

ВВЕДЕНИЕ

Язык программирования C++ поддерживает *объектно-ориентированное программирование*. Объекты – это эффективные повторно используемые компоненты программного обеспечения, моделирующие элементы реального мира. Разработчики программного обеспечения обнаруживают, что использование достижений модульного объектно-ориентированного проектирования может значительно повысить продуктивность групп разработки по сравнению с такой предшествующей популярной технологией программирования, как структурное программирование. Объектно-ориентированные программы легче понимать, корректировать и модифицировать.

Лабораторные работы предлагается выполнять в среде разработки Visual Studio.NET, которая включена в платформу .NET. Платформа .NET включает не только среду разработки для нескольких языков программирования, называемую Visual Studio.NET, но и множество других средств, например, механизмы поддержки баз данных, электронной почты и коммерции [12].

В эпоху стремительного развития Интернета – глобальной информационной сети, объединяющей компьютеры разных архитектур, важнейшими задачами программ становятся:

- *переносимость* – возможность выполнения на различных типах компьютеров;
- *безопасность* – невозможность несанкционированных действий;
- *надежность* – способность выполнять необходимые функции в предопределенных условиях; выдерживать средний интервал между отказами;
- *использование готовых компонентов* – для ускорения разработки;
- *межязыковое взаимодействие* – возможность применять одновременно несколько языков программирования.

Платформа .NET позволяет успешно решать все эти задачи. Для обеспечения переносимости компиляторы, входящие в состав платформы, переводят программу не в машинные коды, а в промежуточный язык (Microsoft Intermediate Language, MSIL, или просто IL), который не содержит команд, зависящих от языка, операционной системы и типа компьютера. Программа на этом языке выполняется не самостоятельно, а под управлением системы, которая называется *общезыковой средой выполнения* (Common Language Runtime, CLR).

Среда CLR может быть реализована для любой операционной системы. При выполнении программы CLR вызывает так называемый JIT-компилятор, переводящий код с языка IL в машинные команды конкретного процессора, которые немедленно выполняются. JIT означает «just in time», что можно перевести как «вовремя», т.е. компилируются только те части, которые требуется выполнить в данный момент.

Компилятор в качестве результата его выполнения создает так называемую *сборку* – файл с расширением exe или dll, который содержит код на языке IL и метаданные. Метаданные представляют собой сведения об объектах, используемых в программе, а также сведения о самой сборке. Они позволяют организовать межязыковое взаимодействие, обеспечивают безопасность и облегчают *развертывание приложений*, т.е. установку программ на компьютеры пользователей.

Во время работы программы среда CLR следит за тем, чтобы выполнялись только разрешенные операции, осуществляет распределение и очистку памяти и обрабатывает возникающие ошибки. Это многократно повышает безопасность и надежность программ.

Платформа .NET содержит огромную *библиотеку классов*, которые можно использовать при программировании на любом языке .NET. Библиотека имеет несколько уровней. На самом нижнем уровне находятся *базовые классы среды*, которые используются при создании любой программы: классы ввода-вывода, обработки строк, управления безопасностью, графического интерфейса пользователя, хранения данных и прочие. Над этим слоем находится набор классов, позволяющий работать с базами данных и XML. Классы самого верхнего уровня поддерживают разработку распределенных приложений, а web-также и windows-приложений. Программа может использовать классы любого уровня.

Платформа .NET рассчитана на объектно-ориентированную технологию создания программ.

В целях закрепления учебного материала приводятся образцы программ, темы лабораторных работ и варианты заданий по каждой лабораторной работе.

Методический комплекс рассчитан на изучение основ языка C++ лицами с малым опытом программирования. Он составлен на основе лекций и лабораторных работ, которые читаются и проводятся на кафедре Информатики и вычислительной техники для аспирантов и магистрантов физических, математических и технических специальностей.

Рекомендуемая литература включает источники по программированию [1-14].

Лабораторная работа №1

РАЗРАБОТКА ПРОСТЕЙШИХ ПРОГРАММ НА С++

Для выполнения заданий необходимо запустить программу Microsoft Visual Studio 6, пользуясь меню Пуск – Программы - Microsoft Visual Studio 6 – Microsoft Visual C++. На экране появится окно Microsoft Visual C++, представленное на рисунке 1.1.

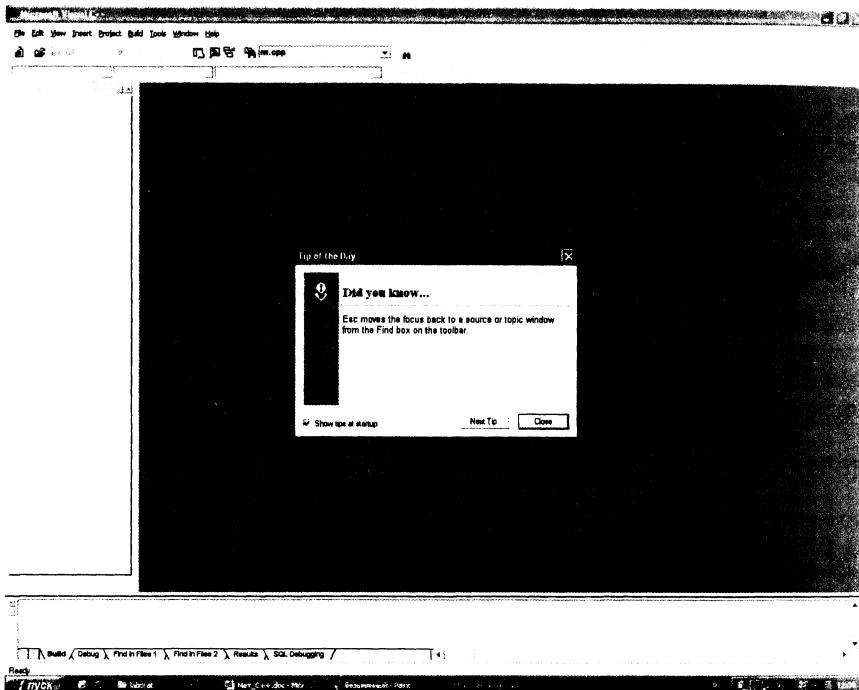


Рисунок 1.1 – Общий вид окна Microsoft Visual C++

Закрываем окно *Tip of the Day* (Close). Создаем новый файл: меню File – New. Набираем текст программы, потом сохраняем его с расширением .cpp: меню File – Save as. Файл сохраняем в вашу рабочую папку. Ключевые слова при этом будут выделенными синим цветом. Теперь откомпилируем программу при помощи меню Compile (синяя стрелка), исправляем ошибки компиляции. Если нет ошибок, запускаем программу на выполнение: меню Execute Program (красный восклицательный знак). Вводим с клавиатуры исходные данные (в качестве разделителя между переменными можно использовать клавишу пробел или клавишу Enter), получаем необходимые результаты выполнения программы.

Задание 1.1

Линейная программа, использование математических функций

Используемые математические функции находятся в стандартной библиотеке `<math.h>`, описание некоторых функций приведено в таблице 1.1.

Таблица 1.1 – Основные математические функции библиотеки math

Имя	Описание	Результат	Пояснения
abs	Модуль	int	$ x $, записывается как <code>abs(x)</code>
acos	Арккосинус	double	<code>acos(x)</code>
asin	Арксинус	double	<code>asin(x)</code>
atan	Арктангенс	double	<code>atan(x)</code>
cos	Косинус	double	<code>cos(x)</code>
exp	Экспонента	double	e^x , записывается как <code>exp(x)</code>
fabs	Модуль	double	$ x $, записывается как <code>fabs(x)</code>
log	Натуральный логарифм	double	$\log_e x$, записывается как <code>log(x)</code>
log10	Десятичный логарифм	double	$\log_{10} x$, записывается как <code>log10(x)</code>
pow	Возведение в степень	double	x^y , записывается как <code>pow(x,y)</code>
sin	Синус	double	<code>sin(x)</code>
sqrt	Квадратный корень	double	\sqrt{x} , записывается как <code>sqrt(x)</code>
tan	Тангенс	double	<code>tan(x)</code>

Используя математические функции, вычислить выражение:

$$f = \frac{\sqrt[4]{y + \sqrt[3]{x - 1}}}{|x - y| (\sin^2 z + \operatorname{tg} z)}$$

При $x = 17.42$; $y = 10.365 * 10^3$; $z = 0.828 * 10^5$ $f = 0.33054$

Для выполнения этого задания необходимо подключить библиотеку ввода/вывода `<iostream.h>` (эта библиотека содержит функции ввода с клавиатуры и функции вывода на экран) и библиотеку математических функций `<math.h>`.

```
#include <iostream.h> //1
#include <math.h> //2
int main () //3
{
    double f, y, x, z; //4
```

```

cout << "Enter x, y, z: \n";
cin >> x >> y >> z; //5
//6

// first method

double numerator = pow ( y + pow ( x - 1, 1./3.), 1./4.); //7
double denominator = fabs (x - y) * (pow (sin (z), 2) + tan (z)); //8
f = numerator / denominator; //9
cout << "First method\n" << "f = " << f << endl << endl; //10

// second method
f = ( pow ( y + pow ( x - 1, 1./3.), 1./4. ) / (fabs (x - y) * (pow (sin (z), 2) +
tan (z)) ); // 11
cout << "Second method\n" << "f = " << f << endl << endl; //12
return 0; //13
}

```

//1, //2 – любая директива начинается со знака # (решетка), затем пишется название директивы include (включить), после этого в угловых скобках пишется название библиотеки, которую необходимо подключить.

//3 – C++ начинает работу с функции main(), затем в фигурных скобках пишутся инструкции, которые должна выполнить main() функция.

//4 – объявление переменных, тип входных и выходных данных вещественный с двойной точностью, переменные не инициализируются.

//5 – вывод на экран сообщения: Enter x, y, z: и перевод курсора экрана на новую строку с помощью Esc-последовательности (\n).

//6 – ввод входных данных (x, y, z) с клавиатуры.

//7 – объявляем переменную, числителю (numerator) которой будет присваиваться значение правой части инструкции, вычисленное по формуле, используемой в задании.

//8 – объявляем переменную, знаменателю (denominator) которой будет присваиваться значение правой части инструкции, вычисленное по формуле, применяемой в задании.

//9 – вычисляем результат, который необходимо присвоить переменной f.

//10 – выводим на экран сообщение о том, что расчет математического выражения выполнен первым способом, на следующей строке выводим на экран имя переменной и ее значение.

//11 – вычисляем результат без промежуточных переменных (numerator, denominator), при этом ответ в двух способах получается один и тот же, так как данные вводятся один раз, и для одного математического выражения.

//12 – выводим на экран сообщение о том, что выполнен расчет математического выражения вторым способом; на следующей строке выводим на экран имя переменной и ее значение.

//13 – поскольку функция main() была определена как int, то она должна вернуть целочисленное значение (в нашем случае 0 – стандартный код успешного завершения программы). Закрытая фигурная скобка, показывает, что программа больше не будет содержать никаких инструкций.

Результаты выполнения программы показаны на рисунке 1.2.

```

Enter x, y, z:
17.421 0.010365 82800
First method
f = 0.330564

Second method
f = 0.330564

Press any key to continue

```

Рисунок 1.2 – Результаты выполнения программы задания 1.1

Индивидуальные задания

Используя математические функции, вычислить выражение:

$$1. \quad \psi = \frac{e^{|x-y|} |x-y|^{x+y}}{\operatorname{arctg}(x) + \operatorname{arctg}(z)} + \sqrt[3]{x^6 + \ln^2 y}.$$

При $x = -2.235 \cdot 10^{-2}$; $y = 2.23$; $z = 15.221$ $\psi = 39.374$.

$$2. \quad b = y^{\sqrt[3]{x}} + \cos^3(y) \frac{|x-y| \left(1 + \frac{\sin^2 z}{\sqrt{x+y}} \right)}{e^{|x-y|} + \frac{x}{2}}.$$

При $x = 6.251$; $y = 0.827$; $z = 25.001$ $b = 0.7121$.

$$3. \quad \psi = \left| x^{\frac{y}{x}} - \sqrt[3]{\frac{y}{x}} \right| + (y-x) \frac{\cos y - \frac{z}{(y-x)}}{1+(y-x)^2}.$$

При $x = 1.825 \cdot 10^2$; $y = 18.225$; $z = -3.298 \cdot 10^{-2}$ $\psi = 1.2131$.

$$4. \quad h = \frac{x^{y+1} + e^{y-1}}{1+x|y-\operatorname{tg} z|} (1+|y-x|) + \frac{|y-x|^2}{2} - \frac{|y-x|^3}{3}.$$

При $x = 2.444$; $y = 0.869 \cdot 10^{-2}$; $z = -0.13 \cdot 10^3$ $h = -0.49871$.

$$5. \quad v = \frac{1+\sin^2(x+y)}{\left| x - \frac{2y}{1+x^2 y^2} \right|} x^{|y|} + \cos^2 \left(\operatorname{arctg} \frac{1}{z} \right).$$

При $x = 3.74 \cdot 10^{-2}$; $y = -0.825$; $z = 0.16 \cdot 10^2$ $v = 1.0553$.

$$6. \quad g = \frac{y^{x+1}}{\sqrt[3]{|y-2|} + 3} + \frac{x+\frac{y}{2}}{2|x+y|} (x+1)^{-\frac{1}{\sin z}}.$$

При $x = 12.3 \cdot 10^{-1}$; $y = 15.4$; $z = 0.252 \cdot 10^3$ $g = 82.8257$.

$$7. \quad u = \frac{\sqrt[3]{8 + |x-y|^2 + 1}}{x^2 + y^2 + 2} - e^{|x-y|} (\operatorname{tg} z + 1)^x.$$

При $x = -4.5$; $y = 0.75 \cdot 10^{-4}$; $z = 0.845 \cdot 10^2$ $u = -55.6848$.

$$8. \quad w = |\cos x - \cos y|^{(1+2 \sin^2 y)} \left(1 + z + \frac{z^2}{2} + \frac{z^3}{3} + \frac{z^4}{4} \right).$$

При $x = 0.4 \cdot 10^4$; $y = -0.875$; $z = -0.475 \cdot 10^{-3}$ $w = 1.9873$.

$$9. \quad a = \ln \left(y^{\sqrt{|z|}} \left(x - \frac{y}{2} \right) + \sin^2 \operatorname{arctg}(z) \right).$$

При $x = -15.246$; $y = 4.642 \cdot 10^{-3}$; $z = 20.001 \cdot 10^2$ $a = -182.036$.

$$10. \quad a = 2^{-x} \sqrt{x + \sqrt[4]{|y|}} \sqrt[3]{e^x - 1 / \sin z}.$$

При $x = 3.981 \cdot 10^2$; $y = -1.625 \cdot 10^3$; $z = 0.512$ $a = 1.26185$.

11. Определить время окончания рабочего дня (в часах и минутах), если известны время его начала (в часах и минутах) и продолжительность (вместе с обедом) (в часах и минутах).

12. Перевести белорусское время (в часах) в московское. (Учесть, что 23 часа по белорусскому времени – это 0 часов по московскому).

13. Определить h – полное количество часов и m – полное количество минут, прошедших от начала суток до того момента (в первой половине дня), когда часовая стрелка повернулась на f градусов ($0 <= f < 360$; f – вещественное число).

14. По номеру n ($n > 0$) некоторого года определить c – номер его столетия (учесть, что, к примеру, началом XX столетия был 1901, а не 1900 год).

15. Присвоить целой переменной k первую цифру из дробной части положительного вещественного числа.

Задание 1.2

Условный оператор if, оператор ?:

Вычислить заданное условное вещественное выражение:

$$X = \begin{cases} b/a - 5, & a > b, \\ 22, & a = b, \\ (a-9)/b, & a < b \end{cases}$$

Для выполнения данного задания необходимо подключить библиотеку ввода/вывода <iostream.h>.

```
#include <iostream.h>
int main()
{
    double b, a, X; //1
    cout << "Enter a, b: \n "; //2
    cin >> a >> b; //3
    if (a>b)
        X = b/a - 5; //4a
    else
        if (a == b)
            X = 22; //4b
        else
            X = (a - 9)/ b; //4
    cout << "X = " << X << endl; //5
    X = a > b ? b/a - 5 : a == b ? 22 : (a - 9)/ b; //6
    cout << "X = " << X << endl; //7
    return 0;
}
```

//1 – объявление переменных, тип входных и выходных данных вещественный с двойной точностью; переменные не инициализируются.

//2 – вывод на экран сообщения: Enter a, b: и перевод курсора экрана на новую строку с помощью Esc-последовательности (\n).

