

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Макаренко Елена Николаевна

Должность: Ректор

Дата подписания: 09.04.2021 18:44:50

Уникальный программный ключ:

c098bc0c1041cb2a4cf926cf171d6715d99a6ae00adc8e27b55cbe1e2dbd7c78

Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Ростовский государственный экономический университет (РИНХ)»

Кафедра «Информационных систем и прикладной информатики»

**И.И. Мирошниченко, Е.Г. Веретенникова,
Н.Г. Савельева, В.Ю. Гречкина**

Методические рекомендации для выполнения самостоятельной работы по дисциплинам «Информатика и программирование», «Информатика»

*Для студентов 1, 2 курса
очной, заочной, очно-заочной формы обучения*

**Направления 09.03.02 Информационные системы и технологии,
09.03.03 Прикладная информатика,
10.03.01 «Информационная безопасность»**

Ростов-на-Дону
2019

УДК005(072.8)
ББК65.29
М64

Рецензенты:

канд. эконом. наук, доц. кафедры Информационных технологий
и защиты информации РГЭУ (РИНХ) ***Е.В. Жилина***,
канд. эконом. наук, доц. кафедры Информационных систем и прикладной
информатики РГЭУ (РИНХ) ***К.Х. Калугян***

Мирошниченко И.И., Веретенникова Е.Г., Савельева Н.Г., Гречкина В.Ю.

М64 Методические рекомендации для выполнения самостоятельной работы по дисциплинам «Информатика и программирование», «Информатика»: метод. рек. [Электронный ресурс]. – Ростов н/Д.: Издательско-полиграфический комплекс РГЭУ (РИНХ), 2019. – 96 с.– Режим доступа: <http://library.rsue.ru>.

В методических рекомендациях представлены технология выполнения, примеры задач и их решение для направлений подготовки «Прикладная информатика», «Информационные системы и технологии» и «Информационная безопасность» всех форм обучения. Могут использоваться для иных направлений и специальностей.

Предназначается для самостоятельной подготовки студентов к практическим и лабораторным занятиям по дисциплинам «Информатика и программирование» и «Информатика», преподавателей, а также для всех интересующихся актуальными проблемами программирования на языке C++.

УДК005(072.8)
ББК65.29

© Мирошниченко И.И., Веретенникова Е.Г.,
Савельева Н.Г., Гречкина В.Ю., 2019
© РГЭУ (РИНХ), 2019

Оглавление

Задания для самостоятельной работы.....	4
К лабораторной работе № 1. Вывод текста. Линейные программы.....	4
К лабораторной работе № 2. Простые конструкции языка.	
Программы с ветвлением, циклами.....	8
К лабораторной работе № 3. Вычисление суммы и произведения по заданной формуле.....	12
К лабораторной работе № 4. Линейные статические массивы.....	16
К лабораторной работе № 5. Двумерные статические массивы.....	22
К лабораторной работе № 6. Указатели.....	29
К лабораторной работе № 7. Строки.....	33
К лабораторной работе № 8. Линейные динамические массивы.....	39
К лабораторной работе № 9. Двумерные динамические массивы.....	43
К лабораторной работе № 10. Функции.....	48
К лабораторной работе № 11. Структуры.....	54
К лабораторной работе № 12. Файлы.....	62
К лабораторной работе № 13. Классы.....	75
К лабораторной работе № 14. Классы. Наследование.....	85
Список использованных и рекомендуемых источников.....	93
Приложение	
Форма титульного листа для отчета по самостоятельным работам.....	95

Задания для самостоятельной работы

К лабораторной работе № 1. Вывод текста. Линейные программы

Пример 1. Составить программу вывода текста на экран.

Порядок выполнения:

1. Пуск, программы, Visual Studio 2015, Создать проект, Visual C++, Консольное приложение Win32, имя (задать имя проекта), расположение (диск, каталог), имя решения (появится автоматически), ОК, консольное приложение, пустой проект, готово.

2. Проект (пункт меню), добавить новый элемент, файл C++ (.cpp), имя (ввести имя программы), добавить.

3. В папке Файлы исходного кода появился файл с введенным именем (имя.cpp). Набрать текст в редакторе. Выполнить: Проект, Построить решение.

4. Отладка программы. Отладка (пункт меню), начать отладку, выполнить построение, ДА. Повторное открытие проекта осуществить с помощью функции Открыть проект.

5. Ввод программного кода:

#include <iostream> // подключение стандартной библиотеки для базовых операций ввода вывода

using namespace std; // пространство имен

int main() // главная функция – точка запуска любого проекта

{

setlocale(LC_ALL, "Russian"); // использование русского алфавита

cout << "Здравствуй, мир!" << endl; // вывод сообщения

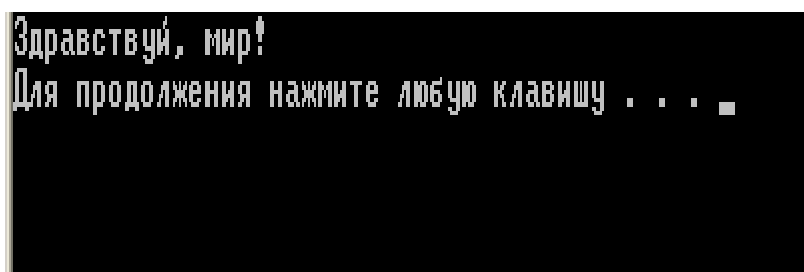
system("pause"); // задержка экрана

return 0; // конец главной функции

}

6. Построение проекта: после ввода программного кода нужно скомпилировать и отладить программу. Для этого: Построение – Компилировать (Ctrl + F7) – Программа скомпилирована. Если в программном коде имеются ошибки, они отобразятся в нижней части окна. Чтобы увидеть, в каком месте ошибка, нужно дважды кликнуть на сообщении об ошибке. Синий маркер укажет на строку в программном коде, где имеется ошибка. Далее создаем исполняемый exe-файл: Построение – Построить решение (F7). Чтобы просмотреть результат выполнения программы, нужно выполнить: Отладка – Запуск без отладки (Ctrl+F5).

Результат выполнения программы:



Задания для самостоятельного выполнения:

Задание 1.1. Вывести на экран, следующий фрагмент текста:

1. Каждый достойный называться человеком, должен иметь охоту и способность к труду.
2. Каждый из нас сам вычеканивает цену своей личности; человек бывает велик или мал в зависимости от собственной воли.
3. Книги вводят нас в лучшее общество, знакомят нас с величайшими умами всех времен.
4. Книги – лучшие товарищи от старости, в то же время они - лучшие руководители юности.
5. Книги обладают способностью бессмертия. Они самые долговечные плоды человеческой деятельности.

6. Красноречие в поступках несравненно лучше красноречия на словах.
7. Самоуважение – есть нравственное здоровье.
8. Трудолюбие – душа всякого дела и залог благосостояния.
9. Нет еще на свете такого холма, вершины которого настойчивость в конце концов не достигнет.
10. Всякое достоинство, всякая сила спокойны именно потому, что уверены в самих себя.
11. Преодолевши, какой ни есть труд, человек чувствует удовольствие.
12. Все радости и несчастья людей созданы их собственными мыслями.
13. Мысль – цветок, слово – завязь, деяние – плод.
14. Сколько в человеке доброты, столько в нем и жизни.
15. Нет победителя сильнее того, кто сумел победить самого себя.
16. Люди живут поступками, а не идеями.
17. Чтобы переваривать знания, надо поглощать их с аппетитом.
18. В жизни нет ничего лучше собственного опыта.
19. Работа избавляет нас от трех великих зол: скуки, порока, нужды
20. Почти во всех делах – самое трудное начало.
21. Что посеешь в юности, то пожнешь в старости.

Задание 1.2. Линейные программы

1. Вычислить значение следующего выражения:

$$y = \frac{2x - 3^x}{\sqrt{4x^2 - 3x + 1}} + \frac{3^{x+1} + 3^{x-2} + 2}{\sin^2(2 + x^2)}$$

2. Вычислить значение следующего выражения:

$$y = \cos(4^x + 2^{x+1} - 8x) + \sin \frac{2}{\sqrt{x+2}}$$

3. Вычислить значение следующего выражения:

$$y = 2\sin^2(3\pi - 2x)\cos^2(5\pi + 2x)$$

4. Вычислить значение следующего выражения:

$$y = 2\sqrt{2} \cos x \cdot \sin\left(\frac{\pi}{4} + 2x\right)$$

5. Вычислить значение следующего выражения:

$$y = \cos^2\left(\frac{3}{8}\pi - \frac{x}{4}\right) - \cos^2\left(\frac{11}{8}\pi + \frac{x}{4}\right)$$

$$u = \frac{xyz - 3,3|x + \sqrt[4]{y}|}{10^7 + \sqrt{x}};$$

6. Вычислить значение выражения:

7. Вычислить значение следующего выражения:

$$r = \frac{\sin x + \cos 0,5x + \sin x - x^2}{1}.$$

$$Y = \frac{\sin 2x \cos x - 2,4 \ln x - \lg x}{\sin x + \frac{x^2 \cos x}{|x|}};$$

Вычислить значение следующего выражения:

8. Вычислить среднее арифметическое чисел x , y , z ;

9. Вычислить длину окружности, площадь круга и объем шара одного и того же заданного радиуса;

10. Вычислить периметр и площадь прямоугольного треугольника по длинам двух катетов.

11. Вычислить разность, сумму и произведение двух введенных с клавиатуры чисел.

12. Вычислить значение выражения: $y = (a + b)^2$, $z = (a - b)^2$, $r = (a \times b)^2$

13. Вычислить значение стоимости товара, если известно количество купленного товара и стоимость единицы товара.

14. Вычислить значение оплаты за электроснабжение, если известно количество расхода электроэнергии и тариф оплаты:

15. Вычислить значение оплаты за использование природного газа, если известно объем расхода газа и цена.

16. Вычислить общую сумму оплаты за коммунальные услуги, если известны поступившие данные об оплате за поставленный газ, израсходованную воду, потребленную электроэнергию

17. Вычислить общую сумму затрат на покупки, если был приобретен следующий товар: молоко – 3 бутылки по цене 58 руб.,60 коп.; йогурт – 5 стаканов по цене 38 руб. 20 коп.; творог – 5 пачек по цене 49руб. 60 коп.

18. Вычислить общую сумму затрат на покупки, если был приобретен следующий товар: паштет печеночный – 5 банок по цене 48 руб.,50 коп.; консервы рыбные – 5 банок по цене 82 руб. 20 коп.; печенье – 5 пачек по цене 70руб.00 коп. Скидка составила на печенье 10% по акции:

19. Вычислить сумму подоходного налога 13% от начисленной заработной платы и сумму заработной платы, выдаваемой на руки сотрудникам после вычета всех удержаний.

К лабораторной работе № 2. Простые конструкции языка.

Программы с ветвлением, циклами

Пример 2.1. Составить программу определения значений переменной Y по одной из формул в зависимости от величины X:

$$y = \begin{cases} x+1, & \text{если } x < -3; \\ 2x+1, & \text{если } -3 \leq x < 2; \\ 3x, & \text{если } x \geq 2. \end{cases}$$

Порядок выполнения:

1. Пуск, программы, Visual Studio 2015, Создать проект, Visual C++, Консольное приложение Win32, имя (задать имя проекта), расположение (диск, каталог), имя решения (появится автоматически), ОК, консольное приложение, пустой проект, готово.

2. Проект (пункт меню), добавить новый элемент, файл C++ (.cpp), имя (ввести имя программы), добавить.

3. В папке Файлы исходного кода появился файл с введенным именем (имя.cpp). Набрать текст в редакторе. Выполнить: Проект, Построить решение.

4. Отладка программы. Отладка (пункт меню), начать отладку, выполнить построение, ДА. Повторное открытие проекта осуществить с помощью функции Открыть проект.

5. Ввод программного кода:

```
#include "stdafx.h"
#include <iostream>
using namespace std;

int main()
{
    double x, y, k;
    do
    {
        cout << "Vvedite chislo: ";
        cin >> x;
        if (x < -3)
            y = x + 1;
        else if (x >= 2)
            y = 3 * x;
        else
            y = 2 * x + 1;
        cout << "y = " << y << endl << "go again? (1 -da/2 - net)";
        cin >> k;
    } while (k == 1);
    system("pause");
    return 0;
}
```

6. Построение проекта: после ввода программного кода нужно скомпилировать и отладить программу. Для этого: Построение – Компилировать (Ctrl + F7) – Программа скомпилирована. Если в программном коде имеются ошибки, они отобразятся в нижней части окна. Чтобы увидеть, в каком месте ошибка, нужно дважды кликнуть на сообщении об ошибке. Синий маркер укажет на строку в программном коде, где имеется ошибка. Далее создаем исполняемый

exe-файл: Построение – Построить решение (F7). Чтобы просмотреть результат выполнения программы, нужно выполнить: Отладка – Запуск без отладки (Ctrl+F5).

