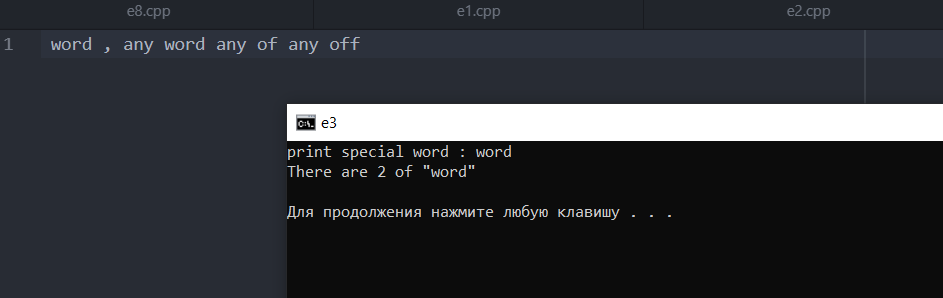
fi.getline(ch,20);

cout << ch ;

}

}

Завдання 11.3



Код :

#include <iostream>

#include <fstream>

#include <cstring>

using namespace std ;

int main( ) {

ofstream out ("e3.txt");

out << " word , any word any of any off ";

out.close();

char specialword [10] ;

cout << "print special word : " ;

cin >> specialword ;

ifstream in ("e3.txt");

char a [10][10];

int i = 0 ;

int counter = 0 ;

while (!in.eof()) {

in >> a[i];

if (strcmp( a[i], specialword ) == 0 ) counter ++ ;

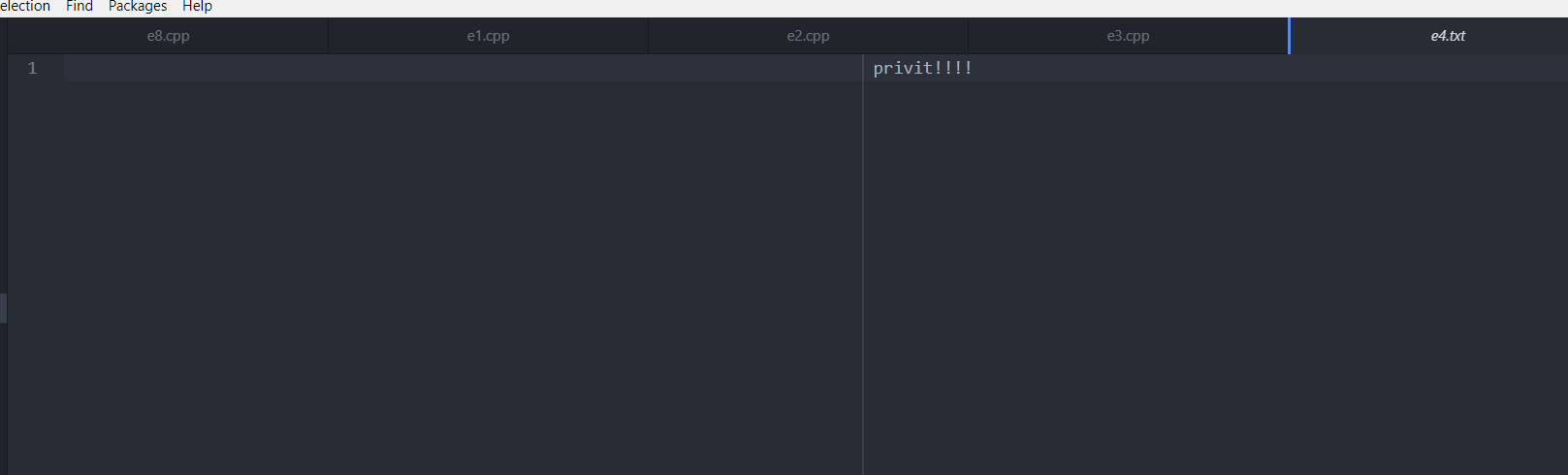
i++ ;