//3 – ввод входных данных (a, b) с клавиатуры.

//4 – использование инструкции if для вычисления переменной X (в круглых скобках – логическое выражение, если оно истинно, то вычисляется).

//4a – если оно ложно, то проверяется логическое выражение в скобках (если оно истинно, то выполняется выражение //4b, если оно ложно, то выполняется выражение //4).

//5 – вывод результата на экран в виде X = значение вычисленного выражения в инструкции if, перевод курсора на новую строку экрана (endl).

//6 – оператор ? для вычисления переменной X: проверяется логическое выражение a > b, если оно истинно, то вычисляется выражение после знака ? и X присваивается результат вычислений, если оно ложно, проверяется логическое выражение a == b: если оно истинно, то вычисляется выражение после знака ? и X присваивается результат вычислений, если оно ложно, то вычисляется выражение, стоящее за знаком (:) и X присваивается результат вычислений.

//7 – вывод результата на экран в виде X = значение вычисленного выражения в инструкции ?, перевод курсора на новую строку экрана (endl). Значение вычисленного выражения инструкции if и оператора ? должны совпадать, ибо независимо от того, каким способом находится значение X, результат будет один и тот же.

Результаты выполнения программы представлены на рисунке 1.3.

```

Enter a, b:
23 28
x = 0.129487
x = 0.129487
Press any key to continue

```

Рисунок 1.3 – Результаты выполнения программы задания 1.2

Индивидуальные задания

Вычислить заданное условное вещественное выражение:

1.

$$X = \begin{cases} b/a + 5, & a < b \\ -5, & a = b \\ (a^*a - b)/b, & a > b \end{cases}$$

2.

$$X = \begin{cases} (a - b)/a + 1, & a > b \\ 25, & a = b \\ (a - 5)/b, & a < b \end{cases}$$

3.

$$X = \begin{cases} a/b - 1, & a > b \\ -25, & a = b \\ (b^3 - 5)/a, & a < b \end{cases}$$

4.

$$X = \begin{cases} 52 * b / a + b, & a > b \\ -125, & a = b \\ (a - 5) / b, & a < b \end{cases}$$

5.

$$X = \begin{cases} 1 - b / a, & a > b \\ -10, & a = b \\ (a - 5) / b, & a < b \end{cases}$$

6.

$$X = \begin{cases} (2+b)/a, & a > b \\ -2, & a = b \\ (a-5)/b, & a < b \end{cases}$$

7.

$$X = \begin{cases} b/a + 61, & a > b \\ -5, & a = b \\ (b-a)/b, & a < b \end{cases}$$

8.

$$X = \begin{cases} (3*a-5)/b, & a < b \\ -4, & a = b \\ (a^3 + b)/a, & a > b \end{cases}$$

9.

$$X = \begin{cases} 2*a/b + 1, & a > b \\ -445, & a = b \\ (b+5)/a, & a < b \end{cases}$$

10.

$$X = \begin{cases} b/a + 10, & a > b \\ 3425, & a = b \\ (2*a-5)/b, & a < b \end{cases}$$

11. Напишите программу, которая считывает два целых числа, определяет и печатает, является ли первое число кратным второму. (Используйте операцию вычисления остатка).

12. Напишите программу, которая считывает целое число, определяет и печатает, четное ли оно или нечетное. (Используйте операцию вычисления остатка).

13. Напишите программу, которая вводит число из пяти цифр, разделяет число на отдельные цифры и печатает их отдельно с тремя пробелами между ними.

14. Значения переменных a , b и c поменять местами так, чтобы оказалось $a>=b>=c$.

15. Смоделируйте игру "крепс". Правила игры:

Игрок бросает две кости. Каждая кость имеет шесть граней, помеченных цифрами как 1, 2, 3, 4, 5, 6. После броска вычисляется сумма цифр двух верхних граней. Если сумма после первого броска равна 7 или 11, игрок выиграл. Если после первого броска сумма равна 2, 3 или 12 (это называется "крепс"), игрок проигрывает (т. е. выигрывает "банк"). Если после первого броска сумма равна 4, 5, 6, 8, 9 или 10, то эта сумма становится "очками" игрока. Чтобы выиграть, игрок должен продолжить бросать кости до тех пор, пока не выпадет

сумма, равная его очкам. Игрок проигрывает, если во время этих бросков выпадет сумма 7.

Задание 1.3

Операторы цикла for, while, do..while

Вычислить и вывести на экран в виде таблицы значения функции, заданной с помощью ряда Тейлора, на интервале от $x_{\text{нач}}$ до $x_{\text{кон}}$ с шагом dx с точностью eps>0. Таблицу снабдить заголовком и шапкой. Каждая строка таблицы должна содержать значение аргумента, значение функции и количество просуммированных членов ряда:

$$\arctg x = x - x^3/3 + x^5/5 - \dots + (-1)^n x^{2n+1}/(2n+1) + \dots$$

$$(|x| < 1)$$

Для выполнения этого задания необходимо подключить библиотеку ввода/вывода `<iostream.h>` и библиотеку математических функций `<math.h>`.

```
#include <iostream.h>
#include <math.h>
int main()
{
    double arctg, y0, numerator, denominator, sign, x, xinitial, xfinally, dx, eps,
    delta, count; //1
    cout << "Enter xinitial, xfinally, dx, eps: \n"; //2
    cin >> xinitial >> xfinally >> dx >> eps; //3
    cout << " X\t ARCTG\t COUNT\n\n"; //4
    for (x = xinitial; x <= xfinally; x+=dx) //5
    {
        y0 = x; //6
        numerator = x; //7
        denominator = 3; //8
        sign = -1; //9
        count = 1; //10
        do{ //11
            numerator *= x*x; //12
            arctg = y0 + sign* numerator / denominator; //13
            sign *= -1; //14
            delta = fabs (y0 - arctg); //15
            y0 = arctg; //16
            denominator += 2; //17
            count++; //18
        } while (delta > eps); //19
        cout << x << "\t" << arctg << "\t" << count << endl; //20
    } //21
    return 0;
}

//1 – объявление переменных: arctg (результат: сумма ряда Тейлора), y0 (начальное значение для результата), numerator (числитель, слагаемого ряда Тейлора), denominator (знаменатель, слагаемого ряда Тейлора), sign (знак + или -), x (переменная цикла for), xinitial (начальное значение аргумента, входная пе-
```

ременная), xfinally (конечное значение аргумента, входная переменная), dx (шаг, с которым изменяется аргумент x, входная переменная), eps (точность вычисления arctg, входная переменная), delta (модуль разности y0 и arctg), count (количество слагаемых в цикле, для достижения точности eps).

//2 – вывод текста на экран “Enter xinitial, xfinally, dx, eps:”, чтобы при запуске программы не оказаться перед пустым экраном и вспоминать, что вам надо сделать, \n – Esc-последовательность, для перевода курсора экрана на новую строку.

//3 – ввод исходных данных xinitial, xfinally, dx, eps с клавиатуры.

//4 – заголовок таблицы: значение аргумента, значение функции, количество просуммированных членов ряда.

//5 – инструкция for. В круглых скобках – инициализация переменной цикла x значением xinitial; после точки с запятой следует логическое выражение: до какого значения должен работать цикл for (xfinally); после точки с запятой следует выражение (изменение переменной цикла x на шаг dx). После закрытой круглой скобки точка с запятой не ставится. Так как цикл for состоит не из одной инструкции цикла, а из нескольких, необходимо открыть фигурную скобку, чтобы выделить для компилятора все инструкции, входящие в цикл.

//6 – присваиваем переменной y0 начальное значение: для данного задания это будет первое слагаемое ряда Тейлора.

//7 – присваиваем переменной numerator – начальное значение для вычисления степени числителя. Для данного задания это будет переменная цикла x.

//8 – присваиваем начальное значение для переменной denominator. Для данного задания это будет значение 3.

//9 – начальное значение для переменной sign будет знак -, так как в данном задании после y0 знак меняется на минус.

//10 – начальное значение переменной count будет 1, так как при вычислении arctg в выражении присутствуют два слагаемых: y0 и sign * numerator / denominator.

//11 – открываем цикл do{}, чтобы вычислить требуемую точность значения функции. Для разных аргументов ряда нужно разное количество слагаемых (count), поэтому вычисления будут проводиться в цикле do-while.

//12 – вычисляем числитель (numerator); на первом шаге он должен быть x^5 . На следующем шаге он будет x^5 и т. д..

//13 – вычисляем значение функции (arctg).

//14 – меняем знак (sign) на +. При следующей итерации он опять станет отрицательным.

//15 – вычисляем модуль разности y0 и arctg.

//16 – на следующей итерации цикла значение y0 будет иметь значение arctg.

//17 – меняем знаменатель (denominator), для следующего слагаемого он должен быть равным 5. При следующей итерации цикла он станет равен 7 и т. д..

//18 – к значению функции добавляется слагаемое (sign* numerator / denominator), поэтому к переменной count должна быть добавлена единица (count++).

//19 – проверяем логическое выражение – достигли ли мы заданной точности eps: если да, выводим из цикла do-while, если нет, остаемся в цикле для следующей итерации.

//20 – выводим на экран значение аргумента (x), значение функции (arctg), количество просуммированных членов ряда (count).

//21 – закрываем цикл for (). В круглых скобках заголовка цикла отражается изменение переменной цикла x на шаг dx; затем проверяется логическое выражение. Если оно истинно, мы остаемся в цикле for, если ложно, выводим из цикла.

Результаты выполнения программы представлены на рисунке 1.4.

X	ARCTG	COUNT
0.2	0.197396	5
0.3	0.291457	7
0.4	0.380906	9
0.5	0.463648	11
0.6	0.540419	15
0.7	0.610726	21
0.8	0.674241	32
0.9	0.732815	64

Рисунок 1.4 – Результаты выполнения программы задания 1.3

Индивидуальные задания

Вычислить и вывести на экран в виде таблицы значения функции, заданной с помощью ряда Тейлора, на интервале от $x_{\text{нач}}$ до $x_{\text{кон}}$ с шагом dx , с точностью $\text{eps} > 0$. Таблицу снабдить заголовком и шапкой. Каждая строка таблицы должна содержать значение аргумента, значение функции и количество просуммированных членов ряда.

1. $\text{arth } x = x + x^3/3! + x^5/5! + \dots + x^{2n+1}/(2n+1)! + \dots$

$x > 1$

2. $\sin x = x - x^2/2 + x^3/3 - \dots + (-1)^{n-1} x^n/n + \dots$

$(|x| < 1)$

3. $\cos x = 1 - x^2/2! + x^4/4! + \dots + x^{2n}/(2n)! + \dots$

$|x| < \infty$

4. $\ln \frac{x+1}{x-1} = 2 (1/x + 1/3x^3 + 1/5x^5 + \dots + 2(1/(2n+1)x^{2n+1}) + \dots)$

$|x| > 1$

$$5. e^{-x} = 1 - x + x^2/2! - x^3/3! + x^4/4! - \dots (-1)^n x^n/n! + \dots$$

$|x| < \infty$

$$6. \ln(x+1) = x - x^2/2 + x^3/3 - x^4/4 + \dots (-1)^n x^{n+1}/(n+1) + \dots$$

$-1 < x \leq 1$

$$7. \ln(1-x) = -(x + x^2/2 + x^4/4 + \dots - x^n/n + \dots)$$

$-1 \leq x < 1$

$$8. \operatorname{arcctg} x = \pi/2 - x + x^3/3 - x^5/5 + \dots + (-1)^{n+1} x^{2n+1}/(2n+1) + \dots$$

$|x| < 1$

$$9. \operatorname{arctg} x = \pi/2 - 1/x + 1/3x^3 - 1/5x^5 + \dots + (-1)^{n+1} / (2n+1)x^{2n+1} + \dots$$

$x > 1$

$$10. \operatorname{arctg} x = -\pi/2 - 1/x + 1/3x^3 - 1/5x^5 + \dots (-\pi/2 + (-1)^{n+1} / (2n+1)x^{2n+1}) + \dots$$

$x < -1$

11. Приближенно вычислить интеграл

$$\int_0^\pi \ln(2 + \sin x) dx,$$

используя формулу прямоугольников при $n=100$:

$$\int f(x) dx \approx h * [f(x_1) + f(x_2) + \dots + f(x_n)],$$

где $h = (b-a)/n$, $x_i = a + ih - h/2$.

12. Числа Фибоначчи определяются формулами

$$f_0 = f_1 = 1; f_n = f_{n-1} + f_{n-2} \text{ при } n = 2, 3, \dots$$

определить 40-е число Фибоначчи, найти первое число Фибоначчи, большее m ($m > 2$).

13. Дано целое число n :

- 1) Сколько цифр в числе n ?
- 2) Чему равна их сумма?
- 3) Найти первую цифру числа n .

14. Дано целое число n :

- 1) Выяснить входит ли цифра 3 в запись числа n^2 .
- 2) Поменять порядок цифр числа n на обратный.
- 3) Переставить первую и последнюю цифры числа n .
- 4) Присписать по единице в начало и в конец записи числа n .

15. Напишите программу, моделирующую бросание двух костей. Программа должна использовать `rand()` для бросания первой кости и затем – снова `rand()` для метания второй кости. Затем нужно подсчитать сумму двух значений. Примечание: поскольку каждая кость может показать целое значение от 1 до 6,

то сумма двух чисел может варьироваться от 2 до 12 с наиболее частым значением суммы 7 и наименее частыми значениями 2 и 12. Ваша программа должна выбросить две кости 36 тыс. раз. Используйте одномерный массив, чтобы подсчитывать, сколько раз выпадает каждая возможная сумма. Напишите программу, которая выполняет тысячу игр в крепс и отвечает на следующие вопросы:

- 1) Сколько игр выиграно при первом бросании, при втором ... при двадцатом, после двадцатого бросания?
- 2) Сколько игр проиграно при первом бросании, при втором ... при двадцатом, после двадцатого бросания?
- 3) Каковы шансы на выигрыш в крепс?
- 4) Какова средняя продолжительность игры в крепс?
- 5) Растут ли шансы выигрыша с увеличением продолжительности игры?

Задание 1.4

Функции пользователя.

Напишите функции, которые вычисляют соответственно минимум и максимум из трех чисел. Вычислите значение выражения:

$$m = \frac{\max(f(x), y, z)}{\min(f(x), y)} + 5, \text{ где функция } f(x) = x^2.$$

Для выполнения этого задания необходимо подключить библиотеку ввода/вывода `<iostream.h>`.

```
#include <iostream.h>
int max (int fx, int y, int z); //1
int min (int fx, int y); //2
int main()
{
    int number1, number2, number3, result; //3
    cout << "Enter number1, number2, number3: \n"; //4
    cin >> number1 >> number2 >> number3; //5
    result = max (number1* number1, number2, number3) / min (number1 *
    number1, number2) + 5; //6
    cout << "result = " << result << endl; //7
    return 0;
}
int max (int fx, int y, int z) //8
{
    if (fx > y && fx > z) //9
        return fx; //10
    else if (y > fx && y>z) //11
        return y; //12
    else //13
        return z; //14
}
int min (int fx, int y) //15
{
    if (fx < y) //16
```

```
return fx;           //17
else                //18
    return y;        //19
}
```

//1 – прототип функции, обязательно заканчивающийся точкой с запятой. Тип возвращаемого значения *int* (целочисленный), имя функции – *max*; функция имеет три аргумента типа *int* (*fx*, *y*, *z*). Это формальные параметры: при вызове функции вместо них будут подставлены реальные аргументы.

//2 – прототип функции: тип возвращаемого аргумента *int*, имя функции *min*, имеет два целочисленных аргумента (*fx*, *y*).

//3 – объявление целочисленных переменных – три входных переменных, четвертая будет содержать результат вычислений.

//4 – вывод текста на экран (пометка, что должен делать программист при запуске программы на выполнение).

//5 – ввод исходных данных с клавиатуры.

//6 – вызов кодов функций *min()* и *max()*: формальные параметры заменяются реальными, причем соответствие типов обязательно вычисляется выражением из постановки задачи и присваивается переменной результата (*result*).

//7 – вывод результата на экран.

//8 – начало описания определения функции *max()*. Заголовок функции он полностью повторяет описание прототипа функции, только в нем отсутствует точка с запятой.

//9 – инструкция *if* и логическое выражение: определяется больше ли значение первого аргумента *fx* значений аргументов *у* и *z*. Если оно истинно, функция вернет значение аргумента *fx* (//10), если ложно (//11), происходит проверка двух оставшихся аргументов на большее значение; если оно истинно функция вернет значение аргумента *у* (//12), если ложно (//13) вернет значение аргумента *z* (//14).