Результат выполнения программы:

```
Vvedite chislo: -3
y = -5
go again? (1 -da/2 - net)1
Vvedite chislo: -4
y = -3
go again? (1 -da/2 - net)1
Vvedite chislo: 4
y = 12
go again? (1 -da/2 - net)2
Для продолжения нажмите любую клавишу . . .
```

Задания для самостоятельного выполнения:

Задание 2.1. Простые конструкции языка. Программы с ветвлением, циклами

1. Составить блок-схему и программу увеличения большего из двух чисел на 1.
2. Составить блок-схему и программу определения значений переменной Y по одной из формул в зависимости от величины X:

$$y = \begin{cases} x+1, & \text{если } x < -3; \\ 2x+1, & \text{если } -3 \leq x < 2; \\ 3x, & \text{если } x \geq 2. \end{cases}$$

3. Составить блок-схему и программу, которая большее из двух произвольных чисел возводит в квадрат, а меньшее увеличивает на единицу.
4. Составить блок-схему и программу, по которой меньшее из двух чисел будет умножено на большее, а большее разделено на меньшее.

5. Составить блок-схему и программу, по которой меньшее из двух чисел будет умножено на пять, а большее разделено на пять.

6. С клавиатуры вводится 10 произвольных чисел. Составить блок-схему и программу определения суммы всех вводимых чисел.

7. Составить блок-схему и программу нахождения суммы удвоенных чётных чисел в интервале от 15 до 90.

8. Составить блок-схему и программу нахождения произведения ряда натуральных чисел от 15 до 41.

9. Составить блок-схему и программу определения наименьшего из десяти произвольных чисел, введённых с клавиатуры.

10. Составить блок-схему и программу определения наибольшего из десяти произвольных чисел, введённых с клавиатуры.

11. С клавиатуры вводится 10 произвольных чисел. Составить блок-схему и программу, определяющие количество введённых отрицательных чисел.

12. С клавиатуры вводится 10 произвольных чисел. Составить блок-схему и программу, определяющие количество введённых положительных чётных чисел.

13. С клавиатуры вводится 10 произвольных чисел. Составить блок-схему и программу, определяющие сумму нечётных чисел и их среднее арифметическое значение.

14. Составить блок-схему и программу, выводящие на экран дисплея таблицу квадратов чисел, меняющихся в интервале от 1 до 20 с шагом 0,5.

15. Составить блок-схему и программу определения площади квадратов со стороной, меняющейся от 2 до 4 см с шагом 3 мм.

16. Составить блок-схему и программу определения площади прямоугольников высотой 4 см с основанием, меняющимся от 3 до 6 см с шагом 2 мм.

17. Составить блок-схему и программу определения объёма кубов со стороной, меняющейся от 5 до 8 см с шагом 2 мм.

18. Составить блок-схему и программу определения значения переменной Y по одной из формул:

$$y = \begin{cases} x+1, & \text{если } x < -3; \\ 2x+1, & \text{если } -3 \leq x < 2; \\ 3x, & \text{если } x \geq 2. \end{cases}$$

Аргумент X меняется в интервале от -5 до 5 с шагом 0,2.

19. Составить блок-схему и программу определения периметров квадратов со стороной, меняющейся от 3 до 9 см с шагом 3 мм.

20. С клавиатуры вводится 10 произвольных чисел. Составить блок-схему и программу, определяющие количество введенных четных чисел и произведение положительных.

К лабораторной работе № 3. Вычисление суммы и произведения по заданной формуле

Пример 3.1. Вычислить значение S:

$$S = \sum_{i=0}^{10} \frac{(-1)^i (i+1)}{i!}$$

Порядок выполнения:

1. Пуск, программы, Visual Studio 2015, Создать проект, Visual C++, Консольное приложение Win32, имя (задать имя проекта), расположение (диск, каталог), имя решения (появится автоматически), ОК, консольное приложение, пустой проект, готово.

2. Проект (пункт меню), добавить новый элемент, файл C++ (.cpp), имя (ввести имя программы), добавить.

3. В папке Файлы исходного кода появился файл с введенным именем (имя.cpp). Набрать текст в редакторе. Выполнить: Проект, Построить решение.

4. Отладка программы. Отладка (пункт меню), начать отладку, выполнить построение, ДА. Повторное открытие проекта осуществить с помощью функции Открыть проект.

5. Ввод программного кода:

```
#include "stdafx.h"
#include <iostream>
#include <math.h>

using namespace std;

int main()
{
    setlocale(0, "Russian");
    double i, s = 0, f=1, x;

    for (i = 0; i <= 10; i++)
    {
        if (i == 0)
            f = 1;
        else
            f *= i;

        x = pow(-1,i)*(i+1) / f;
        cout << "Значение " << i + 1 << " члена выражения = " << x << endl;
        s+=x;
    }
    cout << endl << "Ответ: " << s << endl;
    system("pause");
    return 0;
}
```

6. Построение проекта: после ввода программного кода нужно скомпилировать и отладить программу. Для этого: Построение – Компилировать (Ctrl + F7) – Программа скомпилирована. Если в программном коде имеются ошибки, они отобразятся в нижней части окна. Чтобы увидеть, в каком месте ошибка, нужно дважды кликнуть на сообщении об ошибке. Синий маркер укажет на строку в программном коде, где имеется ошибка. Далее создаем исполняемый exe-файл: Построение – Построить решение (F7). Чтобы просмотреть результат

выполнения программы, нужно выполнить: Отладка – Запуск без отладки (Ctrl+F5).

Результат выполнения программы

```
Значение 1 члена выражения = 1
Значение 2 члена выражения = -2
Значение 3 члена выражения = 1.5
Значение 4 члена выражения = -0.666667
Значение 5 члена выражения = 0.208333
Значение 6 члена выражения = -0.05
Значение 7 члена выражения = 0.00972222
Значение 8 члена выражения = -0.0015873
Значение 9 члена выражения = 0.000223214
Значение 10 члена выражения = -2.75573e-05
Значение 11 члена выражения = 3.03131e-06

Ответ: 2.75573e-07
Для продолжения нажмите любую клавишу . . .
```

Задания для самостоятельного выполнения:

Задание 3.1. Нахождение суммы, произведения по заданной формуле:

1. Вычислить значение S.

$$S = \sum_{i=1}^{10} \frac{i}{i+1}$$

2. Вычислить значение S.

$$S = \sum_{i=0}^{10} \frac{(-1)^i (i+1)}{i!}$$

3. Вычислить значение S.

$$S = \sum_{i=0}^{10} \frac{(-1)^{i+1}}{i!}$$

4. Вычислить значение S.

$$S = \sum_{i=1}^{10} \frac{(-1)^{i+1}}{2i}$$

5. Вычислить значение S.

$$S = \sum_{i=0}^{10} \frac{1}{3^i + 2^i}$$

7. Вычислить значение S.

$$S = \sum_{i=1}^{10} \frac{i^2}{2i-1}$$

9. Вычислить значение S.

$$S = \sum_{i=1}^{10} \frac{i}{2i+3}$$

11. Вычислить значение P.

$$P = \prod_{i=1}^5 \frac{i}{i+3i}$$

13. Вычислить произведение P.

$$P = \prod_{i=2}^5 \left(1 - \frac{1}{i!}\right)^2$$

6. Вычислить значение S.

$$S = \sum_{i=1}^{10} \frac{(-1)^{2i-1}}{(i-1)!}$$

8. Вычислить значение S.

$$S = \sum_{i=1}^{10} \frac{(-1)^{i+1}}{(i^2)!}$$

10. Вычислить значение S.

$$S = \sum_{i=1}^{10} \frac{1}{i+i!}$$

12. Вычислить произведение P.

$$P = \prod_{i=1}^5 \frac{i+1}{i+2}$$

14. Вычислить произведение P.

$$P = \prod_{i=1}^5 \frac{i^2}{(i+1)}$$

15. Вычислить произведение P.

$$P = \prod_{i=1}^5 \frac{i+3}{i-2}$$

16. Вычислить произведение P.

$$P = \prod_{i=1}^5 \frac{i}{i+2i+3}$$

17. Вычислить произведение P.

$$P = \prod_{i=1}^5 \frac{i}{i+1}$$

18. Вычислить произведение P.

$$P = \prod_{i=1}^5 \frac{\sqrt{i}}{1+i}$$

19. Вычислить произведение P.

$$P = \prod_{i=1}^5 \frac{\sqrt{i}}{\sqrt{i+1}}$$

20. Вычислить произведение P.

$$P = \prod_{i=1}^5 \frac{i^3}{i^2+1}$$

К лабораторной работе № 4. Линейные статические массивы

Пример 4.1. Вводится массив из 10-ти элементов. Вывести на экран эти элементы. Найти количество элементов, находящихся в диапазоне между двумя заданными введенными числами. Вывести результат.

Порядок выполнения:

1. Пуск, программы, Visual Studio 2015, Создать проект, Visual C++, Консольное приложение Win32, имя (задать имя проекта), расположение (диск, ка-

талог), имя решения (появится автоматически), ОК, консольное приложение, пустой проект, готово.

2. Проект (пункт меню), добавить новый элемент, файл C++ (.cpp), имя (ввести имя программы), добавить.

3. В папке Файлы исходного кода появился файл с введенным именем (имя.cpp). Набрать текст в редакторе. Выполнить: Проект, Построить решение.

4. Отладка программы. Отладка (пункт меню), начать отладку, выполнить построение, ДА. Повторное открытие проекта осуществить с помощью функции Открыть проект.

5. Ввод программного кода:

```
#include "stdafx.h"
#include <iostream>
#include <math.h>
#include <Windows.h>
#include <ctime>

using namespace std;

int main(void)
{
    setlocale(0, "Russian");
    srand(time(0));
    int i, k, z, x = 0, A[10];
    char ss;
    cout << "Ввести массив самостоятельно? (y/n): ";
    cin >> ss;
    if (ss == 'y')
    {
        for (i = 0; i < 10; i++)
        {
            cout << "Введите " << i + 1 << " число ";
            cin >> A[i];
        }
    }
    else
    {
        for (i = 0; i < 10; i++)
        {
            A[i] = -100 + rand() % 201;
        }
        cout << "ПК формирует для вас массив, пожалуйста, подожди-
        те...\n";
    }
}
```

```

        Sleep(2000);
    }
    cout << "Получился следующий массив: ";
    for (i = 0; i < 10; i++)
    {
        cout << A[i] << " ";
    }
    do {
        cout << "\n\nВведите значения интервала [a,b]\nВведите начальное
значение: ";
        cin >> k;
        cout << "Введите конечное значение: ";
        cin >> z;

        if (k >= z) {
            cout << "\nНекорректный ввод интервала!\n";
            if (k == z)
                cout << "Начальное значение равно конечному!";
            else
                cout << "Начальное значение больше конечного";
        }
    } while (k >= z);

    for (i = 0; i < 10; i++)
    {
        if (A[i] >= k & A[i] <= z)
            x++;
        else
            continue;
    }
    if (x == 0)
        cout << "\nНет элементов массива в интервале от " << k << " до "
<< z << endl;
    else
        cout << "\nКол-во элементов массива в интервале от " << k << " до "
<< z << " равно " << x << endl;
    system("pause");
    return 0;
}

```

6. Построение проекта: после ввода программного кода нужно скомпилировать и отладить программу. Для этого: Построение – Компилировать (Ctrl + F7) – Программа скомпилирована. Если в программном коде имеются ошибки, они отобразятся в нижней части окна. Чтобы увидеть, в каком месте ошибка, нужно дважды кликнуть на сообщении об ошибке. Синий маркер укажет на

строку в программном коде, где имеется ошибка. Далее создаем исполняемый exe-файл: Построение – Построить решение (F7). Чтобы просмотреть результат выполнения программы, нужно выполнить: Отладка – Запуск без отладки (Ctrl+F5).

Результат выполнения программы:

```
Ввести массив самостоятельно? (y/n): n
ПК формирует для вас массив, пожалуйста, подождите...
Получился следующий массив: -44 -37 -80 -16 72 95 -91 35 -32 -66

Введите значения интервала [a,b]
Введите начальное значение: -50
Введите конечное значение: 67

Кол-во элементов массива в интервале от -50 до 67 равно 5
Для продолжения нажмите любую клавишу . . . ■
```

Задания для самостоятельного выполнения:

Задание 4.1. Линейные массивы

1. Задан массив из 10 элементов. Найти сумму элементов массива. Разделить каждый элемент исходного массива на полученное значение. Результат получить в том же массиве. Вывести на печать исходный массив, сумму, преобразованный массив.
2. Задан массив из 10 элементов. Найти произведение нечетных элементов массива. Разделить каждый элемент исходного массива на полученное значение. Результат получить в том же
3. Задан массив из 10 элементов. Найти количество элементов, равных

4. Умножить каждый элемент исходного массива на полученное значение. Результат получить в том же массиве. Вывести на печать исходный массив, количество, преобразованный массив.