}

printf("There are %d of \"%s\"\n", counter , specialword );

}

Завдання 11.4



Код :

#include <iostream>

#include <fstream>

using namespace std ;

int main(){

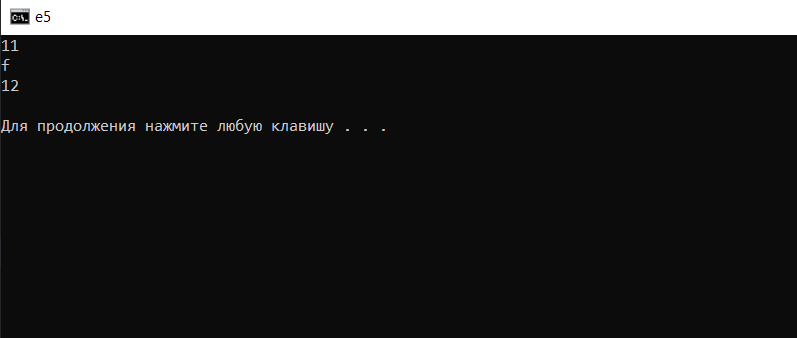
ofstream f1 ("e4.txt");

f1.seekp (81 , ios :: beg );

f1 << "privit!!!!" ;

}

Завдання 11.5



Код :

#include <iostream>

#include <iomanip>

using namespace std ;

class Num {

int number ;

public:

Num( int x ) { number = x ;}

virtual fun(){

cout << dec << number << endl ;

}

};

class octnum : public Num {

int number ;

public:

octnum( int x ) : Num(x) { number = x ;}

fun(){

cout << oct << number << endl ;

}

};

class hexnum : public Num {

int number ;

public:

hexnum( int x ) : Num(x) { number = x ;}

fun(){

cout << hex << number << endl ;

}

};

int main(){

octnum o(9);

hexnum a(15);

Num n(12);

Num \* ptr ;

ptr = & o ;

ptr->fun();

ptr = & a ;

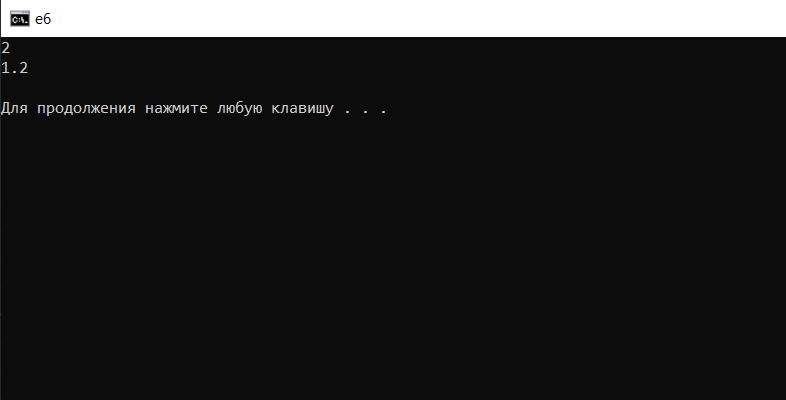
ptr->fun();

ptr = & n ;

ptr->fun();

}

Завдання 11.6



Код :

#include <iostream>

#include <iomanip>

using namespace std;

class dist{

double di ;

public:

dist (double d){di = d ;}

virtual trav\_dist(){

cout << di / 60 << endl ;

}

};

class met : public dist {

double di;

public:

met(double d) : dist(d){ di = d ; }

trav\_dist(){

cout << setprecision(3) << di / 100 << endl ;

}

};

int main(){

dist o(120);