//15 – заголовок функции *min()* повторяет прототип функции *min()* без точки с запятой.

//16 - инструкция *if* и логическое выражение: определяется меньше ли значение первого аргумента *fx* значения аргумента *у*, если оно истинно, функции вернет значение *fx* (//17), если ложно (//18), функция вернет значение аргумента *у* (//19).

Результаты выполнения программы представлены на рисунке 1.5.

Индивидуальные задания

1. Напишите функцию, которая воспринимает целое значение и возвращает число с обратным порядком цифр. Например, воспринимается число 7361, а возвращается число 1637.
2. Напишите функцию, определяющую, является ли число простым или нет. Используйте эту функцию в программе, которая определяет и печатает все простые числа в диапазоне от 1 до 1000. Сколько этих чисел вы должны действительно проверить?

```

ex14.c
Enter number1, number2, number3:
45 37 89
result = 59
Press any key to continue...

```

Рисунок 1.5 – Результаты выполнения программы задания 1.4

3. Даны действительные числа s, t . Получить:
 $f(t, -2s, 1.17) + f(2.2, t, s - t)$,
 где
 $f(a, b, c) = (2a - b - \sin c) / (5 + |c|)$.
4. Даны действительные числа s, t . Получить:
 $g(1.2, s) + g(t, s) - g(2s - 1, st)$,
 где
 $g(a, b) = (a^2 + b^2) / (a^2 + 2ab + 3b^2 + 4)$.
5. Дано действительное число y . Получить:
 $(1.7t(0.25) + 2t(1 + y)) / (6 - t(y^2 - 1))$,
 где

$$t(x) = \frac{\sum_{k=1}^n (x^{2k+1}) / (2k+1)!}{\sum_{k=2}^{n-1} x^{2k} / (2k)!}$$
.
6. Даны действительные числа a, b, c . Получить:
 $(\max(a, a+b) + \max(a, b+c)) / (1 + \max(a+b*c, 1.15))$.
7. Даны действительные числа a, b . Получить:
 $u = \min(a, b), v = \min(a*b, a+b), \min(u + v^2, 3.14)$.
8. Даны действительные числа s, t . Получить:
 $h(s, t) + \max(h^2(s-t, s+t), h^4(s-t, s+t)) + h(1, 1)$,
 где
 $h(a, b) = a / (1 + b^2) + b / (1 + a^2) - (a - b)^2$.
9. Напишите программу, использующую функцию, которая вычисляет стоимость поездки на автомобиле на дачу (туда и обратно). Параметрами функции являются: расстояние до дачи (км), количество бензина, которое потребляет автомобиль на 100 км пробега, цена одного литра бензина. Рекомендуемый вид экрана:

Расстояние до дачи (км.)=

Расход бензина (литров на 100 км пробега) =

Цена литра бензина (руб.) =

Поездка на дачу туда и обратно обойдется в ... руб.

10. Напишите функцию, которая переводит время из часов и минут в секунды. Входными данными являются три целых аргумента (часы, минуты, секунды), а результатом: число секунд с момента, когда часы "пробили" 12. Используйте эту функцию для вычисления промежутка времени в секундах между двумя моментами, находящимися внутри 12-часового круга. Программа должна проверять правильность, введенных пользователем, данных. В случае если данные введены неверно, надо выдавать соответствующее сообщение.

11. Напишите программу вычисления стоимости покупки с учетом скидки. Скидка в 3% предоставляется, если сумма покупки больше 5 тыс. руб., 5% - если сумма больше 10 тыс. руб. Рекомендуемый вид экрана:

Ведите сумму покупки: = ...

Вам предоставляется скидка = ... %

Сумма с учетом скидки составит: ... руб.

12. Напишите функции, которые вычисляют соответственно минимум и максимум из двух чисел. Вычислите значение выражения:

$$n = \frac{\min(f(x) + y, y - z)}{\max(f(x), y)}, \text{ где функция } f(x) = e^x.$$

13. Напишите программу, использующую функцию вычисления стоимости разговора по телефону. Входными параметрами функции являются: название и код города, цена минуты.

Город	Код	Цена минуты (в руб.)
Владивосток	423	2.2
Москва	095	1.0
Мурманск	815	1.2

Результатом работы будет стоимость разговора. Рекомендуемый вид экрана:

Ведите длительность разговора (целое число минут) =

Код города:

Стоимость разговора : ... руб.

14. Напишите программу, использующую функцию, которая вычисляет доход по вкладу. Параметрами функции являются величина вклада, процентная

ставка и срок вклада. Процентная ставка (% годовых) и время хранения (дней) задаются во время работы программы. Рекомендуемый вид экрана:

Ведите исходные данные:

Величина вклада (руб.) =

Срок вклада (дней)=

Процентная ставка (годовых)

Доход ...руб.

Сумма по окончании срока вклада...руб.

15. Напишите функцию `multiple` для двух целых, которая определяет, кратно ли второе число первому. Функция должна получать два целых аргумента и возвращать 1 (true), если второе число кратно первому и 0 (false) в противном случае. Используйте эту функцию в программе, которая вводит несколько пар целых чисел. Количество пар пользователь задает с клавиатуры в основной программе.

Лабораторная работа №2

СТРУКТУРНЫЕ ТИПЫ ДАННЫХ

Задание 2.1

Одномерные массивы

Даны целые числа a_0, \dots . Известно, что $a_i > 0$ и что среди a_2, a_3, \dots есть хотя бы одно отрицательное число. Пусть a_0, \dots, a_{n-1} – члены данной последовательности, предшествующие первому отрицательному члену (n заранее не известно). Получить:

$$\max(a_0, a_0a_1, \dots, a_0a_1\dots a_{n-1}).$$

```
#include <iostream.h>
int max(int a[], int n);
int main()
{
    int *mas, i=0, m;
    mas = new int [50];           //1
    cout << "Enter elements of vector:\n";
    do {
        cin >> mas [i];
        i++;
    } while (mas [i-1] > 0);      //2
    m = max (mas, i-1);          //3
    cout << "\n\nNew vector:\n\n";
    for (int j =0; j < i-1; j++)
        cout << mas[j] << " ";
    cout << "\n\nMaximum = " << m << endl;
    cout << "Maximum = " << m << endl;
    delete [] mas;               //4
    return 0;
}
int max(int a[], int n)
{
    int m, i;
    m = a[0];                   //5
    for ( i = 1; i < n; i++)
    {
        a[i] = a[i] * a[i-1];   //6
        if ( m < a[i])
            m = a[i];           //7
    }
    return m;                   //8
}
```

//1 – выделяем память под 50 элементов целого типа в heap области.

//2 – формируем массив mas, только из положительных целых чисел.

//3 – вызываем функцию max, в качестве первого аргумента передаем адрес массива mas, в качестве второго аргумента – число элементов массива; результат присваиваем переменной m.

//4 – возвращаем выделенную память в heap область.

//5 – задаем начальное значение переменной, которая будет возвращать максима.

//6 – нам необходимо найти максимальное число произведений элементов массива a , которые оператор вычисляет данный оператор.

//7 – с помощью инструкции if находим максимальное произведение.

//8 – возвращаемся в точку вызова функции и передаем максимальное значение из функции переменной m .

Результаты выполнения программы представлены на рисунке 2.1.

```
51 Enter elements vector's:  
2 3 4 5 7 8 6 3 2 1 -9  
New vector:  
2 6 24 120 840 6720 40320 120960 241920 241920  
Maximum = 241920  
Press any key to continue...
```

Рисунок 2.1 – Результаты выполнения программы задания 2.1

Индивидуальные задания

1. Даны целые числа a_0, a_1, \dots . Известно, что $a_i > 0$ и что среди a_2, a_3, \dots есть хотя бы одно отрицательное число. Пусть a_0, \dots, a_{n-1} – члены данной последовательности, предшествующие первому отрицательному члену (п заранее не известно). Получить:

$$\max(a_0^2, \dots, a_{n-1}^2).$$

2. Даны целые числа a_0, a_1, \dots . Известно, что $a_i > 0$ и что среди a_2, a_3, \dots есть хотя бы одно отрицательное число. Пусть a_0, \dots, a_{n-1} – члены данной последовательности, предшествующие первому отрицательному члену (п заранее не известно). Получить:

$$\max(a_0^3, \dots, a_{n-1}^3).$$

3. Даны целые числа a_0, a_1, \dots . Известно, что $a_i > 0$ и что среди a_2, a_3, \dots есть хотя бы одно отрицательное число. Пусть a_0, \dots, a_{n-1} – члены данной последовательности, предшествующие первому отрицательному члену (п заранее не известно). Получить:

$$\min(a_0, 2a_1, \dots, na_{n-1}).$$

4. Даны целые числа a_0, a_1, \dots . Известно, что $a_i > 0$ и что среди a_2, a_3, \dots есть хотя бы одно отрицательное число. Пусть a_0, \dots, a_{n-1} – члены данной последовательности, предшествующие первому отрицательному члену (n заранее не известно). Получить:

$$\min(a_0 + a_1, a_1 + a_2, \dots, a_{n-2} + a_{n-1}).$$

5. Даны целые числа a_0, a_1, \dots . Известно, что $a_i > 0$ и что среди a_2, a_3, \dots есть хотя бы одно отрицательное число. Пусть a_0, \dots, a_{n-1} – члены данной последовательности, предшествующие первому отрицательному члену (n заранее не известно). Поколичеству четных среди a_0, \dots, a_{n-1} .

6. Даны целые числа a_0, a_1, \dots . Известно, что $a_i > 0$ и что среди a_2, a_3, \dots есть хотя бы одно отрицательное число. Пусть a_0, \dots, a_{n-1} – члены данной последовательности, предшествующие первому отрицательному члену (n заранее не известно). Получить:

количество удвоенных нечетных среди a_0, \dots, a_{n-1} .

7. Даны целые числа a_0, a_1, \dots . Известно, что $a_i > 0$ и что среди a_2, a_3, \dots есть хотя бы одно отрицательное число. Пусть a_0, \dots, a_{n-1} – члены данной последовательности, предшествующие первому отрицательному члену (n заранее не известно). Получить:

$$\max(a_0^2, \dots, a_{n-1}^2).$$

8. Даны целые числа a_0, a_1, \dots . Известно, что $a_i > 0$ и что среди a_2, a_3, \dots есть хотя бы одно отрицательное число. Пусть a_0, \dots, a_{n-1} – члены данной последовательности, предшествующие первому отрицательному члену (n заранее не известно). Получить:

$$\max(a_0^3, \dots, a_{n-1}^3).$$

9. Даны целые числа x_0, x_1, \dots, x_{54} . Вычислить величину:

$$x_0(x_1 + x_2)(x_3 + x_4 + x_5)(x_6 + x_7 + x_8 + x_9) \dots (x_{45} + x_{46} + \dots + x_{54})$$

10. Данна последовательность из 100 целых чисел. Определить количество чисел в наиболее длинной последовательности из подряд идущих нулей.

11. По заданным вещественным числам $a_0, a_1, \dots, a_{30}, b_0, b_1, \dots, b_{30}, c_0, c_1, \dots, c_{30}, x, y, z$ вычислить величину
 $((a_0x^{30} + a_1x^{29} + \dots + a_{30}) - (b_0y^{30} + b_1y^{29} + \dots + b_{30})) / (c_0(x+z)^{30} + c_1(x+z)^{29} + \dots + c_{30})$

12. По заданным 20-элементным целым массивам x и y вычислить:

$$u = \begin{cases} \sum_{i=0}^{20} x_i^2, \sum_{i=0}^{15} x_i y_i \\ \sum_{i=0}^{20} y_i^2 \end{cases}$$

13. По заданным 50-элементным вещественным массивам a, b, c вычислить:

$$t = \begin{cases} \min(b_i) / \max(a_i) + \max(c_i) / \min(b_i + c_i), \min(a_i) < \max(b_i) \\ \max(b_i + c_i) + \min(c_i) \end{cases}$$

14. По заданным 40-элементным вещественным векторам x, y, z вычислить:

$$w = \begin{cases} \prod_{i=0}^{39} (\sin(x_i) + 2), \prod_{i=0}^{39} (1 - y_i^2) > 0.5 \\ \prod_{i=0}^{39} (1 - z_i^2) \end{cases}$$

15. Даны 30-элементные вещественные векторы x, y, z . Вычислить величину $(a, a) - (b, c)$, где a обозначает тот из векторов, в котором самый большой минимальный элемент (считать, что такой вектор единственный), b, c обозначают два других вектора, а (p, q) – скалярное произведение p и q .

Задание 2.2

Двухмерные массивы (матрицы)

Дана целочисленная матрица размером $n \times m$. Найти максимальный и минимальный элементы матрицы и поменять их местами.

```
#include <iostream.h>
int maximum (int **matr, int n, int m, int& Nstr, int& Nst);      //1
int minimum (int **matr, int n, int m, int& Nstr, int& Nst);      //2
int main ()
{
    int **matr, n, m, Nstrmin, Nstrmax, Nstmax, min, max, i, j;
    cout << "Enter size of matrix:\n";
    cin >> n >> m;
    matr = new int* [n];                                              //3
    for ( i = 0; i < n; i++)
        matr [i] = new int [m];                                         //4
    cout << "Enter elements of matrix:\n";
    for ( i = 0; i < n; i++)
        for ( j = 0; j < m; j++)
            cin >> matr [i] [j];
    max = maximum (matr, n, m, Nstrmax, Nstmax);                      //5
    min = minimum (matr, n, m, Nstrmin, Nstmin);                         //6
    matr [Nstrmax] [Nstmax] = min;                                       //7
    matr [Nstrmin] [Nstmin] = max;                                       //8
    cout << "\n\n New matrix:\n";
    for ( i = 0; i < n; i++)
    {
        for ( j = 0; j < m; j++)
            cout << matr [i] [j] << " ";
            cout << endl;
    }
    return 0;
}
int maximum (int **matr, int n, int m, int& Nstr, int& Nst)
{
    int max, i, j;
    max = matr [0] [0];                                                 //9
    Nstr = Nst = 0;                                                     //10
    for ( i = 0; i < n; i++)
        for ( j = 0; j < m; j++)
```

```

if (max < matr [i] [j])
{
    max = matr [i] [j];
    Nstr = i;
    Nst = j;
}
return max;
}

int minimum (int **matr, int n, int m, int& Nstr, int& Nst)
{
    int min, i, j;
    min = matr [0] [0]; //11
    Nstr = Nst = 0; //12
    for ( i = 0; i < n; i++)
    for ( j = 0; j < m; j++)
        if (min > matr [i] [j])
    {
        min = matr [i] [j];
        Nstr = i;
        Nst = j;
    }
    return min;
}

```

//1 и //2 – прототипы функций для нахождения максимума и минимума матрицы соответственно. В качестве аргументов функций передаем в функции матрицу, число строк и число столбцов, а также по ссылке номер строки максимума или минимума соответственно, а также номер столбца максимума или минимума соответственно; из функций будут возвращены максимум или минимум соответственно.

//3 – выполняем захват памяти в heap области для массива указателей.

//4 – выделяем память под элементы матрицы в heap области.

//5 и //6 – выполняем вызов функций для нахождения максимума и минимума соответственно. В функции передаем реальные параметры: адрес нулевого элемента матрицы, количество строк матрицы, количество ее столбцов, адрес переменной номера строки максимума или минимума соответственно, адрес переменной номера столбца максимума или минимума соответственно.

//7 и //8 – вместо максимума или минимума присваивается значение минимума или максимума.

//9 – локальной переменной максимума присваивается начальному значению (первому элементу матрицы).

//10 – локальным переменным номеру строки максимума и номеру столбца максимума присваиваются начальные значения позиции переменной максимума.

//11- локальной переменной минимума присваивается начальному значению (первому элементу матрицы).

//12 – локальным переменным номеру строки минимума и номеру столбца минимума присваиваются начальные значения позиции минимума.

Результаты выполнения программы представлены на рисунке 2.2.