5. Задан массив из 10 элементов. Вычислить среднее арифметическое значение четных элементов массива. Умножить каждый элемент исходного массива на полученное значение. Результат получить в том же массиве. Вывести на печать исходный массив, среднее арифметическое, преобразованный массив.

6. Задан массив из 10 элементов. Найти количество элементов, равных 5, их сумму и произведение. Среди полученных данных найти минимальное значение. Умножить каждый элемент исходного массива на полученное минимальное значение. Результат получить в том же массиве. Вывести на печать исходный массив, количество, сумму, произведение, минимальное значение, преобразованный массив.

7. Вычислить сумму и разность элементов двух заданных линейных массивов размерностью 10. Вывести на печать исходные массивы и результат.

8. Задан массив X размером 10. Вычислить значение функции $y = 0,3 * \ln x$ при значениях аргумента, заданных в массиве X, и поместить их в массив Y. Вывести на печать исходный массив X и полученный массив Y.

9. Задан массив из 10 элементов. Вычислить среднее арифметическое значение положительных элементов массива и максимальное среди отрицательных элементов массива. Вывести на печать исходный массив, среднее арифметическое, максимальное значение.

10. Заданы массивы A и B из 4 и 8 элементов соответственно. Объединить положительные элементы массива в массив C. Вывести на печать исходные массивы и полученный массив.

11. Задан массив из 10 элементов. Вычислить среднее арифметическое значение четных, положительных элементов массива. Умножить каждый эле-

мент исходного массива на полученное значение. Результат получить в том же массиве. Вывести на печать исходный массив, среднее арифметическое, преобразованный массив.

12. Задана информация о среднесуточной температуре воздуха по дням месяца января. Определить, температура скольких дней была ниже среднемесячной. Вывести на печать исходный массив, среднемесячное значение, количество дней, удовлетворявших заданному критерию.

13. Задана информация об урожайности сбора зерна в 10 районах области (районы имеют порядковую нумерацию). Определить номер района, собравшего наибольшее количества зерна. Вывести на печать исходный массив, результат максимального сбора урожая и номер района.

14. Задана информация об успеваемости в 12 учебных группах в баллах. Определить на сколько нужно повысить успеваемость в самой отстающей группе, чтобы достичь среднего уровня успеваемости. Вывести на печать исходный массив, результат решения задачи.

15. Известны данные о среднемесячной температуре за год. Определить, какая была самая высокая температура летом и самая низкая зимой. Вывести на печать исходный массив, результат решения задачи.

16. Задан массив размером 14 элементов. Сформировать из исходного два массива, поместив в первый – четные элементы массива, а во второй отрицательные. Вывести на печать исходные массивы, результат решения задачи.

17. Задан массив размерностью 15. Найти сумму двух минимальных элементов массива. Вывести на печать исходный массив, результат решения задачи.

18. В каком из заданных 10 банков выгоднее всего купить валюту? Вывести на печать исходный массив, результат решения задачи.

19. В каком из заданных 10 банков выгоднее всего продать валюту? Вывести на печать исходный массив, результат решения задачи.

20. Составить программу, осуществляющую в одномерном массиве из 12 элементов перестановку двух его частей: содержащих элементы от 1 до m –го элемента и от $(m+1)$ – го до n –го. Например, $m=4$ массив 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 должен преобразоваться в 5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3, 4. Вывести на печать исходный массив, количество, преобразованный массив.

21. Составить программу, осуществляющую в одномерном массиве из 12 элементов сдвиг на k ($k < n$) элементов вправо, при этом первые k элементов заменяются нулями, а последние k утрачиваются. Вывести на печать исходный массив, количество, преобразованный массив.

К лабораторной работе № 5. Двумерные статические массивы

Пример 5.1. Дана матрица $A(n, n)$. Дана матрица $A(n, n)$. Найти сумму элементов по каждому столбцу.

Порядок выполнения:

1. Пуск, программы, Visual Studio 2015, Создать проект, Visual C++, Консольное приложение Win32, имя (задать имя проекта), расположение (диск, каталог), имя решения (появится автоматически), ОК, консольное приложение, пустой проект, готово.

2. Проект (пункт меню), добавить новый элемент, файл C++ (.cpp), имя (ввести имя программы), добавить.

3. В папке Файлы исходного кода появился файл с введенным именем (имя.cpp). Набрать текст в редакторе. Выполнить: Проект, Построить решение.

4. Отладка программы. Отладка (пункт меню), начать отладку, выполнить построение, ДА. Повторное открытие проекта осуществить с помощью функции Открыть проект.

5. Ввод программного кода:

```
#include "stdafx.h"
#include <iostream>
#include <math.h>
#include <Windows.h>
#include <ctime>
using namespace std;
```

```
int main(void)
{
    setlocale(0, "Russian");
```

```

srand(time(0));
int A[50][50], i, j, n, s;
char ss;
cout << "Введите размерность массива: ";
cin >> n;
cout << "Ввести массив самостоятельно? (y/n): ";
cin >> ss;
if (ss == 'y')
{
    for (i = 0; i < n; i++)
    {
        cout << "Введите " << i + 1 << " строку!\n";
        for (j = 0; j < n; j++)
        {
            cout << "Введите " << j + 1 << " число: ";
            cin >> A[i][j];
        }
    }
}
else
{
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
        {
            A[i][j] = -30 + rand() % 61;
        }
    }
    cout << "ПК формирует для вас массив, пожалуйста, подожди-
me...\n";
    Sleep(2000);
}
if (ss == 'y')
    cout << "\nВы ввели: \n\n";
else
    cout << "\nПолучился следующий массив\n\n";
for (i = 0; i < n; i++)
{
    cout << "\t";
    for (j = 0; j < n; j++)
    {
        if (A[i][j] < 0)
            cout << A[i][j] << "\t";
    }
}

```

```

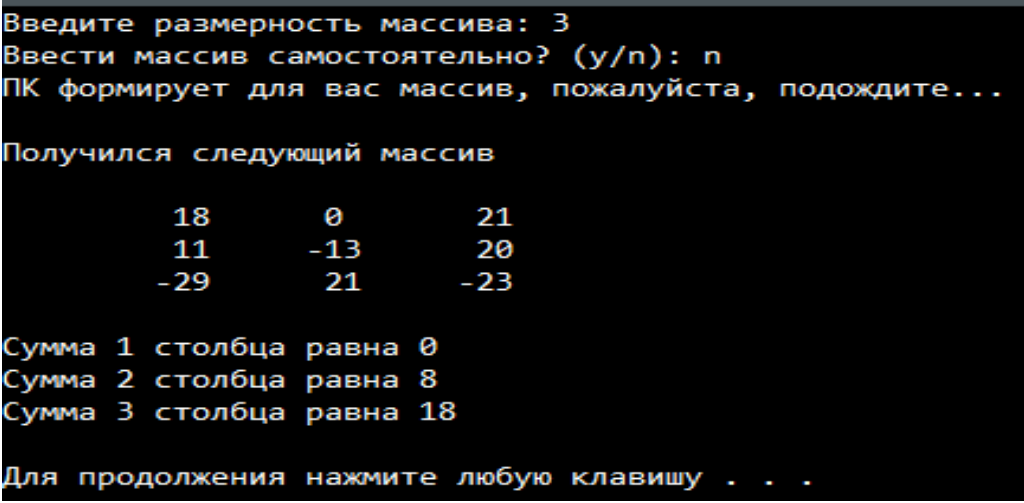
        else
            cout<< " " << A[i][j] << "\t";
    }
    cout << endl;
}
cout << endl;
for (j = 0; j < n; j++)
{
    for (i = 0, s = 0; i < n; i++)
    {
        s += A[i][j];
    }
    cout << "Сумма " << j + 1 << " столбца равна " << s << endl;
}

cout << endl;
system("pause");
return 0;
}

```

6. Построение проекта: после ввода программного кода нужно скомпилировать и отладить программу. Для этого: Построение – Компилировать (Ctrl + F7) – Программа скомпилирована. Если в программном коде имеются ошибки, они отобразятся в нижней части окна. Чтобы увидеть, в каком месте ошибка, нужно дважды кликнуть на сообщении об ошибке. Синий маркер укажет на строку в программном коде, где имеется ошибка. Далее создаем исполняемый exe-файл: Построение – Построить решение (F7). Чтобы просмотреть результат выполнения программы, нужно выполнить: Отладка – Запуск без отладки (Ctrl+F5).

Результат выполнения программы:



```

Введите размерность массива: 3
Ввести массив самостоятельно? (y/n): n
ПК формирует для вас массив, пожалуйста, подождите...

Получился следующий массив

      18      0      21
      11     -13     20
     -29      21     -23

Сумма 1 столбца равна 0
Сумма 2 столбца равна 8
Сумма 3 столбца равна 18

Для продолжения нажмите любую клавишу . . .

```


Задания для самостоятельного выполнения:

Задание 5.1. Двумерные массивы (1)

1. Дана матрица $A(n,n)$. Найти сумму элементов по каждому столбцу.
2. Дана матрица $A(n,n)$. Найти сумму элементов по каждой строке.
3. Дана матрица $A(n,m)$. Получить векторный столбец, элементы которого являются наибольшими по модулю элементов строк исходной матрицы.
4. Дана матрица $A(n,m)$. Получить векторный столбец, элементы которого являются наименьшим по модулю столбцов исходной матрицы.
5. Дана матрица $A(n,n)$. Найти сумму элементов, принадлежащих главной диагонали матрицы, разделив её на максимальный диагональный элемент.
6. Дана матрица $A(n,n)$. Найти произведение элементов, принадлежащих вспомогательной диагонали матрицы, разделив его на максимальный элемент вспомогательной диагонали.
7. Дана матрица $A(n,n)$. Найти максимальный элемент матрицы среди элементов, находящихся выше главной диагонали.
8. Дана матрица $A(n,n)$. Найти максимальный элемент матрицы среди элементов, находящихся ниже главной диагонали.
9. Дана матрица $A(n,n)$. Найти разность между максимальным элементом, принадлежащим главной диагонали, и минимальным элементом, принадлежащим вспомогательной диагонали.
10. Дана матрица $A(n,n)$. Найти сумму нечётных элементов матрицы и произведение элементов, находящихся на главной диагонали.
11. Дана матрица $A(n,m)$. Получить новую матрицу путём деления всех элементов исходной матрицы на элемент, наименьший по модулю.

12. Дана матрица $A(n,n)$. Получить новую матрицу путём деления всех элементов исходной матрицы на элемент, наименьший по модулю на главной диагонали исходной матрицы.

13. Дана матрица $A(n,n)$. Найти сумму элементов по каждому столбцу, разделив её на минимальный диагональный элемент.

14. Дана матрица $A(n,n)$. Каждый минимальный элемент строки поменять на максимальное значение диагонального элемента.

15. Дана матрица $A(n,n)$. Каждый максимальный элемент столбца поменять на минимальное значение диагонального элемента.

16. Дана матрица $A(n,n)$. Получить новую матрицу, каждый элемент столбца которой получается путём деления на максимальный элемент этого же столбца.

17. Дана матрица $A(n,n)$. Получить новую матрицу путём деления всех элементов исходной матрицы на сумму элементов, находящихся на главной диагонали.

18. Дана матрица $A(n,n)$. Получить новую матрицу путём деления всех элементов исходной матрицы на среднее арифметическое значений элементов, находящихся на главной диагонали.

19. Дана матрица $A(n,n)$. Получить новую матрицу путём деления всех элементов исходной матрицы на среднее арифметическое значений элементов, находящихся на вспомогательной диагонали.

20. Дана матрица $A(n,n)$. Получить новую матрицу путём деления всех элементов матрицы на элемент, значение которого является \min в L – строке каждого столбца.