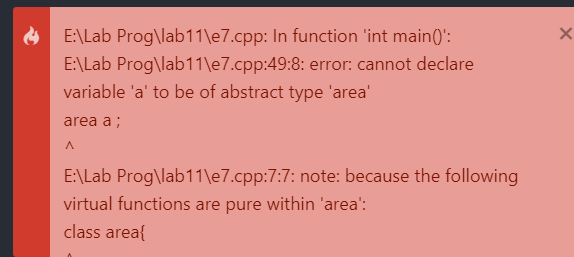
o.trav\_dist();

met a(120);

a.trav\_dist();

}

Завдання 11.7



Код :

#include <iostream>

#include <iomanip>

using namespace std ;

class area{

double dim1 , dim2 ;

public:

void setarea(double d1 , double d2){

dim1 = d1 ;

dim2 = d2 ;

}

void getdim(double& d1 , double &d2){

d1 = dim1;

d2 = dim2 ;

}

virtual double getarea() = 0 ;

};

class rectangle: public area {

public :

double getarea(){

double d1 , d2 ;

getdim (d1,d2);

return d1\*d2;

}

};

class triangle : public area {

public :

double getarea(){

double d1 , d2 ;

getdim(d1,d2);

return 0.5 \* d1 \* d2 ;

}

};

int main(){

area \* p;

rectangle r ;

triangle t ;

r.setarea(3.3,4.5);

t.setarea(4.0,5.0);

p=&r ;

cout << "Rectangle has area : " << p->getarea() << '\n';

p= &t ;

cout << "Triangle has area : " << p->getarea() << '\n' ;

area a ;

/\* не можна так ініціалізувати обєкт, бо там є

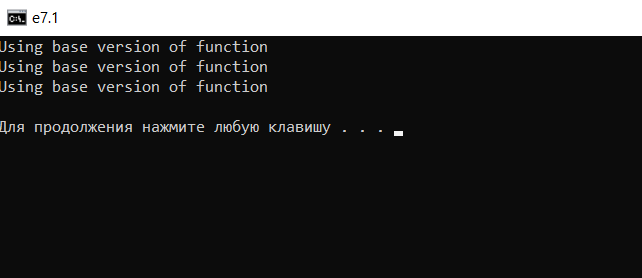
віртуальна функція, яка є pure (чистою, пустою)

але якщо замінити virtual double getarea() = 0 ;

на virtual double getarea() {return 0;} , то все спрацює \*/

}

Завдання 11.7.1



Код :

#include <iostream>

#include <iomanip>

using namespace std ;

class base{

public:

virtual void fun(){

cout << "Using base version of function\n";

}

};

class der1: public base {

public :

/\*void fun(){

cout << "Using derived1 version of function\n";

} \*/

};

class der2 : public der1 {

public :

/\* void fun(){

cout << "Using derived2 version of function\n";

} \*/

};

int main(){

base \* p;

base obj ;

der1 obj1;

der2 obj2;

p = & obj1 ;

p-> fun();

p = & obj ;

p-> fun();

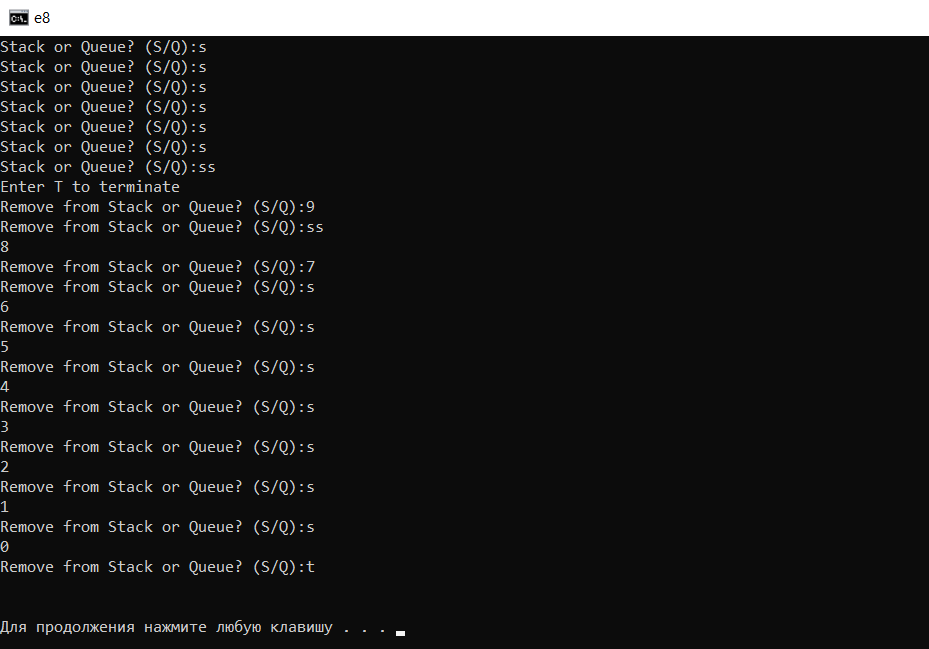
p = & obj2 ;