The screenshot shows a terminal window with the following text:

```
Enter size of matrix:  
3 4  
Enter elements of matrix:  
2 4 8 5  
3 4 1 7  
9 8 6 2  
  
New matrix:  
2 4 8 5  
3 4 9 7  
1 8 6 2  
Press any key to continue
```

Рисунок 2.2 – Результаты выполнения программы задания 2.2

Индивидуальные задания

1. Дано целочисленная прямоугольная матрица. Определить:
 - а) количество строк, не содержащих ни одного нулевого элемента;
 - б) максимальное число, из встречающихся в заданной матрице более одного раза.
2. Дано целочисленная прямоугольная матрица. Определить количество столбцов, не содержащих ни одного нулевого элемента. Характеристикой строки целочисленной матрицы назовем сумму ее положительных четных элементов. Переставляя строки заданной матрицы, надо расположить их в соответствии с ростом характеристик.
3. Дано целочисленная прямоугольная матрица. Определить:
 - а) количество столбцов, содержащих хотя бы один нулевой элемент;
 - б) номер строки, в котором находится самая длинная серия одинаковых элементов.
4. Дано целочисленная прямоугольная матрица. Определить:
 - а) произведение элементов в тех строках, которые не содержат отрицательных элементов;
 - б) максимум среди сумм диагоналей, параллельных главной диагонали матрицы.
5. Дано целочисленная прямоугольная матрица. Определить:
 - а) сумму элементов в тех строках, которые не содержат отрицательных элементов;
 - б) минимум среди сумм модулей элементов диагоналей, параллельных побочной диагонали матрицы.
6. Дано целочисленная прямоугольная матрица. Определить:
 - а) сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент;

б) номера строк и столбцов всех седловых точек матрицы.

Матрица А имеет седловую точку A_{ij} , если A_{ij} является минимальным элементом в i-ой строке, и максимальным – в j-м столбце.

7. Для заданной матрицы размером 8×8 найти такие k, при которых k-я строка матрицы совпадает с k-м столбцом. Найти сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент.

8. Характеристикой столбца целочисленной матрицы назовем сумму модулей его отрицательных нечетных элементов. Требуется переставляя столбцы заданной матрицы, расположить их в соответствии с ростом характеристик. Найти сумму элементов в тех столбцах, которые содержат хотя бы один отрицательный элемент.

9. Соседями элемента A_{ij} в матрице назовем элементы A_{kl} , где $i - 1 \leq k \leq i + 1, j - 1 \leq l \leq j + 1, (k,l) \neq (i,j)$. Операция сглаживания матрицы дает новую матрицу того же размера, каждый элемент которой получается как среднее арифметическое имеющихся соседей соответствующего элемента исходной матрицы. Требуется построить результат сглаживания заданной вещественной матрицы размером 10×10 . В сглаженной матрице найти сумму модулей элементов, расположенных ниже главной диагонали.

10. Элемент матрицы называется локальным минимумом, если он меньше своих соседей. Соседями элемента A_{ij} в матрице назовем элементы A_{kl} , где $i - 1 \leq k \leq i + 1, j - 1 \leq l \leq j + 1, (k,l) \neq (i,j)$. Подсчитать количество локальных минимумов заданной матрицы размером 10×10 . Найти сумму модулей элементов, расположенных выше главной диагонали.

11. Коэффициенты системы линейных уравнений заданы в виде прямоугольной матрицы. С помощью допустимых преобразований привести систему к треугольному виду. Найти количество строк, среднее арифметическое элементов которых меньше заданной величины.

12. Уплотнить заданную матрицу, удаляя из нее строки и столбцы, заполненные нулями. Найти номер первой из строк, содержащих хотя бы один положительный элемент.

13. Осуществить циклический сдвиг элементов квадратной матрицы размером $M \times M$ вправо на k элементов таким образом: элементы первой строки сдвигаются в последний столбец сверху вниз, из него – в последнюю строку справа налево, из нее – в первый столбец снизу вверх, из него в первую строку; для остальных элементов аналогично.

14. Осуществить циклический сдвиг элементов прямоугольной матрицы на p элементов вправо или вниз (в зависимости от введенного режима). p может быть больше количества элементов в строке или столбце.

15. Путем перестановки элементов квадратной вещественной матрицы добиться того, чтобы ее максимальный элемент находился в левом верхнем углу, следующий по величине – в позиции (2,2), следующий по величине – в позиции (3,3) и т. д., заполнив таким образом всю главную диагональ. Найти номер первой из строк, не содержащих ни одного положительного элемента.

Задание 2.3

Структура

Для библиотечных книг, задаются: регистрационный номер, автор, название, год издания, издательство, количество страниц. Вывести список книг с фамилиями авторов в алфавитном порядке, изданных после заданного года.

```
#include <iostream.h>
#include <string.h>
struct Library //1
{
    int file_number;
    char writer[30];
    char title[60];
    int year;
    char publishing_house[30];
    int quantity;
};
Library Books[100]; //2
void new_line(); //3
void Enter(Library *B, int& n); //4
void Sort(Library *B, int n); //5
int main() //6
{
    int date, n, i;
    cout << "Enter date:\n";
    cin >> date;
    Enter(Books, n);
    Sort (Books, n);
    for ( i=0; i <n; i++)
    {
        if (Books[i].year >= date) //10
            cout << Books[i].file_number << " " << Books[i].writer << " "
                << Books[i].title << " " << Books[i].year << " "
                << Books[i].publishing_house << " " << Books[i].quantity << endl;
    }
    return 0;
}
void new_line()
{
    char symbol;
    do {
        cin.get (symbol);
    } while (symbol != '\n');
}
void Enter (Library *B, int& n)
{
    cout <<"Enter number books \n\n";
    cin >> n;
    for(int i = 0; i <n; i++)
    {
        cout << "Enter file_number:\n";
        cin >> B[i].file_number;
        cout << "Enter writer\n";
    }
}
```

```

    cin >> B[i]. writer;
    cout << "Enter title\n";
    new_line(); //11
    cin.getline (B[i]. title, 60); //12
    cout << "Enter year\n";
    cin >> B[i]. year;
    cout << "Enter publishing_house\n";
    new_line();
    cin.getline (B[i]. publishing_house, 30);
    cout << "Enter quantity\n";
    cin >> B[i]. quantity;
}
}

void Sort (Library *B, int n)
{
    Library buffer; //13
    for (int i = 0; i < n-1; i++)
        for (int j = i+1; j < n; j++)
            if (strcmp(B[i]. writer, B[j]. writer) > 0) //14
            {
                buffer = B[i];
                B[i] = B[j];
                B[j] = buffer;
            }
}
}

```

//1 – описание типа struct с именем структуры Library.

//2 – окончание описания типа структуры – закрывающая фигурная скобка с точкой с запятой, причем, точка с запятой обязательна.

//3 – объявление массива структурных переменных Books.

//4 - прототип функции, которая будет очищать буфер ввода при вводе структурных переменных.

//5 – прототип функции ввода структурных переменных с клавиатуры, в качестве аргументов передаем массив структурных переменных, а также их количество. Поскольку ввод количества структурных переменных осуществляется в функции Enter(), то мы передаем туда переменную n по ссылке.

//6 – прототип функции сортировки структурных переменных по авторам, т. к. в Задании сказано, что необходимо вывести на экран авторов в алфавитном порядке.

//7 – объявление переменной date, в которой будет находиться год издания книги, после которого нужно выводить на экран список книг.

//8 – вызов функции для ввода структурных переменных: в функцию передаем адрес нулевого элемента переменной Books и адрес переменной n.

//9 – вызов функции для сортировки массива структурных переменных Books по авторам.

//10 – сравниваем год издания книги с годом, после которого будем выводить на экран список книг.

//11 – вызов функции new_line(), чтобы очистить буфер ввода для строковых переменных.

//12 – вызов функции-члена (метода) get_line(), которая является членом всех входных потоков; аргументы – строковая переменная и максимальная длина строки.

//13 – объявление структурной переменной, которая будет служить буфером при сортировке массива структурных переменных.

//14 – вызов функции сравнения строковых переменных (библиотека string.h). В данном случае нам нужен положительный результат, чтобы сортировка произошла в алфавитном порядке.

Результаты выполнения программы представлены на рисунке 2.3.

```
6x
Enter date:
2003
Enter number books n
4
Enter file_number:
2839
Enter writer
Пушкин
Enter title
Евгений Онегин
Enter year
2003
Enter publishing_house
Москва
Enter quantity
95
Enter file_number:
3478
Enter writer
Достоевский
Enter title
Братья Карамазовы
Enter year
2008
Enter publishing_house
Минск
Enter quantity
652
Enter file_number:
3874
Enter writer
Гоголь
Enter title
Мертвые души
Enter year
2000
Enter publishing_house
Санкт-Петербург
Enter quantity
44?
Enter file_number:
3782
Enter writer
Короткевич
Enter title
Колосья под серпом твоим
Enter year
2007
Enter publishing_house
Минск
Enter quantity
684
3478 Достоевский Братья Карамазовы 2008 Минск 652
3782 Короткевич Колосья под серпом твоим 2007 Минск 684
2839 Пушкин Евгений Онегин 2003 Москва 95
Press any key to continue
```

Рисунок 2.3 – Результаты выполнения программы задания 2.3

Индивидуальные задания

1. Данна ведомость абитуриентов, сдавших вступительные экзамены в институт. В каждой строке этой ведомости записаны фамилия абитуриента, его постоянное местожительства и полученные им оценки по отдельным дисциплинам (например, физике, математике, литературе). Необходимо определить количество абитуриентов, проживающих в городе Минске и сдавших экзамены со средним баллом не ниже 7, распечатать их фамилии в алфавитном порядке.

2. В справочной службе аэропорта хранится расписание вылета самолетов на следующие сутки. Для каждого рейса указаны его номер, тип самолета, пункт назначения, время вылета. Определить все номера рейсов, типы самолетов и время их вылета в заданный пункт назначения.

3. У администратора железнодорожных касс хранится информация о свободных местах в поездах по всем направлениям на ближайшую неделю. Данная информация представлена в следующем виде: дата выезда, конечный пункт назначения, время отправления, количество свободных купейных мест, количество свободных плацкартных мест. Оргкомитет международной конференции обращается к администратору с просьбой зарезервировать 50 купейных мест до Берлина на субботу. При этом время отправления поезда должно быть не позднее 10 часов вечера. Выдать на печать время отправления или сообщение о невозможности выполнить заказ в полном объеме.

4. Написать программу формирования ведомости об успеваемости студентов. Каждая запись этой ведомости должна содержать номер группы, фамилию студента, средний балл за последнюю сессию. Необходимо распечатать списки студентов по группам. В каждой группе фамилии разместить в порядке убывания среднего балла.

5. В предвыборной кампании проводится регистрация кандидатов в депутаты. Каждый кандидат, подавая заявление на регистрацию, указывает номер округа, в котором он собирается баллотироваться, наименование партии, которую он представляет, свой возраст и профессию. Пресс-служба центральной избирательной комиссии выдает информационный бюллетень, в котором содержится следующая информация: количество поданных заявлений на регистрацию кандидатов от каждой политической партии, средний возраст этих кандидатов, наиболее распространенная профессия кандидатов.

6. Сведения об автомобиле включают его марку, номер и фамилию владельца. Представлен массив, содержащий сведения о нескольких автомобилях. Найти:

- а) фамилии владельцев и номера автомобилей данной марки;
- б) количество автомобилей каждой марки.

7. Для получения места в общежитии формируют списки студентов, которые включают ФИО студента, группу, средний балл, доход на члена семьи меньше двух минимальных зарплат, затем – остальных, в порядке уменьшения среднего балла. Вывести список очередности представления мест в общежитии.

8. В справочной службе автовокзала хранится расписание движения автобусов. Для каждого рейса указаны его номер, тип автобуса, пункт назначения, время отправления и прибытия. Вывести информацию о рейсах, которыми

можно воспользоваться для прибытия в пункт назначения раньше заданного времени.

9. Информация об участниках спортивных соревнований содержит: наименование страны, название команды, ФИО игрока, игровой номер, возраст, рост, вес. Вывести информацию о самой молодой, рослой и легкой команде.

10. Информация о сотрудниках предприятия содержит: ФИО, номер отдела, должность, дату начала работы. Вывести списки сотрудников по отделам в порядке убывания стажа.

11. Имеют отдельные списки женихов и невест. Каждая запись списка включает пол, имя, возраст, рост, вес, а также требования к партнеру: наименьший и наибольший возраст, наименьший и наибольший вес, наименьший и наибольший рост. Требуется объединить эти списки в список пар, с учетом требований к партнерам.

12. В библиотеке имеется список книг (каталог). Каждая запись книжного каталога содержит: фамилии авторов, название книги, год издания. Вывести информацию о книгах, в названии которых встречается некоторое ключевое слово (ввести с клавиатуры).

13. Каждая запись списка вакантных рабочих мест содержит: наименование организации, должность, квалификацию (разряд или образование), стаж работы по специальности, заработную плату, наличие социального страхования (да/нет), продолжительность ежегодного оплачиваемого отпуска. Вывести список рабочих мест в соответствии с требованиями клиента.

14. Для участия в конкурсе на замещение вакантной должности сотрудника фирмы желающие подают следующую информацию: Ф.И.О., год рождения, образование (среднее, специальное, высшее), знание иностранных языков (английский, немецкий, французский); владею свободно, читаю и перевожу со словарем, владение компьютером (MSDOS, Windows), стаж работы, наличие рекомендаций. Вывести список претендентов в соответствии с требованиями руководства фирмы.

15. При постановке на учет в ГАИ автолюбители указывают следующие данные: марка автомобиля, год выпуска, номер двигателя, номер кузова, цвет, номерной знак, Ф.И.О. и адрес владельца. Вывести список автомобилей, проходящих техосмотр в текущем году, сгруппированный по маркам автомобилей. Учесть, что если текущий год четный, техосмотр проходят автомобили с четными номерами двигателей, иначе – с нечетными номерами.

Лабораторная работа №3

КЛАССЫ. КОНСТРУКТОРЫ. ПЕРЕГРУЗКА МЕТОДОВ. ФАЙЛЫ

Задание 3.1

Классы. Конструкторы

Создайте класс Pairs. Напишите реализацию конструктора по умолчанию, а также конструкторов с одним и двумя аргументами типа int. Конструктор с одним параметром должен инициализировать первый элемент пары значением, указанным в аргументе, а второй элемент – 0; конструктор по умолчанию инициализирует оба значения нулями. Напишите программу драйвер, который создает и отображает объекты класса.

```
#include <iostream.h>
class Pairs //1
{
public: //2
    Pairs(int a, int b); //3
    Pairs(int a); //4
    Pairs(); //5
    int get_first(); //6
    int get_second(); //7
    void Sum(); //8
protected: //9
    int first, second; //10
};
Pairs:: Pairs(int a, int b) //11
{
    first = a; //13
    second = b; //14
}
Pairs:: Pairs(int a) //15
{
    first = a;
    second = 0;
}
Pairs:: Pairs()
{
    first = 0;
    second = 0;
}
int Pairs::get_first()
{
    return first;
}
int Pairs::get_second()
{
    return second;
}
void Pairs::Sum()
{
    cout << "first + second =" << first + second << endl;
```

```

    }
int main()
{
    Pairs object(7,9), object1(28), object2; //16
    object.Sum(); //17
    object1.Sum(); //18
    object2.Sum(); //19
    cout << "Value first object equal to " << object.get_first() << " "
    << object.get_second() << endl;
    cout << "Value second object equal to " << object1.get_first() << " "
    << object1.get_second() << endl;
    cout << "Value third object equal to " << object2.get_first() << " "
    << object2.get_second() << endl;
    return 0;
}

```

//1 – описание типа класс, ключевое слово class, имя класса Pairs.

//2 – описываем публичные члены класса, к ним можно обращаться через переменные объекты.

//3 – прототип конструктора с двумя аргументами (смотрите условие задания).

//4 – прототип конструктора с одним аргументом.

//5 – прототип конструктора по умолчанию (без аргументов).

//6 – прототип метода доступа к первой переменной объекта, поскольку в классе все переменные являются частными или защищенными, к ним можно обращаться только через методы доступа.

//7 – прототип метода доступа ко второй переменной объекта.

//8 – прототип метода нахождения суммы двух переменных.

//9 – спецификатор частный, ибо дальше пойдут закрытые члены класса: к ним можно обращаться только через методы класса.

//10 – объявление закрытых переменных целого типа (смотрите условие задания).

//11 – описание класса Pairs закончилось (закрытая фигурная скобка точкой с запятой).

//12 – определение конструктора с двумя переменными. Это заголовок функции, в нем обязательно присутствует имя класса (Pairs), а также оператор области видимости (::).

//13 – первая переменная объекта инициализируется значением первого аргумента (*a*) конструктора.