Задание 5.2. Двумерные массивы (2.) Построить матрица по указанным ниже правилам.

$$1. \begin{pmatrix} 1 & 2 & 3 & \dots & n \\ n & n-1 & n-2 & \dots & 1 \\ 1 & 2 & 3 & \dots & n \\ n & n-1 & n-2 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ n & n-1 & n-2 & \dots & 1 \end{pmatrix}$$

(n — четное).

$$2. \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 & 1 \\ 0 & 0 & 0 & \dots & 0 & 2 & 0 \\ 0 & 0 & 0 & \dots & 3 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & n-1 & 0 & \dots & 0 & 0 & 0 \\ n & 0 & 0 & \dots & 0 & 0 & 0 \end{pmatrix}$$

$$3. \begin{pmatrix} n & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & n-1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & n-2 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 2 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{pmatrix}$$

$$4. \begin{pmatrix} 1-2 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 2-3 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 3-4 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & (n-1)n & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & n(n+1) \end{pmatrix}$$

$$5. \begin{pmatrix} 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ 1 & 0 & 0 & \dots & 0 & 0 & 1 \\ 1 & 0 & 0 & \dots & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & \dots & 0 & 0 & 1 \\ 1 & 1 & 1 & \dots & 1 & 1 & 1 \end{pmatrix}$$

$$6. \begin{pmatrix} 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ 2 & 2 & 2 & \dots & 2 & 2 & 0 \\ 3 & 3 & 3 & \dots & 3 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ n-1 & n-1 & 0 & \dots & 0 & 0 & 0 \\ n & 0 & 0 & \dots & 0 & 0 & 0 \end{pmatrix}$$

$$7. \begin{pmatrix} 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ 0 & 1 & 1 & \dots & 1 & 1 & 0 \\ 0 & 0 & 1 & \dots & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 1 & 1 & \dots & 1 & 1 & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 & 1 \end{pmatrix}$$

$$8. \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 & 1 \\ 1 & 1 & 0 & \dots & 0 & 1 & 1 \\ 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 1 & 1 & 0 & \dots & 0 & 1 & 1 \\ 1 & 0 & 0 & \dots & 0 & 0 & 1 \end{pmatrix}$$

$$9. \begin{pmatrix} n & 0 & 0 & \dots & 0 & 0 & 0 \\ n-1 & n & 0 & \dots & 0 & 0 & 0 \\ n-2 & n-1 & n & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 2 & 3 & 4 & \dots & n-1 & n & 0 \\ 1 & 2 & 3 & \dots & n-2 & n-1 & n \end{pmatrix}$$

$$10. \begin{pmatrix} 1 & 2 & 3 & \dots & n-2 & n-1 & n \\ 2 & 3 & 4 & \dots & n-1 & n & 0 \\ 3 & 4 & 5 & \dots & n & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ n-1 & n & 0 & \dots & 0 & 0 & 0 \\ n & 0 & 0 & \dots & 0 & 0 & 0 \end{pmatrix}$$

$$11. \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 & n \\ 0 & 2 & 0 & \dots & 0 & n-1 & 0 \\ 0 & 0 & 3 & \dots & n-2 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 2 & 0 & \dots & 0 & n-1 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 & n \end{pmatrix}$$

$$12. \begin{pmatrix} 1 & 2 & 3 & \dots & n-2 & n-1 & n \\ 2 & 1 & 2 & \dots & n-2 & n-2 & n-1 \\ 3 & 2 & 1 & \dots & n-4 & n-3 & n-2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ n-1 & n-2 & n-3 & \dots & 2 & 1 & 2 \\ n & n-1 & n-2 & \dots & 3 & 2 & 1 \end{pmatrix}$$

13. Дана матрица $A(n,n)$. Получить новую матрицу путём деления всех элементов матрицы на элемент, значение которого равно разности между максимальным и минимальным элементами.

14. Заданы два одномерных массива с различным количеством элементов и натуральное число k . объединить их в один массив, включив второй массив между k -м и $(k+1)$ -м элементами первого, при этом, не используя дополнительный массив. Отсортировать полученный массив в порядке убывания элементов.

15. Даны две последовательности a_1, a_2, \dots, a_n и b_1, b_2, \dots, b_m . Образовать из них новую последовательность чисел так, чтобы она также была неубывающей. Примечание: дополнительный массив не использовать.

16. Сортировка выбором. Дана последовательность чисел a_1, a_2, \dots, a_n . требуется переставить элементы так, чтобы они были расположены по убыванию. для этого в массиве, начиная с первого, выбирается наибольший элемент и ставится на первое место, а первый — на место наибольшего. Затем, начиная со второго, эта процедура повторяется. написать алгоритм сортировки выбором.

17. Пусть даны две неубывающие последовательности действительных чисел a_1, a_2, \dots, a_n и b_1, b_2, \dots, b_m . требуется указать те места, на которые нужно вставлять элементы последовательности b_1, b_2, \dots, b_m в первую последовательность так, чтобы новая последовательность оставалась возрастающей.

18. В таблицу занесены данные о количестве игр, сыгранных каждой из 10 футбольных команд с другой командой. Подсчитать и напечатать общее количество игр, сыгранных каждой командой, а также общее число игр в чемпионате.

19. Известно расстояния между каждыми двумя из 12 городов. Определить, между какими городами расстояние наименьшее.

20. Известно расстояния между каждыми двумя из 12 городов. Определить, между какими городами расстояние наибольшее.

К лабораторной работе № 6. Указатели

Пример 6.1. Использование указателей. Вывод значений переменных и адресов.

Порядок выполнения:

1. Пуск, программы, Visual Studio 2015, Создать проект, Visual C++, Консольное приложение Win32, имя (задать имя проекта), расположение (диск, каталог), имя решения (появится автоматически), ОК, консольное приложение, пустой проект, готово.

2. Проект (пункт меню), добавить новый элемент, файл C++ (.cpp), имя (ввести имя программы), добавить.

3. В папке Файлы исходного кода появился файл с введенным именем (имя.cpp). Набрать текст в редакторе. Выполнить: Проект, Построить решение.

4. Отладка программы. Отладка (пункт меню), начать отладку, выполнить построение, ДА. Повторное открытие проекта осуществить с помощью функции Открыть проект.

5. Ввод программного кода:

```
#include <iostream>
using std::cout;
using std::cin;
using std::endl;
using std::hex;
using std::dec;
//using namespace std;

int main()
{
    long* pnumber(nullptr); // объявление и инициализация указателя
    long number1(55), number2(99); // объявление и инициализация переменных
    pnumber = &number1; // сохранить адрес в указателе
    *pnumber += 11; // увеличить number1 на 11
    cout << endl; // перевод строки
    cout << "number1 = " << number1; // вывод значения переменной
    cout << " &number1 = " << hex << pnumber; // вывод значения адреса
```

```

    cout << " &number1= "<<hex<<pnumber+1;// вывод адреса для следующей перемен-
ной long
    cout << " &number1= "<<hex<<pnumber+2;
    pnumber=&number2;//изменить указатель на адрес number2
    number1= *pnumber*10;// умножить number2 на 10
    cout << endl;//
    cout << " number1 = "<<dec<<number1;// вывод измененного number2 через указа-
тель в десятичном виде
    cout << " pnumber = "<<hex<<pnumber;// вывод адреса number2
    cout << " *pnumber= "<<dec<<*pnumber;// вывод значения по заданному адресу
    cout << endl;
    system ("pause");
    return 0;
}

```

6. Построение проекта: после ввода программного кода нужно скомпилировать и отладить программу. Для этого: Построение – Компилировать (Ctrl + F7) – Программа скомпилирована. Если в программном коде имеются ошибки, они отобразятся в нижней части окна. Чтобы увидеть, в каком месте ошибка, нужно дважды кликнуть на сообщении об ошибке. Синий маркер укажет на строку в программном коде, где имеется ошибка. Далее создаем исполняемый exe-файл: Построение – Построить решение (F7). Чтобы просмотреть результат выполнения программы, нужно выполнить: Отладка – Запуск без отладки (Ctrl+F5).

Результат выполнения программы:

```

number1 = 66 &number1= 0012FF54 &number1= 0012FF58 &number1= 0012FF5C
number1 = 990 pnumber = 0012FF48 *pnumber= 99
Для продолжения нажмите любую клавишу . . .

```

Задания для самостоятельного выполнения:

Задание 6.1. Указатели. Линейные массивы

1. Ввести два числа. Вычислить их сумму и разность с использованием указателя. Вывести адреса и значения исходных данных и результата.

2. Ввести два числа. Вычислить их произведение и разность с использованием указателя. Вывести адреса и значения исходных данных и результата.

3. Ввести два числа. Вычислить разность между произведением и суммой чисел с использованием указателя. Вывести адреса и значения исходных данных и результата.

4. Ввести два числа. Вычислить частное от деления двух чисел, при условии неравного нулю делителя. Вывести адреса и значения исходных данных и результата.

5. Ввести два числа. Вычислить сумму квадратов этих чисел с использованием указателя. Вывести адреса и значения исходных данных и результата.

6. Ввести два числа. Вычислить разность квадратов этих чисел с использованием указателя. Вывести адреса и значения исходных данных и результата.

7. Ввести два числа. Вычислить сумму кубов этих чисел с использованием указателя. Вывести адреса и значения исходных данных и результата.

8. Ввести два числа. Вычислить разность кубов этих чисел с использованием указателя. Вывести адреса и значения исходных данных и результата.

9. Ввести два числа. Вычислить сумму удвоенного первого числа утроенного второго числа с использованием указателя. Вывести адреса и значения исходных данных и результата.

10. Ввести два числа. Вычислить произведение увеличенного вдвое первого числа и уменьшенного вдвое второго числа с использованием указателя. Вывести адреса и значения исходных данных и результата.

11. Вводится массив из 10-ти элементов. Вывести на экран эти элементы. Найдите количество элементов, находящихся в диапазоне между двумя заданными введенными числами. Выведите адреса этих чисел, адрес начала массива, адреса и значения чисел в диапазоне, адрес и значение результата или сообщение об отсутствии таких элементов.

12. Вводится массив из 10-ти элементов. Вывести на экран эти элементы. Вычислите количество элементов массива, равных нулю. Выведите адрес начала массива, адрес и индекс первого нулевого элемента или выдайте сообщение об отсутствии этих элементов.

13. Вводится массив из 10-ти элементов. Вывести на экран эти элементы. Найдите количество элементов, больших заданного числа. Выведите адрес начала массива, адрес введенного числа адрес и индекс результата или выдайте сообщение об отсутствии этих элементов.

14. Вводится массив из 10-ти элементов. Вывести на экран эти элементы. Найдите минимальный и максимальный элементы массива и поменяйте их местами. Выведите адрес начала массива, адреса и значения всех минимальных и максимальных элементов, результирующий массив.

15. Вводится массив из 10-ти элементов. Вывести на экран эти элементы. Найдите номера минимальных по модулю элементов массива. Выведите адрес начала массива, адреса, номера и значения всех минимальных по модулю элементов.

16. Вводится массив из 10-ти элементов. Вывести на экран эти элементы. Вычислите произведение положительных элементов массива. Выведите адрес начала массива, адрес и значение результата или сообщение об отсутствии этих элементов.

17. Вводится массив из 10-ти элементов. Вывести на экран эти элементы. Вычислите произведение отрицательных элементов массива. Выведите адрес начала массива, адрес и значение результата или сообщение об отсутствии этих элементов.

18. Вводится массив из 10-ти элементов. Вывести на экран эти элементы. Вычислите количество отрицательных элементов массива. Выведите адрес начала массива, адрес и значение результата или сообщение об отсутствии этих элементов.

19. Вводится массив из 10-ти элементов. Вывести на экран эти элементы. Вычислите произведение элементов с четными номерами. Выведите адрес начала массива, адрес и значение результата или сообщение об отсутствии этих элементов.

20. Вводится массив из 10-ти элементов. Вывести на экран эти элементы. Вычислите сумму элементов с нечетными номерами. Выведите адрес начала массива, адрес и значение результата или сообщение об отсутствии этих элементов.

К лабораторной работе № 7. Строки

Пример 7.1. Задан текст. Слова в тексте разделены пробелами. В конце текста – точка. Определить, сколько раз в тексте встречается буква «о».

Порядок выполнения:

1. Пуск, программы, Visual Studio 2015, Создать проект, Visual C++, Консольное приложение Win32, имя (задать имя проекта), расположение (диск, каталог), имя решения (появится автоматически), ОК, консольное приложение, пустой проект, готово.

2. Проект (пункт меню), добавить новый элемент, файл C++ (.cpp), имя (ввести имя программы), добавить.

3. В папке Файлы исходного кода появился файл с введенным именем (имя.cpp). Набрать текст в редакторе. Выполнить: Проект, Построить решение.

4. Отладка программы. Отладка (пункт меню), начать отладку, выполнить построение, ДА. Повторное открытие проекта осуществить с помощью функции Открыть проект.