p-> fun();

// буде запускатися віртуальна функція класу base

}

Завдання 11.8



Код :

#include <iostream>

#include <cstdlib>

using namespace std ;

class list{

public :

list \*head ;

list \*tail;

list \*next;

int num ;

list(){ head = tail = next = NULL ;}

virtual void store (int i) = 0 ;

virtual int retrieve () = 0 ;

};

class queue : public list {

public :

void store(int i);

int retrieve();

};

void queue :: store(int i){

list \*item ;

item = new queue ;

if(!item){

cout << "Allocation error.\n";

exit(1);

}

item->num = i ;

if(tail) {tail->next = item ;}

tail = item ;

if(!head)head = tail ;

}

int queue :: retrieve(){

int i ;

if(!head){

cout << "list empty\n";

return 0;

}

i = head -> num ;

head = head -> next ;

return i ;

}

class stack : public list {

public :

void store(int i);

int retrieve();

};

void stack:: store(int i){

list \* item;

item = new stack ;

if(!item){

cout << "Allocation error\n";

exit(1);

}

item-> num = i;

if(head) item ->next = head ;

head = item ;

if(!tail) tail = head ;

}

int stack :: retrieve(){

int i ;

if(!head){

cout << "list empty\n";

return 0 ;

}

i = head ->num ;

head = head ->next ;

return i ;

}

int main() {

list \*p;

queue q\_ob;

stack s\_ob;

char ch;

for (int i=0; i<10; i++) {

cout << "Stack or Queue? (S/Q):";

cin >> ch;

ch = tolower(ch);

if (ch == 'q')

p = &q\_ob;

else

p = &s\_ob;

p->store(i);

}

cout << "Enter T to terminate\n";

for (;;) {

cout << "Remove from Stack or Queue? (S/Q):";

cin >> ch;

ch = tolower(ch);

if (ch == 't')

break;

if (ch == 'q')

p = &q\_ob;

else

p = &s\_ob;

cout << p->retrieve() << '\n';

}

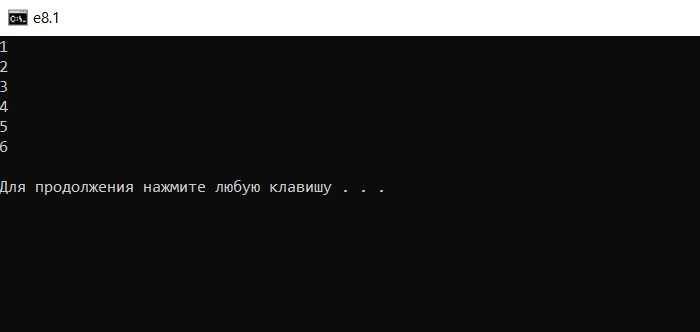
cout << '\n';

return 0;

}

//в мене помилок не було чи я зразу виправив

Завдання 11.8.1



Код :