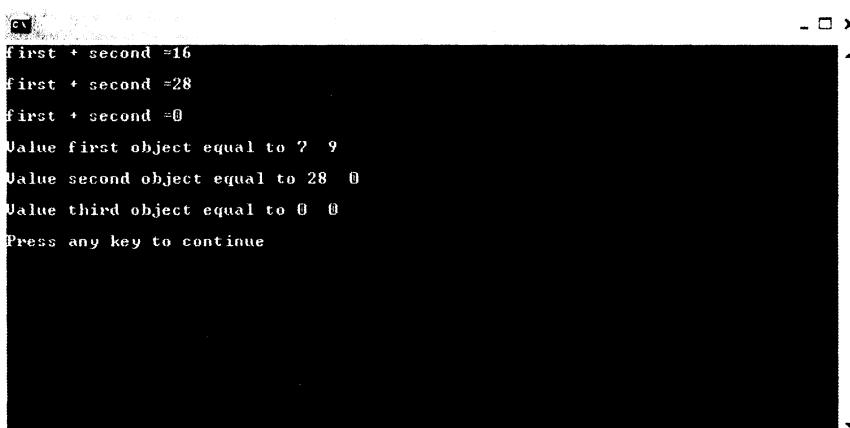
//14 – вторая переменная объекта инициализируется значением второго аргумента (*b*) конструктора.

//15 – определение конструктора с одним аргументом.

//16 – объявление переменных (объектов) типа класс (Pairs) . Первая переменная (object) инициализируется конструктором с двумя аргументами, вторая переменная (object1) инициализируется конструктором с одним аргументом, второй аргумент инициализируется значением по умолчанию (0), третья переменная (object2) инициализируется конструктором по умолчанию , круглые скобки не пишутся, чтобы компилятор мог отличить переменную от функции.

//17 – вызов метода Sum() для первого объекта.
//18 – вызов метода Sum() для второго объекта.
//19 – вызов метода Sum() для третьего объекта.

Результаты выполнения программы представлены на рисунке 3.1.



```
first + second =16
first + second =28
first + second =0
Value first object equal to ? 9
Value second object equal to 28 0
Value third object equal to 0 0
Press any key to continue
```

Рисунок 3.1 – Результаты выполнения программы задания 3.1

Индивидуальные задания

1. Создайте класс Square. Напишите функции-члены, которые вычисляют квадрат целого и квадрат вещественного чисел. Используйте перегрузку имени функции. Переменные должны быть проинициализированы конструктором с двумя аргументами. Напишите программу драйвер, который создает и отображает объекты класса.
2. Создайте класс Average. Напишите функции-члены, которые вычисляют среднее арифметическое двух и трех чисел. Используйте перегрузку имени функции. Переменные должны быть проинициализированы конструктором с тремя аргументами. Напишите программу драйвер, который создает и отображает объекты класса.
3. Создайте класс Figure. Напишите функции-члены, которые вычисляют площади треугольника и прямоугольника. Используйте перегрузку имени функции. Переменные должны быть проинициализированы конструктором с двумя аргументами. Напишите программу драйвер, который создает и отображает объекты класса.
4. Создайте класс Stepen. Напишите функции-члены, которые возводят число в степень n для целого и вещественного чисел. Используйте перегрузку имени функции. Переменные должны быть проинициализированы конструктором с тремя аргументами. Напишите программу драйвер, который создает и отображает объекты класса.

5. Создайте класс Max. Напишите функции-члены, которые находят максимум двух и трех чисел. Используйте перегрузку имени функции. Переменные должны быть проинициализированы конструктором с тремя аргументами. Напишите программу драйвер, который создает и отображает объекты класса.

6. Создайте класс Min. Напишите функции-члены, которые определяют минимум двух и трех чисел. Используйте перегрузку имени функции. Переменные должны быть проинициализированы конструктором с тремя аргументами. Напишите программу драйвер, который создает и отображает объекты класса.

7. Создайте класс Area. Напишите функции-члены, которые вычисляют площадь треугольника: 1) по основанию и высоте; 2) по формуле Герона. Используйте перегрузку имени функции. Переменные должны быть проинициализированы конструктором с тремя аргументами. Напишите программу драйвер, который создает и отображает объекты класса.

8. Создайте класс Product. Напишите функции-члены, которые вычисляют произведение для двух и для трех чисел. Используйте перегрузку имени функции. Переменные должны быть проинициализированы конструктором с тремя аргументами. Напишите программу драйвер, который создает и отображает объекты класса.

9. Создайте класс Summa. Напишите функции-члены, которые вычисляют суммы для двух и для трех чисел. Используйте перегрузку имени функции. Переменные должны быть проинициализированы конструктором с тремя аргументами. Напишите программу драйвер, который создает и отображает объекты класса.

10. Создайте класс Revers. Напишите функции-члены, которые возвращают целое число и строку символов в обратном порядке. Используйте перегрузку имени функции. Переменные должны быть проинициализированы конструктором с двумя аргументами. Напишите программу драйвер, который создает и отображает объекты класса.

11. Рассмотрите депозитный банковский сертификат, часто называемый CD. CD – это банковский счет, запрещающий снятие денежных средств в течение определенного периода (как правило, несколько месяцев). С ним связываются три элемента данных: баланс счета, процентная ставка по счету и срок вклада в месяцах. Переменная-член баланса должна храниться в виде двух значений типа int – одно для долларов и второе для центов; процентную ставку можно представить значением типа double, а срок вклада – значением типа int. Сделайте переменные-члены закрытыми. Включите в класс функции-члены для получения следующих значений: начального баланса, баланса по истечении срока вклада, процентных ставок и срока вклада. Включите в определение класса конструктор по умолчанию. Включите также в определение класса функцию-член для ввода данных с одним формальным параметром типа istream и функцию-член для вывода информации с одним формальным параметром типа ostream. Используйте класс в тестовой программе.

12. Определите класс CounterType. Объект этого типа используется для подсчета предметов, т.е. он хранит счетчик в виде неотрицательного целого числа. Включите в определение класса конструктор по умолчанию, устанавлив-

вающий счетчик на нулевое значение, и конструктор с одним аргументом, устанавливающий счетчик на задаваемое аргументом значение. Предусмотрите функции-члены для увеличения и уменьшения значения счетчика на единицу. Позаботьтесь, чтобы ни одна функция-член не позволяла счетчику принять отрицательное значение. Кроме того, включите в определение функцию-член, возвращающую текущее значение счетчика, и функцию, осуществляющую вывод значения счетчика в поток. Функция-член для осуществления вывода должна иметь один формальный параметр типа `ostream` для выходного потока. Используйте класс в тестовой программе.

13. Определите класс `Month`, являющийся абстрактным типом данных для месяца. Ваш класс должен храниться в виде трех переменных-членов типа `char`, представляющих первые три буквы названия месяца. Включите в класс следующие функции-члены: конструктор для задания месяца с использованием первых трех букв названия месяца в качестве трех аргументов; конструктор для задания месяца, использующий одно целое число в качестве аргумента (1 для января, 2 для февраля и т.д.); конструктор по умолчанию; функцию ввода,читывающую месяц в виде первых трех букв названия месяца; функцию вывода, осуществляющую вывод месяца в виде числа; функцию вывода, осуществляющую вывод месяца в виде первых трех букв названия месяца; функцию-член, возвращающую следующий месяц в виде значения типа `Month`. Функции ввода и вывода должны иметь один параметр потока. Используйте класс в тестовой программе.

14. Напишите и реализуйте функции-члены класса `Counter`, который имитирует поведение калькулятора. Конструктор должен создавать объект `Counter`, который может вести подсчет до максимального значения , не превышающего значения аргумента конструктора (т.е `Counter(9999)` должен создавать калькулятор с максимальным возможным значением 9999). Показания вновь созданного калькулятора должна быть равны 0. Функция-член `void reset();` устанавливает значение счетчика калькулятора равным 0. Функции-члены `void inc1();`, `void inc10();`, `void inc100();` и `void inc1000();` предназначены для увеличения на единицу разрядов единиц, десятков, сотен и тысяч соответственно. Перенос единицы в старший разряд при введении стоимости покупки должен осуществляться автоматически и состоять в увеличении соответствующей переменной-члена. Функция-член `bool overflow();` предназначена для обнаружения переполнения (переполнение – это результат увеличения значения счетчика сверх величины, указанной в конструкторе при объявлении соответствующего объекта).Предусмотрите клавиши для ввода центов, десятков центов, долларов и десятков долларов.

15. Создайте класс `Complex` для выполнения арифметических действий с комплексными числами. Напишите программу драйвера для проверки вашего класса.

Комплексные числа имеют форму:

`realPart + imaginaryPart * j,`

где `j` – квадратный корень из `-1`.

Используйте переменные с плавающей запятой для представления закрытых данных этого класса. Создайте функцию конструктор, которая позволяет

объекту этого класса принимать начальные значения при его объявлении. Создайте открытые функции-члены для каждого следующего пункта:

- a) сложение двух комплексных чисел: отдельно складывают действительные и мнимые части;
- b) вычитание двух комплексных чисел: действительная часть правого операнда вычитается из действительной части левого операнда, а мнимая часть правого операнда из мнимой части левого операнда;
- v) Печать комплексных чисел в форме (a, b), где a – действительная часть, а b – мнимая часть.

Задание 3.2

Файлы

Написать программу, которая считывает английский текст из файла и выводит на экран сначала вопросительные, а затем восклицательные предложения.

```
#include <iostream.h>
#include <fstream.h>
#include <stdlib.h>
#include <string.h>
void new_line(); //1
file1(); //2
int main()
{
    file1();
    return 0;
}
void new_line()
{
    char symbol;
    do {
        cin.get (symbol);
    } while (symbol != '\n');
}

void file1()
{
    ifstream in_stream; //3
    char **mas_line, *str;
    int i=0, n=0, k;
    str = new char [60];
    mas_line = new char*[20];
    for ( i = 0; i <20; i++)
        mas_line[i] = new char [60];
    in_stream.open ("f.txt");
    if (in_stream.fail()) //4
        { cout << "Cannot open input file\n";
          exit(1); } //5
    i=0;
    while (!in_stream.eof())
    {
        new_line(); //6
    }
}
```

```

    new_line());
    in_stream.getline(str, 60); //8;
    strcpy(mas_line[i],str);
    i++;
    n++;
}
for (i=0; i<n; i++)
{ k = strlen(mas_line[i]);
  if (mas_line[i] [k-1] == '!') //9
  cout << mas_line[i] << endl;
}
for (i=0; i<n; i++)
{ k = strlen(mas_line[i]);
  if (mas_line[i] [k-1] == '?') //10
  cout << mas_line[i] << endl;
}
in_stream.close(); //12
}

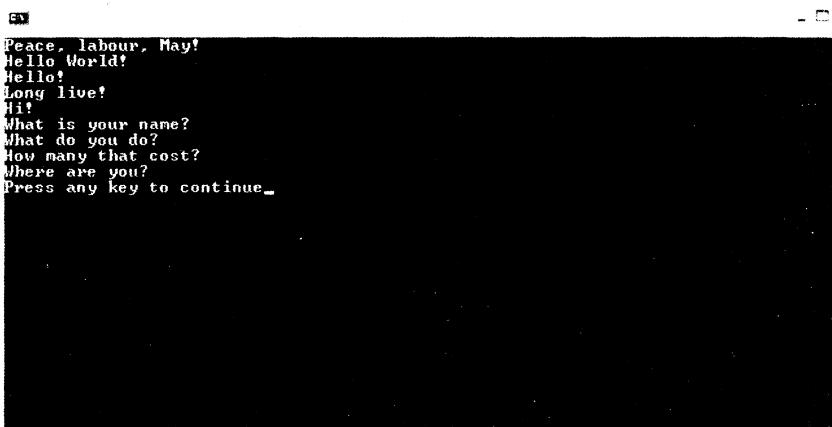
//1 – директивой #include подключаем стандартную библиотеку файлового потока ввода/вывода.
//2 – подключаем стандартную библиотеку функций, для функции exit().
//3 – имя класса входного файлового потока (ifstream). Объявляем переменную объект для работы с файлом f.txt.
//4 – связываем физический файл (f.txt) на диске с файловой переменной (in_stream), используя метод open() из библиотеки fstream.
//5 – проверка с помощью метода fail(), был ли открыт файл на диске.
//6 – если файл не был открыт, с помощью функции exit() выходим из программы.
//7 – если файл открыт, продолжается выполнение программы: здесь осуществляется проверка конца файла с помощью метода eof().
//8 – с помощью метода getline() читаем записи из файла f.txt.
//9 – вычисляем длину каждой записи, чтобы потом определить знак препинания (! или ?).
//10, //11 – определение знака препинания (! или ?) соответственно.
//12 – с помощью метода close() закрываем файл f.txt.

```

Результаты выполнения программы представлены на рисунке 3.2.

Индивидуальные задания

1. Написать программу, которая считывает из текстового файла три предложения и выводит их в обратном порядке.
2. Написать программу, которая считывает текст из файла и выводит на экран только предложения, содержащие введенное с клавиатуры слово.
3. Написать программу, которая считывает текст из файла и выводит на экран только строки, содержащие двузначные числа.
4. Написать программу, которая считывает текст из файла и выводит на экран слова, начинающиеся с гласных букв.
5. Написать программу, которая считывает текст из файла и выводит на экран, меняя местами, каждые два соседних слова.



```
Peace, labour, May!
Hello World!
Hello!
Hello!
Long live!
Hi!
What is your name?
What do you do?
How many that cost?
Where are you?
Press any key to continue...
```

Рисунок 3.2 – Результаты выполнения программы задания 3.2

6. Написать программу, которая считывает текст из файла и определяет, сколько в нем слов, состоящих не более чем из четырех букв.
7. Написать программу, которая считывает текст из файла и выводит на экран только предложения, не содержащие запятых.
8. Написать программу, которая считывает текст из файла и выводит на экран только цитаты, т. е. предложения заключенные в кавычки.
9. Написать программу, которая считывает текст из файла и выводит на экран только предложения, с заданным количеством слов.
10. Написать программу, которая считывает английский текст из файла и выводит на экран слова текста, начинающиеся и оканчивающиеся на гласные.
11. Написать программу, которая считывает текст из файла и выводит на экран только строки, не содержащие двузначных чисел.
12. Написать программу, которая считывает текст из файла и выводит на экран только предложения, начинающиеся с тире, перед которым могут находиться только пробельные символы.
13. Написать программу, которая считывает английский текст из файла и выводит его на экран, заменяя прописной каждую первую букву слов, начинающихся с гласной.
14. Написать программу, которая считывает текст из файла и выводит его на экран, заменяя цифры от 0 до 9 словами «ноль», «один» ... «девять» и начиная каждое предложение с новой строки.
15. Написать программу, которая считывает текст из файла, находит самое длинное слово и определяет, сколько раз оно встретилось в тексте.

Лабораторная работа №4

ДРУЖЕСТВЕННЫЕ ФУНКЦИИ. ПЕРЕГРУЗКА ОПЕРАТОРОВ

Задание 4.1

Напишите программу, которая должна содержать пользовательский класс *Array*, включающий:

- конструктор с параметрами для создания динамических целочисленных массивов и установки начальных значений их элементов: *Array(...)* (реальные размеры массива – количество строк и столбцов передается в конструктор через параметры);

- деструктор *~Array()*;

- функцию-член печати текущего состояния массива *void Print(...)*;

- функцию-член переустановки текущего значения массива *void Set(...)*;

- функцию-друга, которая определяет сумму элементов в тех строках массива, которые не содержат отрицательных элементов.

Программа должна проиллюстрировать косвенный способ обращения к элементам массива. Коды функции-члена и функции-друга определите вне пространства класса:

```
#include <iostream.h>
class Array
{
public:
    Array (int a, int b);
    ~Array(); //1
    void Print();
    void Set();
    friend void Run( Array& object); //2
private:
    int **mas, n, m;
};
Array:: Array(int a, int b)
{
    n = a;
    m = b;
    mas = new int* [n];
    for (int i = 0; i < n; i++)
        mas[i] = new int [m];
    for ( i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            mas[i][j] = 0;
}
Array:: ~Array() //3
{
    for (int i = 0; i < n; i++)
        delete mas[i];
    delete []mas;
}
void Array::Print()
{
```

```

for (int i = 0; i < n; i++)
{
    for (int j = 0; j < m; j++)
        cout << mas[i][j] << " ";
    cout << endl;
}
cout << endl;
}
void Array::Set()
{
    cout << "Enter elements of matrix: \n";
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            cin >> mas[i][j];
}
void Run(Array& object) //4
{
    int i, j, Sum;
    bool negative;
    for (i = 0; i < object.n; i++)
    { Sum = 0;
        negative = false;
        for (j = 0; j < object.m; j++)
            if (object.mas[i][j] < 0)
                { negative = true; break; }
        if (!negative)
            for (j = 0; j < object.m; j++)
                Sum += object.mas[i][j];
        if (!negative) cout << "Sum " << i + 1 << " of line equal " << Sum << endl;
    }
}
int main() //5
{
    Array array(3,4);
    array.Print();
    array.Set();
    Run(array);
    return 0;
}

```

//1 – прототип деструктора. Деструктор, как и конструктор, имеет имя класса, перед которым стоит знак тильда (~).