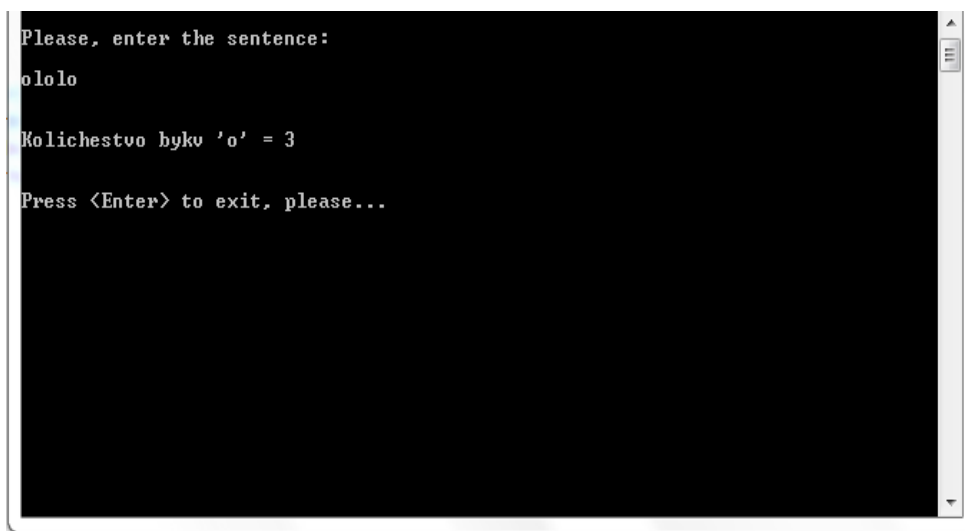
5. Ввод программного кода:

```
// #include "stdafx.h"
#include <string.h>
#include <string>
#include <conio.h>
#include <iostream>
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
    char str [50];
    int i, kol;
    string s;
    cout << "\nPlease, enter the sentence: \n\n";
    kol = 0;
    getline(cin, s);
    strcpy(str,s.c_str());
    for (i = 0; i <= strlen (str) - 1; i++)
        if (str [i] == 'o' || str [i] == 'O') kol++;
    cout << "\n\nKolichestvo bykv 'o' = " << kol << '\n';
    cout << "\n\nPress <Enter> to exit, please...";
    getch();
    return 0;
}
```

6. Построение проекта: после ввода программного кода нужно скомпилировать и отладить программу. Для этого: Построение – Компилировать (Ctrl +

F7) – Программа скомпилирована. Если в программном коде имеются ошибки, они отобразятся в нижней части окна. Чтобы увидеть, в каком месте ошибка, нужно дважды кликнуть на сообщении об ошибке. Синий маркер укажет на строку в программном коде, где имеется ошибка. Далее создаем исполняемый exe-файл: Построение – Построить решение (F7). Чтобы просмотреть результат выполнения программы, нужно выполнить: Отладка – Запуск без отладки (Ctrl+F5).

Результат выполнения программы:

A screenshot of a console window with a black background and white text. The text displayed is: "Please, enter the sentence:", followed by the user input "ololo" on the next line. Then, the program output "Kolichestvo bykv 'o' = 3" is shown. The final line is "Press <Enter> to exit, please...". The console window has a standard Windows-style title bar and a scrollbar on the right side.

Пример 7.2. Задана строка текста. Слова в текст отделены пробелами, в конце текста – точка. Найти самые короткие слова в тексте.

```
#include "stdafx.h"
#include <iostream>
#include<string>
using namespace std;
int main() {
    setlocale(LC_ALL, "Russian");
    int min1, i, j = 0;
    char s1[] = "найти шифр было невозможно.";
    int r[100];
    char *s2;
    char *next = 0;
    s2 = strtok_s(s1, " ;", &next);
    char *d[15];
    while (s2 != NULL)
```

```

    {
        cout << s2 << endl;
        d[j] = s2;
        cout << d[j] << " элемент массива " << endl;
        j++;
        s2 = strtok_s(NULL, " .", &next);
    }

    cout << endl;
    for (i = 0; i < j; i++)
    {
        cout << d[i] << "  длина элемента  " << strlen(d[i]) << endl;
        r[i] = strlen(d[i]);
    }
    min1 = r[0];
    for (i = 0; i < j; i++)
    {
        if (r[i] < min1) min1 = r[i];
    }
    cout << min1 << "  минимальная длина  " << endl;
    for (i = 0; i < j; i++)
    {
        if (r[i] == min1)

            cout << r[i] << "  " << d[i] << endl;
    }
    system("pause");
    return 0;
}

```

Результат выполнения программы:

```

найти
найти элемент массива
шифр
шифр элемент массива
было
было элемент массива
невозможно
невозможно элемент массива

найти    длина элемента    5
шифр    длина элемента    4
было    длина элемента    4
невозможно    длина элемента    10
4 минимальная длина
4 шифр
4 было
Для продолжения нажмите любую клавишу . . . _

```

Пояснение к примеру: функция `strtok()` возвращает из `s1` лексему, отделенную разделителем. `s2`- указатель, ссылающийся на следующий символ, начиная с которого должен осуществляться поиск следующей лексемы. Каждый последующий вызов функции с нулевым указателем в качестве значения первого аргумента продолжает поиск лексем, начиная с того места в строке, на которое указывает возвращаемое в результате последнего вызова значение указателя. Функция возвращает нулевой указатель, если лексемы не обнаружены.

Задания для самостоятельного выполнения:

Задание 7.1 Строки

1. Задан текст, заканчивающийся точкой. Слова в тексте разделены пробелами. Сколько слов в тексте, начинается на букву «н».
2. Задан текст, заканчивающийся точкой. Слова в тексте разделены пробелами. Сколько слов, начинается с буквы `b`.
3. Задан текст, заканчивающийся точкой. Слова в тексте разделены пробелами. Сколько в тексте букв `r`, `k`, `t`.
4. Задан текст, заканчивающийся точкой. Слова в тексте разделены пробелами. Сколько в тексте символов «*», «;», «:».
5. Задан текст, заканчивающийся точкой. Слова в тексте разделены пробелами. Сколько в тексте букв `r`, `k`, `t`.
6. Задан текст, заканчивающийся точкой. Слова в тексте разделены пробелами. Найти длину самого короткого слова и самого длинного слова.
6. Задан текст, заканчивающийся точкой. Слова в тексте разделены пробелами. Вывести на экран слова, содержащие удвоенным «nn».
7. Задан текст, заканчивающийся точкой. Слова в тексте разделены пробелами. Преобразовать текст, удалив слова с удвоенным «nn».

8. Задан текст, заканчивающийся точкой. Слова в тексте разделены пробелами. Сколько, сколько раз в тексте встречается слово из 4-х букв.
9. Задан текст, заканчивающийся точкой. Слова в тексте разделены пробелами. Сколько букв k в последнем ее слове.
- 11 Задан текст, заканчивающийся точкой. Слова в тексте разделены пробелами. Сколько в тексте букв «в».
12. Задан текст, заканчивающийся точкой. В тексте есть одна открывающаяся и одна закрывающаяся скобка. Вывести на экран все символы, расположенные внутри этих скобок.
13. Задан текст, заканчивающийся точкой. Слова в тексте разделены пробелами. Найти количество слов, состоящих из пяти символов.
14. Задан текст, заканчивающийся точкой. Слова в тексте разделены пробелами. Найти слово в строке, имеющее 6 символов.
15. Задан текст, заканчивающийся точкой. Слова в тексте разделены пробелами. Определить количество символов во втором слове.
16. Задан текст, заканчивающийся точкой. Слова в тексте разделены пробелами. Найти слова с четным количеством символов.
17. Задан текст, заканчивающийся точкой. Слова в тексте разделены пробелами. Найти порядковый номер слова максимальной длины.
18. Задан текст, заканчивающийся точкой. Слова в тексте разделены пробелами. Подсчитать количество слов, начинающихся с буквы “а”.
19. Задан текст, заканчивающийся точкой. Слова в тексте разделены пробелами. Подсчитать количество слов, заканчивающихся на букву “ь”.
20. Задан текст, заканчивающийся точкой. Слова в тексте разделены пробелами. Вывести на экран порядковый номер слова «максимальной» в тексте и номер позиции в строке, с которой оно начинается.

К лабораторной работе № 8. Линейные динамические массивы

Пример 8.1. Использование динамических массивов. Найти сумму нечетных и произведение четных элементов массива из n элементов.

Порядок выполнения:

1. Пуск, программы, Visual Studio 2015, Создать проект, Visual C++, Консольное приложение Win32, имя (задать имя проекта), расположение (диск, каталог), имя решения (появится автоматически), ОК, консольное приложение, пустой проект, готово.

2. Проект (пункт меню), добавить новый элемент, файл C++ (.cpp), имя (ввести имя программы), добавить.

3. В папке Файлы исходного кода появился файл с введенным именем (имя.cpp). Набрать текст в редакторе. Выполнить: Проект, Построить решение.

4. Отладка программы. Отладка (пункт меню), начать отладку, выполнить построение, ДА. Повторное открытие проекта осуществить с помощью функции Открыть проект.

5. Ввод программного кода:

```
#include <iostream>
#include <ctime>
using namespace std;

int main()
{
    system("color 0f");
    setlocale(0, "");
    srand(time(0));
    int a(1), i, r(0), n;
    cout << "Введите размерность массива: ";
    cin >> n;
    int *mas = new int[n];
    for (i = 0; i < n; i++)
    {
        mas[i] = 0 + rand() % 20;
        cout << mas[i] << " ";
    }
}
```

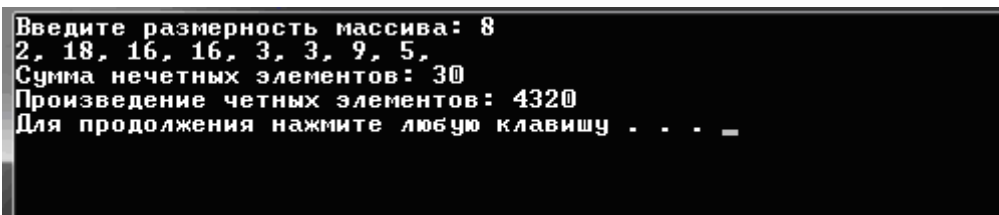
```

    }
    cout << endl;
    for (i = 0; i < n; i++)
    {
        if (i % 2 != 0)
            a = a*mas[i];
        else r = r + mas[i];
    }
    cout << "Сумма нечетных элементов: " << r << endl;
    cout << "Произведение четных элементов: " << a << endl;
    delete[] mas;
    system("pause");
    return 0;
}

```

6. Построение проекта: после ввода программного кода нужно скомпилировать и отладить программу. Для этого: Построение – Компилировать (Ctrl + F7) – Программа скомпилирована. Если в программном коде имеются ошибки, они отобразятся в нижней части окна. Чтобы увидеть, в каком месте ошибка, нужно дважды кликнуть на сообщении об ошибке. Синий маркер укажет на строку в программном коде, где имеется ошибка. Далее создаем исполняемый exe-файл: Построение – Построить решение (F7). Чтобы просмотреть результат выполнения программы, нужно выполнить: Отладка – Запуск без отладки (Ctrl+F5).

Результат выполнения программы:



```

Введите размерность массива: 8
2, 18, 16, 16, 3, 3, 9, 5,
Сумма нечетных элементов: 30
Произведение четных элементов: 4320
Для продолжения нажмите любую клавишу . . . _

```

Задания для самостоятельного выполнения:

Задание 8.1. Линейные динамические массивы

1. В массив $a[n]$ занесены натуральные числа. Найти сумму тех элементов, которые кратны заданному числу k .

2. Дана последовательность целых чисел A_1, A_2, \dots, A_n . Выяснить, какое число встречается раньше - положительное или отрицательное. Поменять местами положительные и отрицательные числа.

3. Дана последовательность действительных чисел A_1, A_2, \dots, A_n . Выяснить, будет ли она возрастающей.

4. Дана последовательность натуральных чисел A_1, A_2, \dots, A_n . Создать массив из четных чисел этой последовательности. Если таких чисел нет, то вывести сообщение.

5. При поступлении в Вуз Абитуриенты, получившие двойку на первом экзамене, ко второму не допускаются. В массиве $A[p]$ записаны оценки экзаменуемых, полученные на первом экзамене. подсчитать, сколько человек не допущено ко второму экзамену.

6. Пригодность детали оценивается по размеру B , который должен соответствовать интервалу $(A - b, A + b)$. определить, имеются ли в партии из n деталей бракованные. Если да, то подсчитать их количество, в противном случае выдать отрицательный ответ.

7. У вас есть доллары. Вы хотите обменять их на рубли. Есть информация о стоимости купли-продажи в банках города. В городе n банков. Составьте программу, определяющую, какой банк выбрать, чтобы выгодно обменять доллары на рубли.

8. Дан целочисленный массив с количеством элементов из n . Напечатать те его элементы, индексы которых являются степенями двойки (1, 2, 4, 8, 16, ...).

9. У прилавка магазина выстроилась очередь из n покупателей. Время обслуживания i -го покупателя равно t_i ($i = 1, \dots, n$). определить время S_i пребывания i -го покупателя в очереди.

10. «Суперзамок». Секретный замок для сейфа состоит из 10 расположенных в ряд ячеек, в которые надо вставить игральные кубики. Но дверь открыва-

ется только в том случае, когда в любых трех соседних ячейках сумма точек на передних гранях кубика равна 10. (игральный кубик имеет на каждой грани от 1 до 6 точек.) напишите программу, которая разгадывает код замка при условии, что два кубика уже вставлены в ячейки.