#include <iostream>

#include <cstdlib>

using namespace std ;

class list{

public :

list \*head ;

list \*tail;

list \*next;

int num ;

list(){ head = tail = next = NULL ;}

virtual void store (int i) = 0 ;

virtual int retrieve () = 0 ;

};

class queue : public list {

public :

void store(int i);

int retrieve();

};

void queue :: store(int i){

list \*item ;

item = new queue ;

if(!item){

cout << "Allocation error.\n";

exit(1);

}

item-> num = i ;

if(!head) {

item->next = new queue ; // память потрібна для майбутніх swapiв

head = item ; // для першого елемента

}

else

{

list \* p = head;

int t ;

for ( ; p->next != NULL ; p = p->next ) {

if (item->num < p->num ) {

t = p->num ;

p->num = item->num ; // swapи

item->num = t ;

}

}

p->num = item->num ;

p->next = new queue ;

}

}

int queue :: retrieve(){

int i ;

if(!head){

cout << "list empty\n";

return 0;

}

i = head -> num ;

head = head -> next ;

return i ;

}

int main() {

list \*p;

queue ob;

p = & ob ;

p -> store(2);

p -> store(1);

p -> store(3);

p -> store(6);

p -> store(5);

p -> store(4);

cout << p-> retrieve() << endl ;

cout << p-> retrieve() << endl ;

cout << p-> retrieve() << endl ;

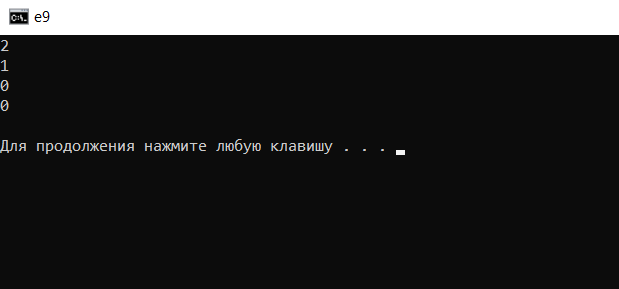
cout << p-> retrieve() << endl ;

cout << p-> retrieve() << endl ;

cout << p-> retrieve() << endl ;

}

Завдання 11.9



Код :

#include <iostream>

#include <cstdlib>

using namespace std ;

class list{

public :

list \*head ;

list \*tail;

list \*next;

int num ;

list(){ head = tail = next = NULL ;}

virtual void store (int i) = 0 ;

virtual int retrieve () = 0 ;

};

class queue : public list {

public :

void store(int i);

int retrieve();

void operator + (int i){

this -> store(i);

}

void operator - (int i){

for (; 0 < i; i--) retrieve();

}

};

void queue :: store(int i){

list \*item ;

item = new queue ;

if(!item){

cout << "Allocation error.\n";

exit(1);

}

item->num = i ;

if(tail) {tail->next = item ;}

tail = item ;

if(!head)head = tail ;

}

int queue :: retrieve(){

int i ;

if(!head){

return 0;

}

i = head -> num ;

head = head -> next ;

return i ;

}

class stack : public list {

public :

void store(int i);

int retrieve();

void operator + (int i){

this -> store(i);

}

void operator - (int i){

for (; 0 < i; i--) retrieve();

}

};

void stack:: store(int i){

list \* item;

item = new stack ;

if(!item){

cout << "Allocation error\n";

exit(1);

}

item-> num = i;

if(head) item ->next = head ;

head = item ;

if(!tail) tail = head ;

}

int stack :: retrieve(){

int i ;

if(!head){

return 0 ;

}

i = head ->num ;

head = head ->next ;

return i ;

}

int main() {

list \*p;

queue qob;

stack sob;

sob + 2 ;

p = & sob ;

cout << p->retrieve() << endl ;

qob + 1 ;

p = & qob ;

cout << p->retrieve() << endl ;

sob - 1 ;

p = & sob ;

cout << p->retrieve() << endl ;

qob - 1 ;

p = & qob ;

cout << p->retrieve() << endl ;

}