//2 – прототип функции-друга. Перед типом возвращаемого значения стоит ключевое слово friend, которое позволяет самой обычной функции (не принадлежащей классу) обращаться к закрытым и частным членам класса.

//3 – описание деструктора. Заголовок деструктора описывается также как конструктора и методов, т. е. пишется имя класса и оператор разрешения области видимости (::).

//4 – описание функции-друга. В заголовке не пишется имя класса и отсутствует оператор разрешения области видимости, поскольку эта функция не является членом класса.

//5 – вызов функции-друга. Вызов ничем не отличается от вызова самой обычной функции, т. е. не надо писать имя объекта и оператор-точку.

Результаты выполнения программы представлены на рисунке 4.1.

```
"D:\_CatalinMetodicheskaya\Debug\Debug1.exe"
0 0 0 0
0 0 0 0
0 0 0 0
Enter elements of matrix:
3 7 2 3
2 9 4 1
9 3 7 2
Sum 1 of Line equal to 16
Sum 2 of Line equal to 16
Sum 3 of Line equal to 21
Press any key to continue...
```

Рисунок 4.1 – Результаты выполнения программы задания 4.1

Индивидуальные задания

1. Напишите программу, которая должна содержать пользовательский класс *Array*, включающий:
 - конструктор с параметрами для создания динамических целочисленных массивов и установки начальных значений их элементов: *Array(...)* (реальные размеры массива – количество строк и столбцов передается в конструктор через параметры);
 - деструктор *~Array()*;
 - функцию-член печати текущего состояния массива *void Print(...)*;
 - функцию-член переустановки текущего значения массива *void Set(...)*;
 - функцию-друга, которая определяет количество строк, содержащих хотя бы один нулевой элемент.Программа должна проиллюстрировать косвенный способ обращения к элементам массива. Коды функции-члена и функции-друга определите вне пространства класса.
2. Напишите программу, которая должна содержать пользовательский класс *Array*, включающий:
 - конструктор с параметрами для создания динамических целочисленных массивов и установки начальных значений их элементов: *Array(...)* (реальные размеры массива – количество строк и столбцов передается в конструктор через параметры);
 - деструктор *~Array()*;
 - функцию-член печати текущего состояния массива *void Print(...)*;
 - функцию-член переустановки текущего значения массива *void Set(...)*;
 - функцию-друга, которая определяет номер первой строки, из не содержащих положительных элементов.

Программа должна проиллюстрировать косвенный способ обращения к элементам массива. Коды функции-члена и функции-друга определите вне пространства класса.

3. Напишите программу, которая должна содержать пользовательский класс *Array* включающий:

-конструктор с параметрами для создания динамических целочисленных массивов и установки начальных значений их элементов: *Array(...)* (реальные размеры массива – количество строк и столбцов передается в конструктор через параметры):

- деструктор *~Array()*;
- функцио-член печати текущего состояния массива *void Print(...)*;
- функцио-член переустановки текущего значения массива *void Set(...)*;
- функцио-друга, которая определяет номер первого столбца, из содержащих хотя бы один нулевой элемент.

Программа должна проиллюстрировать косвенный способ обращения к элементам массива. Коды функции-члена и функции-друга определите вне пространства класса.

4. Напишите программу, которая должна содержать пользовательский класс *Array*, включающий:

- конструктор с параметрами для создания динамических целочисленных массивов и установки начальных значений их элементов: *Array(...)* (реальные размеры массива – количество строк и столбцов передается в конструктор через параметры):

- деструктор *~Array()*;
- функцио-член печати текущего состояния массива *void Print(...)*;
- функцио-член переустановки текущего значения массива *void Set(...)*;
- функцио-друга, которая определяет номер первого столбца, из не содержащих отрицательных элементов.

Программа должна проиллюстрировать косвенный способ обращения к элементам массива. Коды функции-члена и функции-друга определите вне пространства класса.

5. Напишите программу, которая должна содержать пользовательский класс *Array*, включающий:

-конструктор с параметрами для создания динамических целочисленных массивов и установки начальных значений их элементов: *Array(...)* (реальные размеры массива – количество строк и столбцов передается в конструктор через параметры):

- деструктор *~Array()*;
- функцио-член печати текущего состояния массива *void Print(...)*;
- функцио-член переустановки текущего значения массива *void Set(...)*;
- функцио-друга, которая определяет сумму модулей элементов, расположенных выше главной диагонали.

Программа должна проиллюстрировать косвенный способ обращения к элементам массива. Коды функции-члена и функции-друга определите вне пространства класса.

6. Напишите программу, которая должна содержать пользовательский класс *Array*, включающий:

- конструктор с параметрами для создания динамических целочисленных массивов и установки начальных значений их элементов: *Array(...)* (реальные размеры массива – количество строк и столбцов передается в конструктор через параметры):

- деструктор *~Array()*;

- функцию-член печати текущего состояния массива *void Print(...)*;

- функцию-член переустановки текущего значения массива *void Set(...)*;

- функцию-друга, которая определяет количество строк, не содержащих ни одного нулевого элемента.

Программа должна проиллюстрировать косвенный способ обращения к элементам массива. Коды функции-члена и функции-друга определите вне пространства класса.

7. Напишите программу, которая должна содержать пользовательский класс *Array*, включающий:

- конструктор с параметрами для создания динамических целочисленных массивов и установки начальных значений их элементов: *Array(...)* (реальные размеры массива – количество строк и столбцов передается в конструктор через параметры):

- деструктор *~Array()*;

- функцию-член печати текущего состояния массива *void Print(...)*;

- функцию-член переустановки текущего значения массива *void Set(...)*;

- функцию-друга, которая определяет количество столбцов, не содержащих ни одного нулевого элемента.

Программа должна проиллюстрировать косвенный способ обращения к элементам массива. Коды функции-члена и функции-друга определите вне пространства класса.

8. Напишите программу, которая должна содержать пользовательский класс *Array*, включающий:

- конструктор с параметрами для создания динамических целочисленных массивов и установки начальных значений их элементов: *Array(...)* (реальные размеры массива – количество строк и столбцов передается в конструктор через параметры):

- деструктор *~Array()*;

- функцию-член печати текущего состояния массива *void Print(...)*;

- функцию-член переустановки текущего значения массива *void Set(...)*;

- функцию-друга, которая определяет номер первой из строк, содержащих хотя бы один положительный элемент.

Программа должна проиллюстрировать косвенный способ обращения к элементам массива. Коды функции-члена и функции-друга определите вне пространства класса.

9. Напишите программу, которая должна содержать пользовательский класс *Array*, включающий:

- конструктор с параметрами для создания динамических целочисленных массивов и установки начальных значений их элементов: *Array(...)* (реальные размеры массива – количество строк и столбцов передается в конструктор через параметры):

- деструктор *~Array();*
- функцию-член печати текущего состояния массива *void Print(...);*
- функцию-член переустановки текущего значения массива *void Set(...);*
- функцию-друга, которая определяет произведение элементов в тех строках, которые не содержат отрицательных элементов.

Программа должна проиллюстрировать косвенный способ обращения к элементам массива. Коды функции-члена и функции-друга определите вне пространства класса

10. Напишите программу, которая должна содержать пользовательский класс *Array*, включающий:

- конструктор с параметрами для создания динамических целочисленных массивов и установки начальных значений их элементов: *Array(...)* (реальные размеры массива – количество строк и столбцов передается в конструктор через параметры):

- деструктор *~Array();*
- функцию-член печати текущего состояния массива *void Print(...);*
- функцию-член переустановки текущего значения массива *void Set(...);*
- функцию-друга, которая определяет сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент.

Программа должна проиллюстрировать косвенный способ обращения к элементам массива. Коды функции-члена и функции-друга определите вне пространства класса.

11. Напишите программу, которая должна содержать пользовательский класс *Array*, включающий:

- конструктор с параметрами для создания динамических целочисленных массивов и установки начальных значений их элементов: *Array(...)* (реальные размеры массива – количество строк и столбцов передается в конструктор через параметры):

- деструктор *~Array();*
- функцию-член печати текущего состояния массива *void Print(...);*
- функцию-член переустановки текущего значения массива *void Set(...);*
- функцию-друга, которая определяет сумму элементов в тех столбцах, которые содержат хотя бы один отрицательный элемент.

Программа должна проиллюстрировать косвенный способ обращения к элементам массива. Коды функции-члена и функции-друга определите вне пространства класса.

12. Напишите программу, которая должна содержать пользовательский класс *Array*, включающий:

- конструктор с параметрами для создания динамических целочисленных массивов и установки начальных значений их элементов: *Array(...)* (реальные

размеры массива – количество строк и столбцов передается в конструктор через параметры):

- деструктор `~Array();`
- функцию-член печати текущего состояния массива `void Print(...);`
- функцию-член переустановки текущего значения массива `void Set(...);`
- функцию-друга, которая определяет произведение элементов в тех строках, которые не содержат элементов, кратных 3.

Программа должна проиллюстрировать косвенный способ обращения к элементам массива. Коды функции-члена и функции-друга определить вне пространства класса.

13. Напишите программу, которая должна содержать пользовательский класс `Array`, включающий:

- конструктор с параметрами для создания динамических целочисленных массивов и установки начальных значений их элементов: `Array(...)` (реальные размеры массива – число строк и столбцов передается в конструктор через параметры):

- деструктор `~Array();`
- функцию-член печати текущего состояния массива `void Print(...);`
- функцию-член переустановки текущего значения массива `void Set(...);`
- функцию-друга, которая транспонирует матрицу, для чего поменяйте строки и столбцы местами.

Программа должна проиллюстрировать косвенный способ обращения к элементам массива. Коды функции-члена и функции-друга определите вне пространства класса.

14. Напишите программу, которая должна содержать пользовательский класс `Array`, включающий:

- конструктор с параметрами для создания динамических целочисленных массивов и установки начальных значений их элементов: `Array(...)` (реальные размеры массива – количество строк и столбцов передается в конструктор через параметры):

- деструктор `~Array();`
- функцию-член печати текущего состояния массива `void Print(...);`
- функцию-член переустановки текущего значения массива `void Set(...);`
- функцию-друга, которая увеличивает значения элементов вдвое в тех строках, в которых находится максимальный элемент.

Программа должна проиллюстрировать косвенный способ обращения к элементам массива. Коды функции-члена и функции-друга определить вне пространства класса.

15. Напишите программу, которая должна содержать пользовательский класс `Array`, включающий:

- конструктор с параметрами для создания динамических целочисленных массивов и установки начальных значений их элементов: `Array(...)` (реальные размеры массива – число строк и столбцов передается в конструктор через параметры):

- деструктор `~Array();`
- функцию-член печати текущего состояния массива `void Print(...);`

```

        for(int j =0; j<object1.m; j++)
            temp.matr[i][j] = object1.matr[i][j] + object2.matr[i][j];
        return temp;
    }

    Matrix operator - (Matrix object1, Matrix object2)
    {
        Matrix temp(object1.n,object1.m);
        for(int i =0; i<object1.n; i++)
            for(int j =0; j<object1.m; j++)
                temp.matr[i][j] = object1.matr[i][j] - object2.matr[i][j];
        return temp;
    }

    Matrix operator * (Matrix object1, int scalar)
    {
        Matrix temp(object1.n,object1.m);
        for(int i =0; i<object1.n; i++)
            for(int j =0; j<object1.m; j++)
                temp.matr[i][j] = object1.matr[i][j] * scalar;
        return temp;
    }

    Matrix operator / (Matrix object1, int scalar)
    {
        Matrix temp(object1.n,object1.m);
        for(int i =0; i<object1.n; i++)
            for(int j =0; j<object1.m; j++)
                temp.matr[i][j] = object1.matr[i][j] / scalar;
        return temp;
    }

    Matrix Matrix:: operator =(Matrix object1)
    {
        for(int i =0; i<n; i++)
            for(int j =0; j<m; j++)
                this->matr[i][j] = object1.matr[i][j];
        return *this;
    }

    void Matrix::Enter()
    {
        cout << "Enter elements of matr: \n";
        for(int i =0; i<n; i++)
            for(int j =0; j<m; j++)
                cin>> matr[i][j];
        cout << endl << endl;
    }

    void Matrix::Show( int index_line, int index_column)
    {
        cout << "Element with number " << index_line << " " << index_column
        << " equal to " << matr [index_line][index_column] << endl << endl;
    }

    void Matrix:: Show()
    {
        for(int i =0; i<n; i++)
        {
            for(int j =0; j<m; j++)

```

- функцию-член переустановки текущего значения массива
void Set(...);
- функцию-друга, которая определяет сумму элементов главной и побочной диагонали.

Программа должна проиллюстрировать косвенный способ обращения к элементам массива. Коды функции-члена и функции-друга определите вне пространства класса.

Задание 4.2

Перегрузка операторов

Описать класс Matrix для работы с двухмерным массивом целых чисел (матрицей). Обеспечить следующие возможности:

- задание произвольных целых границ индексов при создании объекта;
- поэлементное сложение и вычитание массивов с одинаковыми границами индексов;
- умножение и деление всех элементов массива на скаляр;
- присвоение одного объекта другому;
- вывод на экран элемента массива по заданным индексам и всего массива.

Написать программу, демонстрирующую все разработанные элементы класса.

```
#include <iostream.h>
class Matrix
{
public:
    Matrix (int a, int b)
    {
        n = a;
        m = b;
        matr = new int* [n];
        for (int i = 0; i < n; i++)
            matr[i] = new int [m];
    }
    friend Matrix operator +(Matrix object1, Matrix object2); //1
    friend Matrix operator -(Matrix object1, Matrix object2); //2
    friend Matrix operator *(Matrix object1, int scalar); //3
    friend Matrix operator /(Matrix object1, int scalar); //4
    Matrix operator =(Matrix object1); //5
    void Enter(); //6
    void Show(int index_line, int index_column); //7
    void Show(); //8
private:
    int n, m, **matr; //9
};
Matrix operator + (Matrix object1, Matrix object2) //10
{
    Matrix temp(object1.n,object1.m);
    for(int i = 0; i<object1.n; i++)

```

```

        for(int j =0; j<object1.m; j++)
            temp.matr[i][j] = object1.matr[i][j] + object2.matr[i][j];
            return temp;
    }
Matrix operator - (Matrix object1, Matrix object2)           //11
{
    Matrix temp(object1.n,object1.m);
    for(int i =0; i<object1.n; i++)
        for(int j =0; j<object1.m; j++)
            temp.matr[i][j] = object1.matr[i][j] - object2.matr[i][j];
            return temp;
}
Matrix operator * (Matrix object1, int scalar)             //12
{
    Matrix temp(object1.n, object1.m);
    for(int i =0; i<object1.n; i++)
        for(int j =0; j<object1.m; j++)
            temp.matr[i][j] = object1.matr[i][j] * scalar;
            return temp;
}
Matrix operator / (Matrix object1, int scalar)           //13
{
    Matrix temp(object1.n, object1.m);
    for(int i =0; i<object1.n; i++)
        for(int j =0; j<object1.m; j++)
            temp.matr[i][j] = object1.matr[i][j] / scalar;
            return temp;
}
Matrix Matrix:: operator =(Matrix object1)                 //14
{
    for(int i =0; i<n; i++)
        for(int j =0; j<m; j++)
            this->matr[i][j] = object1.matr[i][j];
            return *this;
}
void Matrix::Enter()                                      //15
{
    cout << "Enter elements of matr: \n";
    for(int i =0; i<n; i++)
        for(int j =0; j<m; j++)
            cin >> matr[i][j];
            cout << endl << endl;
}
void Matrix::Show (int index_line, int index_column)      //16
{
    cout << "Element with number " << index_line << " " << index_column
        << " equal to " << matr [index_line][index_column] << endl << endl;
}
void Matrix:: Show()                                     //17
{
    for(int i =0; i<n; i++)
    {
        for(int j =0; j<m; j++)

```

```

        cout << matr [i][j] << " ";
        cout << endl;
    }
    cout << endl << endl;
}
int main ()
{
    Matrix matrix1(3,4), matrix2(3,4), matrix3(3,4), matrix4(3,4); //18
    matrix1.Enter(); //19
    matrix2.Enter(); //20
    matrix3 = matrix1 + matrix2; //21
    matrix3.Show(); //22
    matrix4 = matrix1 - matrix2; //23
    matrix4.Show(); //24
    matrix3 = matrix1 * 10; //25
    matrix3.Show(); //26
    matrix4 = matrix3; //27
    matrix4.Show(); //28
    matrix4 = matrix3 / 5; //29
    matrix4.Show(); //30
    matrix1.Show(2,3); //31
    matrix2.Show(0,0); //32
    return 0;
}

```

//1, //2, //3, //4 – прототипы операторов перегрузки (сложения, вычитания, умножения, деления). Перегружаемые операторы бинарные, поэтому используем функции-друзья. Функции-друзья всегда используются, когда имеется два и более аргументов.