11. Дан массив из n четырехзначных натуральных чисел. Вывести на экран только те, у которых сумма первых двух цифр равна сумме двух последних.

12. Дан массив из n четырехзначных натуральных чисел. Вывести на экран только те, значение которых начинается с 1.

13. Дан массив из n четырехзначных натуральных чисел. Вывести на экран только те, у которых сумма первых двух цифр больше 9.

14. Дан массив из n четырехзначных натуральных чисел. Вывести на экран только те, у которых сумма двух последних цифр меньше 12.

15. Дан массив из n четырехзначных натуральных чисел. Вывести на экран только те, у которых сумма первых двух цифр больше сумме двух последних.

16. Дан массив из n четырехзначных натуральных чисел. Вывести на экран только те, у которых сумма первых двух цифр равна последней цифре.

17. Дан массив из n четырехзначных натуральных чисел. Вывести на экран только те, у которых сумма первых двух цифр больше последней цифры.

18. Дан массив из n четырехзначных натуральных чисел. Вывести на экран только те, у которых произведение первых двух цифр больше 15.

19. Дан массив из n четырехзначных натуральных чисел. Вывести на экран только те, у которых сумма первых двух цифр меньше сумме двух последних.

20. Дан массив из n четырехзначных натуральных чисел. Вывести на экран только те, у которых сумма первой и последней цифры равна

К лабораторной работе № 9. Двумерные динамические массивы

Пример 9.1. Двумерный динамический массив. Дана целочисленная прямоугольная матрица. Найти наименьший элемент среди нечетных столбцов этой матрицы. Вывести на экран исходный и преобразованный массив.

Порядок выполнения:

1. Пуск, программы, Visual Studio 2015, Создать проект, Visual C++, Консольное приложение Win32, имя (задать имя проекта), расположение (диск, каталог), имя решения (появится автоматически), ОК, консольное приложение, пустой проект, готово.

2. Проект (пункт меню), добавить новый элемент, файл C++ (.cpp), имя (ввести имя программы), добавить.

3. В папке Файлы исходного кода появился файл с введенным именем (имя.cpp). Набрать текст в редакторе. Выполнить: Проект, Построить решение.

4. Отладка программы. Отладка (пункт меню), начать отладку, выполнить построение, ДА. Повторное открытие проекта осуществить с помощью функции Открыть проект.

5. Ввод программного кода:

```
#include <iostream>
#include <ctime>
#include <iomanip>
using namespace std;

int main()
{
    system("color 0f");
    setlocale(0, "");
    srand(time(0));
    int i, n, m, row, col, min;
    srand(time(0));
    cout << "Введите количество строк: ";
    cin >> n;
```

```

    cout << "Введите количество столбцов: ";
    cin >> m;
    float **ptrarray = new float*[n];
    for (i = 0; i < n; i++)
        ptrarray[i] = new float[m];
    for (row = 0; row < n; row++)
    {
        for (col = 0; col < m; col++)
            ptrarray[row][col] = 10 + rand() % 30;
    }
    cout << endl;
    for (row = 0; row < n; row++)
    {
        for (col = 0; col < m; col++)
            cout << ptrarray[row][col] << " ";
        cout << endl;
    }
    min = ptrarray[0][0];
    for (col = 0; col < m; col+=2)
    {
        for (row = 0; row < n; row++)
        {
            if (ptrarray[row][col] <= min)
                min = ptrarray[row][col];
        }
    }
    cout << endl;
    cout << "Минимальный элемент среди нечетных столбцов: " << min <<
endl;
    for (i = 0; i < n; i++)
        delete[]ptrarray[i];
    system("pause");
    return 0;
}

```

6. Построение проекта: после ввода программного кода нужно скомпилировать и отладить программу. Для этого: Построение – Компилировать (Ctrl + F7) – Программа скомпилирована. Если в программном коде имеются ошибки, они отобразятся в нижней части окна. Чтобы увидеть, в каком месте ошибка, нужно дважды кликнуть на сообщении об ошибке. Синий маркер укажет на строку в программном коде, где имеется ошибка. Далее создаем исполняемый exe-файл: Построение – Построить решение (F7). Чтобы просмотреть результат

выполнения программы, нужно выполнить: Отладка – Запуск без отладки (Ctrl+F5).

Результат выполнения программы:

```
Введите количество строк: 6
Введите количество столбцов: 6

39 33 29 35 24 37
18 29 22 16 19 17
38 16 34 25 35 27
36 31 38 10 14 23
34 27 38 34 24 35
34 14 27 27 26 31

Минимальный элемент среди нечетных столбцов: 14
Для продолжения нажмите любую клавишу . . . _
```

Задание для самостоятельного выполнения:

Задание 9.1. Двумерные динамические массивы

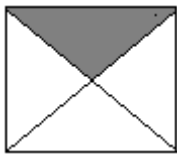
1. Дана матрица $A(n,n)$. Необходимо заменить строки квадратной матрицы ее столбцами, а столбцы – строками.
2. Дана целая матрица $A(n,n)$. Определить, является ли заданная целая матрица магическим квадратом, то есть в которой сумма всех элементов во всех строках и столбцах одинаковы.
3. Дана матрица $A(n,n)$. Упорядочить в порядке возрастания K -тую строку матрицы ($1 \leq K \leq N$).
4. Дана матрица $A(n,n)$. Включить в матрицу элементы новой K -ой строки ($1 \leq K \leq N$).
5. Дана матрица $A(n,n)$. Удалить из матрицы все нулевые элементы, сдвинув оставшиеся. Освободившиеся места занулить.
6. Дана матрица $A(n,n)$. Переставить в матрице K и L столбцы. ($1 \leq K \leq N$), ($1 \leq L \leq M$)
7. Дана матрица $A(n,n)$. Сформировать два одномерных массива. В один переслать по строкам верхний треугольник матрицы, включая элементы главной диагонали, в другой нижний треугольник. Вывести результат..

8. Дана целая матрица $A(n,n)$. Для заданной целой матрицы напечатать индексы всех ее седловых точек и их значения (элемент матрицы называется седловой точкой, если он является наименьшим в своей строке и одновременно наибольшим в своем столбце).

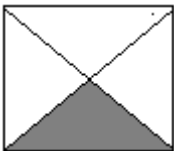
9. Дана целая матрица $A(n,n)$, значение элементов которой не повторяются. Исключить из нее строку и столбец, на пересечении которых расположен \max элемент главной диагонали.

10. Получить массив $B(n-k, m-l)$ из массива $A(n, m)$ удалением n -строки и k -ого столбца.

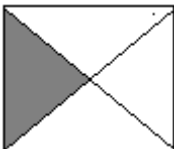
11. Дана целочисленная квадратная матрица $A(n,n)$. Вычислите сумму элементов матрицы, выделенной черным цветом. Выведите на экран исходный массив и результат вычисления.



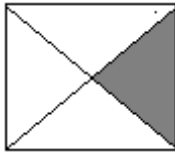
12. Дана целочисленная квадратная матрица $A(n,n)$. Вычислите сумму элементов матрицы, выделенной черным цветом. Выведите на экран исходный массив и результат вычисления.



13. Дана целочисленная квадратная матрица $A(n,n)$. Вычислите сумму элементов матрицы, выделенной черным цветом. Выведите на экран исходный массив и результат вычисления.



14. Дана целочисленная квадратная матрица $A(n,n)$. Вычислите сумму элементов матрицы, выделенной черным цветом. Выведите на экран исходный массив и результат вычисления.



15. Дана целочисленная квадратная матрица $A(n,n)$. Вычислите сумму элементов матрицы, выделенной черным цветом. Выведите на экран исходный массив и результат вычисления.



16. Дана целочисленная квадратная матрица $A(n,n)$. Вычислите сумму элементов матрицы, выделенной черным цветом. Выведите на экран исходный массив и результат вычисления.



17. Дана целочисленная квадратная матрица $A(n,n)$. Вычислите сумму элементов матрицы, выделенной черным цветом. Выведите на экран исходный массив и результат вычисления.



18. Дана целочисленная квадратная матрица $A(n,n)$. Вычислите сумму элементов матрицы, выделенной черным цветом. Выведите на экран исходный массив и результат вычисления.



18. Дана целая матрица $A(n,n)$, значение элементов которой не повторяются. Исключить из нее первую строку и первый столбец.

19. Дана целая матрица $A(n,n)$, значение элементов которой не повторяются. Включить в нее заданную строку $B(n)$ в качестве первой строки.

К лабораторной работе № 10. Функции

Пример 10.1. Пример функции, которая осуществляет передачу параметров по значению.

Порядок выполнения:

1. Пуск, программы, Visual Studio 2015, Создать проект, Visual C++, Консольное приложение Win32, имя (задать имя проекта), расположение (диск, каталог), имя решения (появится автоматически), ОК, консольное приложение, пустой проект, готово.

2. Проект (пункт меню), добавить новый элемент, файл C++ (.cpp), имя (ввести имя программы), добавить.

3. В папке Файлы исходного кода появился файл с введенным именем (имя.cpp). Набрать текст в редакторе. Выполнить: Проект, Построить решение.

4. Отладка программы. Отладка (пункт меню), начать отладку, выполнить построение, ДА. Повторное открытие проекта осуществить с помощью функции Открыть проект.

5. Ввод программного кода:

```
#include <iostream>
#include<string>
using namespace std;
void ng(int x, int y)
{
    cout<<"x= "<<x<<" y="<<y<<endl;
    x=100;
    y=200;
}
```



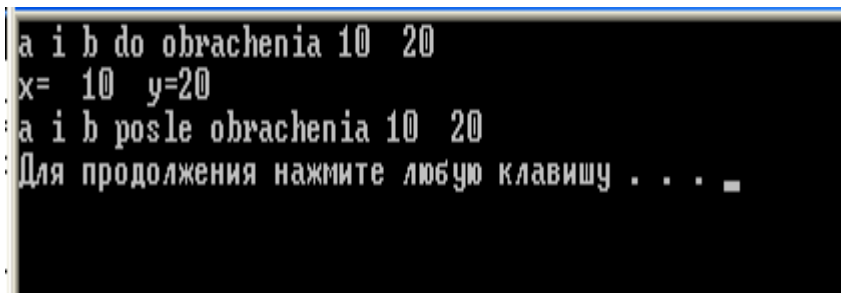
```

int main(){
    int a,b;
    a=10; b=20;
    cout<<"a i b do obrachenia "<<a<<" "<<b<<endl;
    ng( a, b);
    cout<<"a i b posle obrachenia "<<a<<" "<<b<<endl;
    system("pause");
    return 0;
}

```

6. Построение проекта: после ввода программного кода нужно скомпилировать и отладить программу. Для этого: Построение – Компилировать (Ctrl + F7) – Программа скомпилирована. Если в программном коде имеются ошибки, они отобразятся в нижней части окна. Чтобы увидеть, в каком месте ошибка, нужно дважды кликнуть на сообщении об ошибке. Синий маркер укажет на строку в программном коде, где имеется ошибка. Далее создаем исполняемый exe-файл: Построение – Построить решение (F7). Чтобы просмотреть результат выполнения программы, нужно выполнить: Отладка – Запуск без отладки (Ctrl+F5).

Результат выполнения программы:



```

a i b do obrachenia 10 20
x= 10 y=20
a i b posle obrachenia 10 20
Для продолжения нажмите любую клавишу . . .

```

Пример 10.2. Пример функции, которая осуществляет передачу параметров по ссылке.

```

// передача параметров по ссылке
#include <iostream>

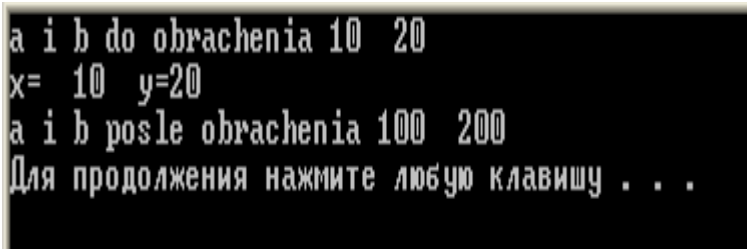
```

```

#include<string>
using namespace std;
void ng(int &x, int &y)
{
    cout<<"x= "<<x<<" y="<<y<<endl;
    x=100;
    y=200;
}
int main(){
    int a,b;
    a=10; b=20;
    cout<<"a i b do obrachenia "<<a<<" "<<b<<endl;
    ng( a, b);
    cout<<"a i b posle obrachenia "<<a<<" "<<b<<endl;
    system("pause");
    return 0;
}

```

Результат выполнения программы:



```

a i b do obrachenia 10 20
x= 10 y=20
a i b posle obrachenia 100 200
Для продолжения нажмите любую клавишу . . .

```

Пример 10.3. Пример функции, которая осуществляет передачу параметров по адресу.

```

#include <iostream>
#include<string>
using namespace std;
void ng(int *x, int *y)
{
    cout<<"x= "<<x<<" y="<<y<<endl;
    *x=100;
    *y=200;
}
int main(){
    int a,b;
    a=10; b=20;
    cout<<"a i b do obrachenia "<<a<<" "<<b<<endl;
    ng(&a,&b);
}

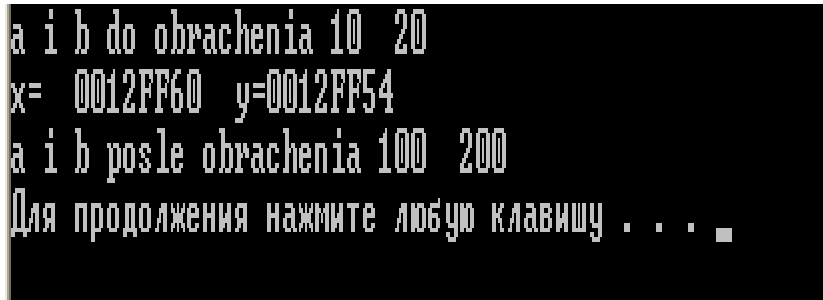
```

```

        cout<<"a i b posle obrachenia "<<a<<" "<<b<<endl;
    system("pause");
    return 0;
}

```

Результат выполнения программы:



```

a i b do obrachenia 10 20
x= 0012FF60 y=0012FF54
a i b posle obrachenia 100 200
Для продолжения нажмите любую клавишу . . .

```

Пример 10.4. Пример функции, которая осуществляет передачу всего массива в качестве параметров по ссылке.

```

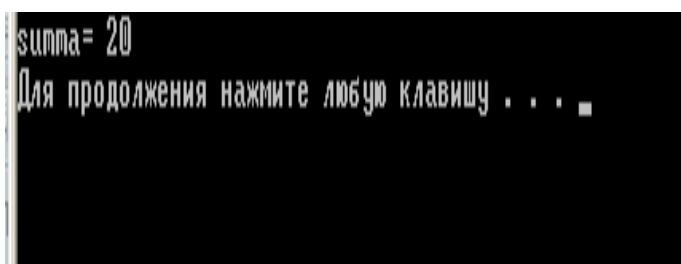
#include <iostream>
using namespace std;
int sum1( int*, int );
//int sum1(int mas[], int n);
//int sum1(int mas[n], int n);
int const n=10;
int main()
{
    int marks[ n ] = {3,4,5,4,4};
    cout<<"summa= "<<sum1(marks,5)<<endl;
    system("pause");
    return 0;
}

int sum1( int* mas, int n)
// int sum1(int mas[], int n)
//int sum1(int mas[n], int n)

    {int s=0;
    for (int i=0; i<n; i++)
        s+=mas[i];
    return s;
}

```

Результат выполнения программы:



```

summa= 20
Для продолжения нажмите любую клавишу . . .

```

Пример 10.5. Пример функции, которая осуществляет передачу всего массива в качестве параметров по значению.

```
#include <iostream>
using namespace std;
void element(int);
int sum1( int*, int );
//int sum1(int mas[], int n);
//int sum1(int mas[n], int n);
int const n=5;
int main()
{
    int marks[ n ] = {3,4,5,4,4};
    for (int i=0; i<n; i++)
        cout<<"element massiva: "<<marks[i]<<endl;
    // передача всего массива по ссылке
    cout<<"summa= "<<sum1(marks,5)<<endl;
    //передача элемента массива по значению
    cout<<"2-oi element ="<<marks[2]<<endl;
    element(marks[2]);
    cout<<"2 - oi element ="<<marks[2]<<endl;
    system("pause");
    return 0;}
int sum1( int* mas, int n)
// int sum1(int mas[], int n)
//int sum1(int mas[n], int n)

    {int s=0;
    for (int i=0; i<n; i++)
        s+=mas[i];
    return s;
}
void element(int k)
{cout<<"element v funczii="<<k*2<<endl;
}
```

Результат выполнения программы:

```
element massiva: 3
element massiva: 4
element massiva: 5
element massiva: 4
element massiva: 4
summa= 20
2-oi element =5
element v funczii=10
2 - oi element =5
Для продолжения нажмите любую клавишу . . .
```

Задания для самостоятельного выполнения:

Задание 10.1. Функции

1. Описать функцию вычисления количества отрицательных чисел в массиве W из N элементов. Выполнить обращение к функции.
2. Описать функцию вычисления количества положительных чисел в массиве W из N элементов. Выполнить обращение к функции.
3. Описать функцию вычисления количества чётных чисел в массиве W из N элементов. Выполнить обращение к функции.
4. Описать функцию вычисления количества нечётных чисел в массиве W из N элементов. Выполнить обращение к функции.
5. Вариант 12. Описать функцию вычисления количества чисел, делящихся на 3 без остатка в массиве W из N элементов. Выполнить обращение к функции.
6. Описать функцию вычисления количества чисел, делящихся на 7 без остатка в массиве из N элементов. Выполнить обращение к функции.
7. Описать функцию вычисления количества чисел, делящихся на 5 без остатка в массиве W из N элементов. Выполнить обращение к функции.
8. Описать функцию, нахождения суммы отрицательных чисел, в массиве W из N элементов. Выполнить обращение к функции.
9. Описать функцию, нахождения суммы положительных чисел, в массиве W из N элементов. Выполнить обращение к функции.
10. Описать функцию, нахождения произведения отрицательных чисел, в массиве W из N элементов. Выполнить обращение к функции.
11. Описать функцию вычисления количества нечетных положительных чисел в массиве W из N элементов. Выполнить обращение к функции.
12. Описать функцию вычисления количества чисел, больших «5» в массиве W из N элементов. Выполнить обращение к функции.

13. Описать функцию вычисления количества чётных чисел в массиве W из N элементов. Выполнить обращение к функции.

14. Описать функцию вычисления количества ненулевых чисел в массиве W из N элементов. Выполнить обращение к функции.

15. Вариант 12. Описать функцию вычисления количества чисел, делящихся на 7 без остатка в массиве W из N элементов. Выполнить обращение к функции.

16. Описать функцию вычисления количества чисел, делящихся на 4 без остатка в массиве W из N элементов. Выполнить обращение к функции.

17. Описать функцию вычисления количества чисел, делящихся на 6 без остатка в массиве W из N элементов. Выполнить обращение к функции.

18. Описать функцию, нахождения суммы четных чисел, в массиве W из N элементов. Выполнить обращение к функции.

19. Описать функцию, нахождения суммы отрицательных чисел, в массиве W из N элементов. Выполнить обращение к функции.

20. Описать функцию, нахождения произведение четных чисел, в массиве W из N элементов. Выполнить обращение к функции.

К лабораторной работе № 11. Структуры

Пример 11.1. Разработка структуры

Примечание: во всех вариантах при обращении к элементам структуры должны использоваться как приемы непосредственного обращения к одним элементам структуры, так и через указатель к другим.

Опишите структуру FLAT (количество комнат, площадь, этаж, адрес, цена). Введите клавиатуры данные в массив, состоящий из пяти элементов типа FLAT. Выведите на экран информацию о квартирах, стоимость которых не больше введенного с клавиатуры числа. Если такой

квартиры нет, то программа должна выдать соответствующее сообщение на экран.

Порядок выполнения:

1. Пуск, программы, Visual Studio 2015, Создать проект, Visual C++, Консольное приложение Win32, имя (задать имя проекта), расположение (диск, каталог), имя решения (появится автоматически), ОК, консольное приложение, пустой проект, готово.

2. Проект (пункт меню), добавить новый элемент, файл C++ (.cpp), имя (ввести имя программы), добавить.

3. В папке Файлы исходного кода появился файл с введенным именем (имя.cpp). Набрать текст в редакторе. Выполнить: Проект, Построить решение.

4. Отладка программы. Отладка (пункт меню), начать отладку, выполнить построение, ДА. Повторное открытие проекта осуществить с помощью функции Открыть проект.

5. Ввод программного кода:

```
#include "stdafx.h"
```

```
#include <math.h>
```

```
#include <iostream>
```

```
#include <Windows.h>
```

```
using namespace std;
```

```
struct address {  
    char street[200];  
    int house;  
};
```

```
struct FLAT {  
    int rooms;  
    float area;  
    int floor;  
    address ars;  
    float price;  
    void print() {  
        cout << "Кол-во комнат: " << rooms << "\nПлощадь: " << area << "\nЭтаж:  
    " << floor;  
        cout << "\nАдрес: " << ars.street << ", " << ars.house << "\nСтоимость: " <<  
    price << " тыс. руб." << endl;  
    }  
};
```

```
void Enter_Data(FLAT blds[], int cnt_blds) {  
    for (int i = 0; i < cnt_blds; i++) {  
        cout << "Введите данные " << i+1 << " квартиры:\n";
```

```

        cout << "Кол-во комнат: "; cin >> blds[i].rooms;
        cout << "Площадь: "; cin >> blds[i].area;
        cout << "Этаж: "; cin >> blds[i].floor;
        cout << "Улица: "; cin >> blds[i].ars.street;
        cout << "Дом: "; cin >> blds[i].ars.house;
        cout << "Стоимость (тыс. руб.): "; cin >> blds[i].price;
        cout << "-----" << endl;
    }
}

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    setlocale(0, "Russian");
    int cnt_blds;
    float price;
    bool exst = false;

    cout << "Введите кол-во квартир: "; cin >> cnt_blds; cout << endl;
    FLAT *blds = new FLAT [cnt_blds];

    Enter_Data(blds, cnt_blds);
    system("cls");
    do {
        cout << "Введите стоимость квартиры: "; cin >> price;
        if (price < 0)
            cout << "Некорректное значение!\n";
    } while (price < 0);
    cout << "\nКвартиры, стоимость которых не превышает " << price << " тыс.
руб.:" << endl;

    for (int i = 0; i < cnt_blds; i++) {
        if (blds[i].price <= price) {
            cout << endl << i + 1 << " квартира:\n";
            blds[i].print();
            cout << "-----";
            exst = true;
        }
    }
    if (!exst)
        cout << "\nКвартир не найдено!\n-----";
    cout << endl;
    system("pause");
    return 0;
}

```


6. Построение проекта: после ввода программного кода нужно скомпилировать и отладить программу. Для этого: Построение – Компилировать (Ctrl + F7) – Программа скомпилирована. Если в программном коде имеются ошибки, они отобразятся в нижней части окна. Чтобы увидеть, в каком месте ошибка, нужно дважды кликнуть на сообщении об ошибке. Синий маркер укажет на строку в программном коде, где имеется ошибка. Далее создаем исполняемый exe-файл: Построение – Построить решение (F7). Чтобы просмотреть результат выполнения программы, нужно выполнить: Отладка – Запуск без отладки (Ctrl+F5).

Результат выполнения программы:

```
Введите кол-во квартир: 5
Введите данные 1 квартиры:
Кол-во комнат: 1
Площадь: 26.4
Этаж: 2
Улица: Нахимова
Дом: 84
Стоимость (тыс. руб.): 850
-----
Введите данные 2 квартиры:
Кол-во комнат: 4
Площадь: 144.8
Этаж: 22
Улица: Современная
Дом: 101
Стоимость (тыс. руб.): 30000
-----
Введите данные 3 квартиры:
Кол-во комнат: 3
Площадь: 66.2
Этаж: 8
Улица: Коммунистическая
Дом: 12
Стоимость (тыс. руб.): 3000
-----
Введите данные 4 квартиры:
Кол-во комнат: 3
Площадь: 74.9
Этаж: 4
Улица: Интернациональная
Дом: 43
Стоимость (тыс. руб.): 3360
-----
Введите данные 5 квартиры:
Кол-во комнат: 2
Площадь: 42.5
Этаж: 9
Улица: Луганская
Дом: 30
Стоимость (тыс. руб.): 1950

Введите стоимость квартиры: 3000
Квартиры, стоимость которых не превышает 3000 тыс. руб.:
1 квартира:
Кол-во комнат: 1
Площадь: 26.4
Этаж: 2
Адрес: Нахимова, 84
Стоимость: 850 тыс. руб.
-----
3 квартира:
Кол-во комнат: 3
Площадь: 66.2
Этаж: 8
Адрес: Коммунистическая, 12
Стоимость: 3000 тыс. руб.
-----
5 квартира:
Кол-во комнат: 2
Площадь: 42.5
Этаж: 9
Адрес: Луганская, 30
Стоимость: 1950 тыс. руб.
-----
Для продолжения нажмите любую клавишу . . . ■
```

```
Введите стоимость квартиры: -2
Некорректное значение!
Введите стоимость квартиры: 1

Квартиры, стоимость которых не превышает 1 тыс. руб.:

Квартир не найдено!
-----
Для продолжения нажмите любую клавишу . . . ■
```

Задание для самостоятельного выполнения:

Задание 11.1. Структуры

1. Опишите структуру ПЕРЕВОЗКИ (тип самолета, количество рейсов, налет в часах, налет в тысячах километров). Введите клавиатуры данные в массив, состоящий из шести элементов типа ПЕРЕВОЗКИ. Выведите на экран информацию о самолетах, налет часов которых больше введенного с клавиатуры числа. Если такого самолета нет, то программа должна выдать соответствующее сообщение на экран.