//5 – прототип оператора перегрузки (присваивание). Перегружаемый оператор унарный, поэтому можно использовать метод, а не функцию-друга.

//6, //7, //8 – прототипы методов (ввод данных объекта, вывод на экран элемента массива по заданным индексам и вывод всего массива).

//9 – частные члены класса (количество строк и столбцов, имя матрицы). По условию задания они все целого типа.

//10, //11, //12, //13 – определения функций-друзей для перегрузки операций сложения, вычитания, умножения, деления.

//14 – определение метода для перегрузки операции присваивания. Здесь явно используется указатель this (обычно он в явном виде не используется, ибо компилятор сам ставит его в нужных местах).

//15, //16, //17 – определение методов ввода/вывода данных.

//18 – объявление четырех объектов. По условию задачи необходимо задавать целые произвольные индексы для объектов (здесь 3 и 4).

//19, //20 – вызов метода ввода данных для первого и второго объекта;

//21, //23, //25, //27, //29 – используя перегруженные операции сложения, вычитания, умножения, деления, выполняем эти действия над объектами соответственно.

//22, //24, //26, //28, //30 – вывод результатов выполненных операций на экран;

//31, //32 – вывод на экран элемента массива по заданным индексам.

Результаты выполнения задания представлены на рисунке 4.2.

```

Enter elements of matr:
3 5 2 2
3 9 3 8
4 5 8 2

Enter elements of matr:
3 9 6 2
3 6 7 2
3 5 6 9

8 14 13 4
9 15 10 10
3 10 14 11

-2 -4 1 0
3 3 -4 6
3 0 2 -2

30 50 20 20
60 90 30 80
40 50 80 20

30 50 20 20
60 90 30 80
40 50 80 20

6 10 14 4
12 18 6 16
3 10 16 4

Element with number 2 - 3 equal to 2
Element with number 0 - 0 equal to 5
Press any key to continue...

```

Рисунок 4.2 – Результаты выполнения программы задания 4.2

Индивидуальные задания

1. Описать класс для работы с одномерным массивом целых чисел (вектором). Обеспечить следующие возможности:

- задание произвольных целых границ индексов при создании объекта;
- обращение к отдельному элементу массива с контролем выхода за пределы массива;
- поэлементное сложение и вычитание массивов с одинаковыми границами индексов;
- умножение и деление всех элементов массива на скаляр;
- вывод на экран элемента массива по заданному индексу и всего массива.

Написать программу, демонстрирующую все разработанные элементы класса.

2. Описать класс для работы с одномерным массивом строк фиксированной длины. Обеспечить следующие возможности:

- задание произвольных целых границ индексов при создании объекта;
- обращение к отдельной строке массива по индексу с контролем выхода за пределы массива;

- поэлементное сцепление двух массивов с образованием нового массива;
- слияние двух массивов с исключением повторяющихся элементов;
- вывод на экран элемента массива по заданному индексу и всего массива.

Написать программу, демонстрирующую все разработанные элементы класса.

3. Описать класс многочленов от одной переменной, задаваемых степенью многочлена и массивом коэффициентов. Обеспечить следующие возможности:

- вычисление значения многочлена для заданного аргумента;
- сложение, вычитание и умножение многочленов с получением нового объекта-многочлена;
- получение многочлена заданного по индексу;
- вывод на экран описания многочлена.

Написать программу, демонстрирующую все разработанные элементы класса.

4. Описать класс, обеспечивающий представление матрицы произвольного размера с возможностью изменения количества строк и столбцов, вывода на экран подматрицы любого размера и всей матрицы, доступа по индексам к элементу матрицы.

Написать программу, демонстрирующую все разработанные элементы класса.

5. Описать класс для работы с восьмеричным числом, хранящимся в виде строки символов. Реализовать конструкторы, методы и следующие операции:

- присваивание, реализующее значимую семантику;
- сравнение;
- преобразование в десятичное число;
- форматный вывод;
- доступ к заданной цифре числа по индексу.

Написать программу, демонстрирующую все разработанные элементы класса.

6. Описать класс «домашняя библиотека». Предусмотреть возможность работы с произвольным количеством книг, поиска книги по какому-либо признаку (по автору, по году издания или категории), добавление книг в библиотеку, удаления книг из нее, доступа к книге по номеру.

Написать программу, демонстрирующую все разработанные элементы класса.

7. Описать класс «домашняя библиотека». Предусмотреть возможность работы с произвольным количеством книг, поиска книги по какому-либо признаку (по автору, по году издания или категории), добавления книг в библиотеку, удаления книг из нее, доступа к книге по номеру.

Написать программу, демонстрирующую все разработанные элементы класса.

8. Описать класс «записная книжка». Предусмотреть возможности работы с произвольным числом записей, поиска записи по какому-либо признаку (на-

пример, по фамилии, дате рождения или номеру телефона), добавление и удаление записей, сортировки по фамилии и доступа к записям по номеру.

Написать программу, демонстрирующую все разработанные элементы класса.

9. Описать класс «записная книжка». Предусмотреть возможности работы с произвольным числом записей, поиска записи по какому-либо признаку (например, по фамилии, дате рождения или номеру телефона), добавления и удаления записей, сортировки по фамилии и доступа к записям по номеру.

Написать программу, демонстрирующую все разработанные элементы класса.

10. Описать класс, реализующий тип данных «вещественная матрица» и работу с ними. Класс должен реализовать следующие операции над матрицами:

- сложение, вычитание (как с другой матрицей, так и с числом);
- комбинированные операции присваивания ($+=$, $-=$);
- сравнение на равенство / неравенство;
- вычисление обратной и транспонированной матрицы;
- доступ к элементу по индексам.

Написать программу, демонстрирующую все разработанные элементы класса.

11. Описать класс, реализующий тип данных «вещественная матрица» и работу с ними. Класс должен реализовать следующие операции над матрицами:

- умножение, деление (как на другую матрицу, так и на число);
- комбинированные операции присваивания ($*=$, $/=$);
- возведение в степень;
- доступ к элементу по индексам.

Написать программу, демонстрирующую все разработанные элементы класса.

12. Описать класс, реализующий тип данных «вещественная матрица» и работу с ними. Класс должен реализовать следующие операции над матрицами:

- методы, реализующие проверку типа матрицы (квадратная, диагональная, нулевая, единичная);
- сравнение на равенство / неравенство;
- доступ к элементу по индексам.

Написать программу, демонстрирующую все разработанные элементы класса.

13. Описать класс «множество», позволяющий выполнять основные операции: добавление и удаление элемента, пересечение, объединение и разность множеств.

Написать программу, демонстрирующую все разработанные элементы класса.

14. Описать класс «автостоянка» для хранения сведений об автомобилях. Для каждого автомобиля записываются госномер, цвет, фамилия владельца и признак присутствия на стоянке. Обеспечить возможность поиска автомобиля по разным критериям, вывода списка присутствующих и отсутствующих на стоянке автомобилей, доступа к имеющимся сведениям по номеру места.

Написать программу, демонстрирующую все разработанные элементы класса.

15. Описать класс «колода карт», включающий закрытый массив элементов класса «карта». В карте хранятся масть и номер. Обеспечить возможность вывода карты по номеру, вывода всех карт, перемешивания колоды и выдачи всех карт из колоды по одной и по 6 штук, в случайном порядке.

Написать программу, демонстрирующую все разработанные элементы класса.

Лабораторная работа №5

НАСЛЕДОВАНИЕ. ВИРТУАЛЬНЫЕ ФУНКЦИИ

Задание 5.1

Наследование

Реализовать иерархию классов X->Y. В классе X присутствуют данные x1, x2. В классе Y-дополнение у. Методами классов представить конструктор, методы ввода-вывода данных. В производном классе Y присутствует метод Run(), который определяет: сумму всех данных класса.

```
#include <iostream.h>
class X //1
{
public:
    X(int a, int b) //2
    {
        x1 = a;
        x2 = b;
    }
    void Enter() //3
    {
        cout << "\nEnter x1 and x2:\n";
        cin >> x1 >> x2;
    }
    void Show() //4
    {
        cout << "\nx1 = " << x1 << " x2 = " << x2 << endl << endl;
    }
protected: //5
    int x1, x2; //6
};

class Y: public X //7
{
public:
    Y(int a, int b, int c):X(a, b) //8
    {
        y = c;
    }
    void Run() //9
    {
        cout << "\nSum x1, x2, y = " << x1 + x2 + y << endl << endl;
    }
protected: //10
    int y;
};

int main()
{
    Y object_y(5, 18, 28); //11
```

```
object_y.Run(); //12
object_y.Enter(); //13
object_y.Run(); //14
object_y.Show(); //15
return 0;
}
```

//1 – описание базового класса X (ключевое слово class, имя класса X).

//2 – определение конструктора базового класса, методы можно определять в классе, если они определяются небольшим количеством инструкций; поскольку при определении метода в классе компилятор подставляет код метода не в ходе выполнения программы, а при компиляции, такое определение методов называется Inline-функциями (т. е. они компилируются вместе с основной программой).

//3 – определение метода ввода данных базового класса с клавиатуры.

//4 – определение метода вывода данных класса на экран.

//5 – данные объявляются со спецификатором protected. Так как они будут участвовать в наследовании, т.е. их значения будут использоваться в производном классе.

//6 – объявление данных базового класса, ибо в условии задачи не указан явно тип данных, мы использовали целый тип (int).

//7 – описание производного класса: после двоеточия пишется сначала спецификатор доступа к данным членам и методам класса (public), потом – имя базового класса.

//8 – определение конструктора производного класса, поскольку данные x1 и x2 были определены в конструкторе базового класса, мы явным образом описываем инициализацию базового класса (:X(a,b)), а переменную производного класса инициализируем значением переменной int с.

//9 – метод вычисления суммы переменных базового и производного классов.

//10 – спецификатор protected, в данном классе можно было бы использовать вместо него спецификатор private, так как производный класс Y не наследуется другим классом.

//11 – объявление объекта класса Y: инициализируем данные базового класса, в данном случае – x1=5, x2=18, и данные производного класса, в данном случае – y=28.

//12 – вызов метода, вычисляющего сумму всех данных класса.

//13 – вызов метода ввода данных базового класса.

//14 – вызов метода, вычисляющего сумму всех данных класса с новыми значениями x1 и x2.

//15 – вызов метода вывода данных базового класса на экран.

Результаты выполнения программы представлены на рисунке 5.1.

```
Sum x1, x2, y = 51
Enter x1 and x2:
54 57
Sum x1, x2, y = 139
x1 = 54 x2 = 57
Press any key to continue.
```

Рисунок 5.1 – Результаты выполнения программы задания 5.1

Индивидуальные задания

Реализовать иерархию классов X->Y. В классе X присутствуют данные x1, x2. В классе Y – данное у. Методами классов представить конструктор: методы ввода/вывода данных. В производном классе Y присутствует метод Run(), который определяет:

1. Произведение всех данных класса.
2. Сумму квадратов всех данных класса.
3. Значение $x1 + x2 - y$
4. Значение $(x1 + x2)/y$
5. Значение $(x1 + x2)*y$
6. Значение $x1*y + x2$
7. Значение $x1+x2*y$
8. Произведение квадратов всех данных класса
9. Значение $x1*x2 + y$
10. Значение $x1*x2 / y$
11. Значение $x1*x2 - y$
12. Значение $(x1 - x2)/y$
13. Значение $(x1 - y)* x2$
14. Значение $x1/y - x2*y$
15. Значение $x1-x2*(cos(y))$

Задание 5.2

Виртуальные функции

В этом задании рассмотрим разницу выполнения программ с ключевым словом `virtual` и этой же программы без ключевого слова `virtual`. Результаты выполнения программ показаны на рисунках 5.2 и 5.3.

Рассмотрим класс:

```
class base
{
public:
    virtual void aim() { cout << "base\n"; }
};

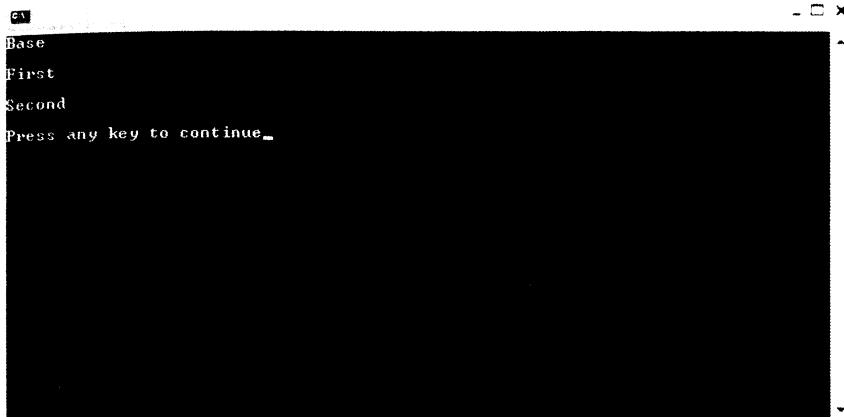
a) Создайте два производных класса от base класса и определите в каждом из них функцию aim(), выводящую имя класса.
б) Объявите объект каждого класса и вызовите его функцию aim().
в) Присвойте адрес объекта производного класса указателю на базовый класс и вызовите функцию aim(), используя этот указатель.
г) Удалите слово virtual из определения базового класса, выполните ваши код еще раз и сравните полученные результаты.
```

```
#include <iostream.h>
class Base
{
public:
    virtual void aim() { cout << "Base\n\n"; }
};

class First: public Base
{
public:
    void aim() { cout << "First\n\n"; }
};

class Second: public Base
{
public:
    void aim() { cout << "Second\n\n"; }
};

int main()
{
    Base base, *p;
    First first;
    Second second;
    p = &base;
    p -> aim();
    p = &first;
    p -> aim();
    p = &second;
    p -> aim();
    return 0;
}
```



```
Base
First
Second
Press any key to continue
```

Рисунок 5.2 – Результаты выполнения программы задания 5.2 с ключевым словом **virtual** в базовом классе



```
Base
Base
Base
Press any key to continue
```

Рисунок 5.3 – Результаты выполнения программы задания 5.2 без ключевого слова **virtual** в базовом классе

Задание 5.3

Напишите программу, которая должна содержать:

- базовый класс, включающий:

- а) два элемента в виде строк символов S1 и S2;*
- б) конструктор с параметрами для создания динамических строковых объектов;*
- в) деструктор;*
- г) виртуальный метод просмотра текущего состояния объекта;*

```

{
    cout << "S1 = " << S1 << endl
    << "S2 = " << S2 << endl << endl;
}
void Line::Show()
{
    cout << "S1 = " << S1 << endl
    << "S2 = " << S2 << endl
    << "S3 = " << S3 << endl << endl;
}
void Line::Run()
{
    int i, l1, l2, l3, count;
    bool f1 = false, f2 = false, f3 = false;
    l1 = strlen (S1);
    l2 = strlen (S2);
    l3 = strlen (S3);
    count = 0;
    for (i = 0; i < l1; i++)
        if (isdigit (S1[i]))
            count++;
    if (count == l1)
        {cout << "Number line S1 = " << atoi (S1) << endl;
        f1 = true;
    }
    count = 0;
    for (i = 0; i < l2; i++)
        if (isdigit (S2[i]))
            count++;
    if (count == l2)
        {cout << "Number line S2 = " << atoi (S2) << endl;
        f2 = true;
    }
    count = 0;
    for (i = 0; i < l3; i++)
        if (isdigit (S3[i]))
            count++;
    if (count == l3)
        {cout << "Number line S3 = " << atoi (S3) << endl;
        f3 = true;
    }
    if (!f1 && !f2 && !f3) cout << "The same as lines are absent\n\n";
}
int main ()
{
    Base base("This is a Base.", "This is second line."), *p;
    Line Object1 ("Hello", "World!", "Happy New Year!", "Tastes differ."), Ob-
    ject2("12345", "Apple", "Limon");
    p = &base;
    p->Show();
    p = &Object1;
    p->Show();
    Object1.Run();
    p = &Object2;
}
//15
//16
//17
//18
//19
//20
//21
//22
//23
//24
//25
//26

```

- производный класс, включающий:

- a) элемент в виде строки символов S3;
- б) конструктор с параметрами для создания динамических строковых объектов, передающий данные конструктору базового класса;
- в) переопределенный метод просмотра текущего состояния объектов;

г) метод, реализующий следующую задачу:

если выясняется, что хотя бы одна из строк S1, S2 и S3 только из цифр, если да, то надо преобразовать такую строку в число;

- функцию main, в которой создаются объекты и указатели на них, демонстрируется работа методов классов.

```
#include <iostream.h>
#include <string.h>
#include <ctype.h>
#include <stdlib.h>
class Base
{
public:
    Base (char *str1, char *str2); //1
    ~Base (); //2
    virtual void Show(); //3
protected:
    char *S1, *S2; //4
};
class Line: public Base //5
{
public:
    Line (char *str1, char *str2, char *str3):Base(str1, str2) //6
    {
        S3 = new char [strlen(str3) + 1]; //7
        strcpy(S3, str3); //8
    }
    void Show (); //9
    void Run (); //10
private:
    char *S3; //11
};
Base::Base(char *str1, char *str2) //12
{
    S1 = new char [strlen(str1) + 1];
    S2 = new char [strlen(str2) + 1];
    strcpy(S1, str1);
    strcpy(S2, str2);
}
Base::~Base () //13
{
    delete []S1;
    delete []S2;
}
void Base::Show() //14
```

```
p -> Show(); //27  
Object2.Run(); //28  
return 0;  
}
```

//1 – прототип конструктора.

//2 – прототип деструктора.

//3 – прототип метода просмотра текущего состояния объекта. Поскольку нам надо переопределить этот метод в производном классе, мы объявляем его с ключевым словом `virtual`.

//4 – исходные данные базового класса, который наследуется производным классом, поэтому строковые переменные мы объявляем не частными (`private`), а защищенными (`protected`), ибо частные члены не участвуют в наследовании.

//5 – производный класс (`Line`) наследует базовый класс (`Base`) со спецификатором `public`.

//6 – определение конструктора `Line`: так как строковые переменные `S1`, `S2` были определены в базовом классе, мы явным образом сообщаем об этом компилятору (`:Base(str1, str2)`).

//7 – выделяем в `heap` области память для строковой переменной `S3`, используя оператор `new` и функцию `strlen` (она вычисляет длину строки, без учета ноль символа и находится в библиотеке `string`), поскольку для `S3` необходим ноль символ, для него мы добавляем единицу (+1).

//8 – с помощью функции копирования строковых переменных (`strcpy`), копируем инициализирующую строку `str3` в строковую переменную `S3`.

//9 – прототип метода `Show()`; с таким же интерфейсом был описан метод `Show()` базового класса – это одна из особенностей виртуальных функций: один интерфейс – множество определений. Ключевое слово `virtual` достаточно описать в базовом классе, и на протяжении всей иерархии описанный метод, будет виртуальным.

//10 – прототип метода `Run()`, в котором будут выполняться требования задачи.

//11 – объявление строковой переменной производного класса.

//12 – определение конструктора базового класса вне класса (конструктор производного класса мы определили в классе).

//13 – определение деструктора базового класса.

//14 – определение виртуального метода базового класса.

//15 – определение виртуального метода производного класса.

//16 – определение метода `Run()`, согласно которому требуется проверять состоят ли строки из цифр, и если да необходимо перевести каждую строку в целое число.

//17 – проверка строковых элементов с помощью функции `isdigit()` (библиотека `cctype`) – это цифра или другой символ.

//18 – функция `atoi()` (библиотека `stdlib`) переводит строковую переменную в число.

//19 – объявление объекта базового класса `Base`. В круглых скобках инициализируем с помощью конструктора строковые переменные `S1`, `S2`. Указатель объявляется всегда только в базовом классе.

//20 – объявляем и инициализируем два объекта производного класса Line.
//21 – переменной указатель: присваиваем ссылку на адрес объекта base
(благодаря этой ссылке компилятор будет знать, именно в точке вызова, код какой виртуальной функции подставить).
//22 – вызов виртуального метода просмотра текущего состояния объекта из базового класса.
//23 – указателю присваивается ссылка на адрес объекта Object1;
//24 - вызов виртуального метода просмотра текущего состояния первого объекта из производного класса.
//25 – вызов метода Run() для первого объекта. При выполнении этого метода произойдет выяснение того сколько цифр содержится в каждой строке.
//26 – указателю присваивается ссылка на адрес объекта Object2.
//27 – вызов виртуального метода просмотра текущего состояния второго объекта из производного класса.
//28 – вызов метода Run() для второго объекта. При выполнении этого метода устанавливается, сколько цифр содержится в строке.

Результаты выполнения программы представлены на рисунке 5.4.

```
D:\C++> D:\C++\Line.cpp
S1 = This is a Base.
S2 = This is second line.

S1 = Hello World!
S2 = Happy New Year!
S3 = Letters differ.

The same as Lines are absent

S1 = 12346
S2 = Apple
S3 = Linon

Number Line S1 = 12346
Press any key to continue
```

Рисунок 5.4 – Результаты выполнения программы задания 5.3

Индивидуальные задания

1. Напишите программу, которая должна содержать:
 - базовый класс, включающий:
 - а) два элемента в виде строк символов *S1* и *S2*;
 - б) конструктор с параметрами для создания динамических строковых объектов;
 - в) деструктор;
 - г) виртуальный метод просмотра текущего состояния объекта;
 - производный класс, включающий:
 - а) элемент в виде строки символов *S3*;

- б) конструктор с параметрами для создания динамических строковых объектов, передающий данные конструктору базового класса;
- в) переопределенный метод просмотра текущего состояния объектов;
- г) метод, реализующий следующую задачу:

в самой длинной строке из S1, S2 и S3 (считайте, что такая строка одна) найдите и замените определенный символ в строке (заменяемый и заменяющий символы вводятся с клавиатуры);

- функцию main, в которой создаются объекты и указатели на них, демонстрируется работа методов классов.

- 2. Напишите программу, которая должна содержать:

- базовый класс, включающий:

- а) два элемента в виде строк символов S1 и S2;

б) конструктор с параметрами для создания динамических строковых объектов;

- в) деструктор;

г) виртуальный метод просмотра текущего состояния объекта;

- производный класс, включающий:

- а) элемент в виде строки символов S3;

б) конструктор с параметрами для создания динамических строковых объектов, передающий данные конструктору базового класса;

в) переопределенный метод просмотра текущего состояния объектов;

- г) метод, реализующий следующую задачу:

определите, состоит ли хотя бы одна из строк S1, S2 и S3 только из строчных букв латинского алфавита (а...z), если да, то отсортируйте символы в этой строке по алфавиту в порядке возрастания;

- функцию main, в которой создаются объекты и указатели на них, демонстрируется работа методов классов.

- 3. Напишите программу, которая должна содержать:

- базовый класс, включающий:

- а) два элемента в виде строк символов S1 и S2;

б) конструктор с параметрами для создания динамических строковых объектов;

- в) деструктор;

г) виртуальный метод просмотра текущего состояния объекта;

- производный класс, включающий:

- а) элемент в виде строки символов S3;

б) конструктор с параметрами для создания динамических строковых объектов, передающий данные конструктору базового класса;

в) переопределенный метод просмотра текущего состояния объектов;

- г) метод, реализующий следующую задачу:

зашифруйте строки заменой исходных символов на символы с кодом большим на 3;

- функцию main, в которой создаются объекты и указатели на них, демонстрируется работа методов классов.

- 4. Напишите программу, которая должна содержать:

- базовый класс, включающий:

- а) два элемента в виде строк символов *S1* и *S2*;
 - б) конструктор с параметрами для создания динамических строковых объектов;
 - в) деструктор;
 - г) виртуальный метод просмотра текущего состояния объекта;
- производный класс, включающий:
- а) элемент в виде строки символов *S3*;
 - б) конструктор с параметрами для создания динамических строковых объектов, передающий данные конструктору базового класса;
 - в) переопределенный метод просмотра текущего состояния объектов;
 - г) метод, реализующий следующую задачу:

определите, содержат ли строки S1, S2 и S3 цифры;

- функцию *main*, в которой создаются объекты и указатели на них, демонстрируется работа методов классов.

5. Напишите программу, которая должна содержать:

- базовый класс, включающий:

- а) два элемента в виде строк символов *S1* и *S2*;

б) конструктор с параметрами для создания динамических строковых объектов;

в) деструктор;

г) виртуальный метод просмотра текущего состояния объекта;

- производный класс, включающий:

- а) элемент в виде строки символов *S3*;

б) конструктор с параметрами для создания динамических строковых объектов, передающий данные конструктору базового класса;

в) переопределенный метод просмотра текущего состояния объектов;

г) метод, реализующий следующую задачу:

определите, какая из строк S1, S2 и S3 содержит самое длинное слово (считайте, что такое слово одно);

- функцию *main*, в которой создаются объекты и указатели на них, демонстрируется работа методов классов.

6. Напишите программу, которая должна содержать:

- базовый класс, включающий:

- а) два элемента в виде строк символов *S1* и *S2*;

б) конструктор с параметрами для создания динамических строковых объектов;

в) деструктор;

г) виртуальный метод просмотра текущего состояния объекта;

- производный класс, включающий:

- а) элемент в виде строки символов *S3*;

б) конструктор с параметрами для создания динамических строковых объектов, передающий данные конструктору базового класса;

в) переопределенный метод просмотра текущего состояния объектов;

г) метод, реализующий следующую задачу:

в строках (S1, S2 и S3) с четной длиной поменяйте местами символы на четных и нечетных позициях;

- функцию `main`, в которой создаются объекты и указатели на них, демонстрируется работа методов классов.

7. Напишите программу, которая должна содержать:

- базовый класс, включающий:

а) два элемента в виде строк символов `S1` и `S2`;

б) конструктор с параметрами для создания динамических строковых объектов;

в) деструктор;

г) виртуальный метод просмотра текущего состояния объекта;

- производный класс, включающий:

а) элемент в виде строки символов `S3`;

б) конструктор с параметрами для создания динамических строковых объектов, передающий данные конструктору базового класса;

в) переопределенный метод просмотра текущего состояния объектов;

г) метод, реализующий следующую задачу:

в строках (`S1`, `S2` и `S3`) с нечетной длиной удалите центральный элемент;

- функцию `main`, в которой создаются объекты и указатели на них, демонстрируется работа методов классов.

8. Напишите программу, которая должна содержать:

- базовый класс, включающий:

а) два элемента в виде строк символов `S1` и `S2`;

б) конструктор с параметрами для создания динамических строковых объектов;

в) деструктор;

г) виртуальный метод просмотра текущего состояния объекта;

- производный класс, включающий:

а) элемент в виде строки символов `S3`;

б) конструктор с параметрами для создания динамических строковых объектов, передающий данные конструктору базового класса;

в) переопределенный метод просмотра текущего состояния объектов;

г) метод, реализующий следующую задачу:

определите количество слов в каждой из строк `S1`, `S2` и `S3`;

- функцию `main`, в которой создаются объекты и указатели на них, демонстрируется работа методов классов.

9. Напишите программу, которая должна содержать:

- базовый класс, включающий:

а) два элемента в виде строк символов `S1` и `S2`;

б) конструктор с параметрами для создания динамических строковых объектов;

в) деструктор;

г) виртуальный метод просмотра текущего состояния объекта;

- производный класс, включающий:

а) элемент в виде строки символов `S3`;

б) конструктор с параметрами для создания динамических строковых объектов, передающий данные конструктору базового класса;

в) переопределенный метод просмотра текущего состояния объектов;

г) метод, реализующий следующую задачу:

в самой длинной строке (считайте, что такая строка одна) из S1, S2 и S3 поменяйте местами первый и последний символы;

- функцию main, в которой создаются объекты и указатели на них, демонстрируется работа методов классов.

10. Напишите программу, которая должна содержать:

- базовый класс, включающий:

а) два элемента в виде строк символов S1 и S2;

б) конструктор с параметрами для создания динамических строковых объектов;

в) деструктор;

г) виртуальный метод просмотра текущего состояния объекта;

- производный класс, включающий:

а) элемент в виде строки символов S3;

б) конструктор с параметрами для создания динамических строковых объектов, передающий данные конструктору базового класса;

в) переопределенный метод просмотра текущего состояния объектов;

г) метод, реализующий следующую задачу:

определите номера позиций запрашиваемого символа (вводится с клавиатуры) в каждой из строк S1, S2 и S3;

- функцию main, в которой создаются объекты и указатели на них, демонстрируется работа методов классов.

11. Напишите программу, которая должна содержать:

- базовый класс, включающий:

- два элемента в виде строк символов S1 и S2;

- конструктор с параметрами для создания динамических строковых объектов;

- деструктор;

- виртуальный метод просмотра текущего состояния объекта;

- производный класс, включающий:

- элемент в виде строки символов S3;

- конструктор с параметрами для создания динамических строковых объектов, передающий данные конструктору базового класса;

- переопределенный метод просмотра текущего состояния объектов;

- метод, реализующий следующую задачу:

определите длину самой короткой строки из S1, S2 и S3 (считайте, что такая строка одна) и если она больше 4, то удалите из строки два первых и два последних элемента;

- функцию main, в которой создаются объекты и указатели на них, демонстрируется работа методов классов.

12. Напишите программу, которая должна содержать:

- базовый класс, включающий:

а) два элемента в виде строк символов S1 и S2;

б) конструктор с параметрами для создания динамических строковых объектов;

в) деструктор;

г) виртуальный метод просмотра текущего состояния объекта;

- производный класс, включающий:

а) элемент в виде строки символов *S3*;

б) конструктор с параметрами для создания динамических строковых объектов, передающий данные конструктору базового класса;

в) переопределенный метод просмотра текущего состояния объектов;

г) метод, реализующий следующую задачу:

зашифруйте строки с четной длиной, поменяв местами первый символ с третьим, второй с четвертым и т.д.;

- функцию *main*, в которой создаются объекты и указатели на них, демонстрируется работа методов классов.

13. Напишите программу, которая должна содержать:

- базовый класс, включающий:

а) два элемента в виде строк символов *S1* и *S2*;

б) конструктор с параметрами для создания динамических строковых объектов;

в) деструктор;

г) виртуальный метод просмотра текущего состояния объекта;

- производный класс, включающий:

а) элемент в виде строки символов *S3*;

б) конструктор с параметрами для создания динамических строковых объектов, передающий данные конструктору базового класса;

в) переопределенный метод просмотра текущего состояния объектов;

г) метод, реализующий следующую задачу:

определите суммарное количество повторений запрашиваемого символа (вводится с клавиатуры) в строках S1, S2 и S3.

- функцию *main*, в которой создаются объекты и указатели на них, демонстрируется работа методов классов.

14. Напишите программу, которая должна содержать:

- базовый класс, включающий:

а) два элемента в виде строк символов *S1* и *S2*;

б) конструктор с параметрами для создания динамических строковых объектов;

в) деструктор;

г) виртуальный метод просмотра текущего состояния объекта;

- производный класс, включающий:

а) элемент в виде строки символов *S3*;

б) конструктор с параметрами для создания динамических строковых объектов, передающий данные конструктору базового класса;

в) переопределенный метод просмотра текущего состояния объектов;

г) метод, реализующий следующую задачу:

*в строках S1, S2 и S3 с четной длиной замените каждый второй символ на *;*

- функцию *main*, в которой создаются объекты и указатели на них, демонстрируется работа методов классов.

15. Напишите программу, которая должна содержать:

- базовый класс, включающий:

- а) два элемента в виде строк символов $S1$ и $S2$;
- б) конструктор с параметрами для создания динамических строковых объектов;
- в) деструктор;
- г) виртуальный метод просмотра текущего состояния объекта;
- производный класс, включающий:
- а) элемент в виде строки символов $S3$;
- б) конструктор с параметрами для создания динамических строковых объектов, передающий данные конструктору базового класса;
- в) переопределенный метод просмотра текущего состояния объектов;
- г) метод, реализующий следующую задачу:
- определите, содержат ли строки $S1$, $S2$ и $S3$ пробелы;*
- функцию `main`, в которой создаются объекты и указатели на них, демонстрируется работа методов классов.