2. Опишите структуру КВАРТИРА (адрес, этаж, количество комнат, площадь, фирма по продаже недвижимости, ФИО менеджера). Введите клавиатуры данные в массив, состоящий из пяти элементов типа КВАРТИРА. Выведите на экран информацию о квартирах, расположенных на первом этаже введенного с клавиатуры города. Если такой квартиры нет, то программа должна выдать соответствующее сообщение на экран.

3. Опишите структуру АВТО (марка машины, цвет, цена, максимальная скорость). Введите клавиатуры данные в массив, состоящий из шести элементов типа АВТО. Выведите на экран информацию о машинах, скорость которых меньше введенного с клавиатуры числа. Если такой машины нет, то программа должна выдать соответствующее сообщение на экран.

4. Опишите структуру ТЕЛЕФОН (ФИО абонента, номер телефона, плата за телефон, год установки). Введите клавиатуры данные в массив, состоящий из пяти элементов типа ТЕЛЕФОН. Выведите на экран информацию о телефонах, год установки которых больше введенного с клавиатуры числа. Если такого телефона нет, то программа должна выдать соответствующее сообщение на экран.

5. Опишите структуру МУЗЕИ (название музея, время работы, стоимость билета, адрес). Введите клавиатуры данные в массив, состоящий из шести элементов типа МУЗЕИ. Выведите на экран информацию о музеях, стоимость билета в которых меньше введенного с клавиатуры числа. Если такого музея нет, то программа должна выдать соответствующее сообщение на экран.

6. Опишите структуру ТУРИСТ (страна, продолжительность тура, транспорт, цена). Введите клавиатуры данные в массив, состоящий из семи элементов типа ТУРИСТ. Выведите на экран информацию о странах, стоимость билета в которые меньше введенного с клавиатуры числа. Если такой страны нет, то программа должна выдать соответствующее сообщение на экран.

7. Опишите структуру ПРАЙС (название товара, цена, название магазина). Введите клавиатуры данные в массив, состоящий из семи элементов типа ПРАЙС. Выведите на экран информацию о товарах, продающихся в магазине, название которого введено с клавиатуры. Если такого магазина нет, то программа должна выдать соответствующее сообщение на экран.

8. Опишите структуру СКЛАД (наименование, фирма-изготовитель, количество на складе, цена). Введите клавиатуры данные в массив, состоящий из пяти элементов типа СКЛАД. Выведите на экран информацию о холодильниках, стоимость которых не больше введенного с клавиатуры числа. Если такого холодильника нет, то программа должна выдать соответствующее сообщение на экран.

9. Опишите структуру ГОРОСКОП (знак зодиака, фамилия и имя, день рождения). Введите клавиатуры данные в массив, состоящий из семи элементов типа ГОРОСКОП. Выведите на экран фамилии, имена и дни рождения всех «стрельцов». Если таких нет, то программа должна выдать соответствующее сообщение на экран.

10. Опишите структуру СТУДЕНТ (название учебного заведения, фамилия и инициалы студента, номер группы, средний балл за учебный год). Введите клавиатуры данные в массив, состоящий из шести элементов типа СТУДЕНТ. Выведите на экран информацию о студентах, у которых средний балл выше введенного с клавиатуры числа. Если таких данных нет, то программа должна выдать соответствующее сообщение на экран.

11. Опишите структуру СПОРТ (фамилия и инициалы спортсмена, вид спорта, возраст, количество лет в секции). Введите клавиатуры данные в массив, состоящий из шести элементов типа СПОРТ. Выведите на экран информацию о спортсменах, у которых возраст выше введенного с клавиатуры числа. Если таких данных нет, то программа должна выдать соответствующее сообщение на экран.

12. Опишите структуру МУЗ_ШКОЛА (фамилия и инициалы учащегося, вид инструмента, возраст, класс). Введите клавиатуры данные в массив, состоящий из шести элементов типа МУЗ_ШКОЛА. Выведите на экран информацию об учащихся, у которых возраст выше введенного с клавиатуры числа. Если таких данных нет, то программа должна выдать соответствующее сообщение на экран.

13. Опишите структуру РАБОТА (фамилия и инициалы сотрудника, стаж работы, возраст, заработная плата). Введите клавиатуры данные в массив, состоящий из шести элементов типа РАБОТА. Выведите на экран информацию о сотрудниках, у которых стаж работы выше введенного с клавиатуры числа. Ес-

ли таких данных нет, то программа должна выдать соответствующее сообщение на экран.

14. Опишите структуру ДЕТ_САД (фамилия и инициалы ребенка, адрес, возраст, группа). Введите клавиатуры данные в массив, состоящий из шести элементов типа ДЕТ_САД. Выведите на экран информацию о детях, ФИО которых начинается на букву «К». Если таких данных нет, то программа должна выдать соответствующее сообщение на экран.

15. Опишите структуру ШКОЛА (фамилия и инициалы учителя, стаж работы, возраст, предмет). Введите клавиатуры данные в массив, состоящий из шести элементов типа ШКОЛА. Выведите на экран информацию об учителях, стаж работы которых выше введенного с клавиатуры числа. Если таких данных нет, то программа должна выдать соответствующее сообщение на экран.

16. Опишите структуру ОЛИМПИАДА (фамилия и инициалы учащегося, предмет, номер школы, балл). Введите клавиатуры данные в массив, состоящий из шести элементов типа ОЛИМПИАДА. Выведите на экран информацию об учащихся, у которых балл выше введенного с клавиатуры числа. Если таких данных нет, то программа должна выдать соответствующее сообщение на экран.

17. Опишите структуру ПЕРЕПИСЬ (фамилия и инициалы гражданина, адрес, возраст, год рождения). Введите клавиатуры данные в массив, состоящий из шести элементов типа ПЕРЕПИСЬ. Выведите на экран информацию о гражданах, у которых год рождения равен введенному с клавиатуры числу. Если таких данных нет, то программа должна выдать соответствующее сообщение на экран.

18. Опишите структуру АПТЕКА (наименование лекарственного препарата, фирма-изготовитель, количество на складе, цена). Введите клавиатуры данные в массив, состоящий из пяти элементов типа АПТЕКА. Выведите на экран информацию о количестве лекарственных препаратов, наименование ко-

торых равно введенному наименованию с клавиатуры. Если такого препарата нет, то программа должна выдать соответствующее сообщение на экран.

19. Опишите структуру МНОГОБОРЬЕ (фамилия и инициалы спортсмена, вид спорта, возраст, количество очков). Введите клавиатуры данные в массив, состоящий из шести элементов типа МНОГОБОРЬЕ. Выведите на экран информацию о спортсменах, у которых количество очков выше среднего. Если таких данных нет, то программа должна выдать соответствующее сообщение на экран.

20. Опишите структуру ТЕСТИРОВАНИЕ (фамилия и инициалы учащегося, предмет балл, класс). Введите клавиатуры данные в массив, состоящий из шести элементов типа ТЕСТИРОВАНИЕ. Выведите на экран информацию об учащихся, у которых балл выше среднего балла. Если таких данных нет, то программа должна выдать соответствующее сообщение на экран.

К лабораторной работе № 12. Файлы

Пример 12.1.

Создать на МД файл, записи которого имеют следующую структуру:

- **лицевой счёт;**
- **оплата за газ (руб.);**
- **оплата за мусор (руб.).**

Вывести следующий документ:

Оплата за коммунальные услуги за ____ месяц

Лицевой счёт	Оплата за газ (руб.)	Оплата за мусор (руб.)	Всего
Итого	√	√	√

Порядок выполнения:

1. Пуск, программы, Visual Studio 2015, Создать проект, Visual C++, Консольное приложение Win32, имя (задать имя проекта), расположение

(диск, каталог), имя решения (появится автоматически), ОК, консольное приложение, пустой проект, готово.

2. Проект (пункт меню), добавить новый элемент, файл C++ (.cpp), имя (ввести имя программы), добавить.

3. В папке Файлы исходного кода появился файл с введенным именем (имя.cpp). Набрать текст в редакторе. Выполнить: Проект, Построить решение.

4. Отладка программы. Отладка (пункт меню), начать отладку, выполнить построение, ДА. Повторное открытие проекта осуществить с помощью функции Открыть проект.

5. Ввод программного кода:

```
#include "stdafx.h"
#include <iostream>
#include <Windows.h>
#include <fstream>
#include <iomanip>
#include <lmcons.h>

using namespace std;

struct INVENT {
    char account[10] = "0000";
    float gaz = 100000.99;
    float musor = 100000.86;
    float sum = 100000.99;
    void ent_data() {
        cout << "Введите лицевой счет: "; cin.ignore(); cin.getline(account, 50);
        if (strlen(account) > 4) {
            account[4] = NULL;
        }
        cout << "Введите плату за газ: "; cin >> gaz;
        cout << "Введите плату за мусор: "; cin >> musor;
        sum = gaz + musor;
    }
};

void ChoiceCreate(char path[]);
void ChoiceShow(char path[]);
void ChoiceRemove(char path[]);

void main()
{
```

```

SetConsoleCP(1251);
SetConsoleOutputCP(1251);
setlocale(0, "Russian");
char namefile[10], raz[] = ".txt", path[70] = "C:\\\\users\\", path_next[] = "\\Desktop\\";
int choice;
cout << "1 – создание файла;\n2 – просмотр содержимого файла;\n3 – удаление
файла;\n4 – выход;";
do {
    cout << "\nВыберите действие: "; cin >> choice;
    if (choice < 1 || choice > 4)
        cout << "Некорректное значение!!!";
} while (choice < 1 || choice > 4);
if(choice == 4)
    exit(0);
TCHAR urnm[UNLEN + 1];
DWORD urnm_len = UNLEN + 1;
GetUserName((TCHAR*)urnm, &urnm_len);
char name[257];
wcstombs_s(0, name, urnm, 257);
cout << "Введите название файла: "; cin.ignore(); cin.getline(namefile, 10);
strcat_s(path, 70, name);
strcat_s(path, 70, path_next);
strcat_s(path, 70, namefile);
strcat_s(path, 70, raz);

switch (choice) {
case 1:
    ChoiceCreate(path);
    break;
case 2:
    ChoiceShow(path);
    break;
case 3:
    ChoiceRemove(path);
    break;
}
system("pause");
}
void ChoiceCreate(char path[]){
    cout.fill('-');
    int cnt_elm;
    float sum_gaz = 0, sum_musor = 0;
    cout << "Введите кол-во материалов: "; cin >> cnt_elm;
    INVENT *elms = new INVENT[cnt_elm];
    for (int i = 0; i < cnt_elm; i++) {
        cout << "Введите данные " << i + 1 << " абонента:\n";
        elms[i].ent_data();
        cout << setw(35) << "\n";
    }
}

```



```

ofstream file(path);
file.precision(2);
if (!file.is_open())
    cout << "ERROR! FILE IS NOT OPEN!!!";
else {
    file << "\n\t\tОплата за коммунальные услуги за _____ месяц\n\n";
    file << "\t\tЛицевой\t\t\tОплата за\t\t\tОплата за\t\t\tВсего\n";
    file << "\t\tс чем\t\t\tгаз(руб.)\t\t\tмусор (руб.)\t\n";
    for (int i = 0; i < cnt_elm; i++) {
        file << "\t\t" << elms[i].account << "\t\t" << fixed << elms[i].gaz << "
\t\t" << fixed << elms[i].musor << " \t\t" << fixed << elms[i].sum << endl;;
        sum_gaz += elms[i].gaz;
        sum_musor += elms[i].musor;
    }
    file << "\n\t\tИтого\t\t\t" << sum_gaz << " \t\t" << sum_musor << " \t\t" <<
sum_gaz+sum_musor << endl;
    cout << "Данные успешно записаны!\n" << setw(35) << "\n";
    file.close();
}
}

void ChoiceShow(char path[]) {
    char str[200];
    ifstream file(path);
    if (!file.is_open()) {
        cout.fill('-'); cout << setw(63) << "\nERROR! FILE IS NOT FOUND!!!\n" <<
setw(35) << "\n";
    }
    else {
        while (!file.eof()) {
            file.getline(str, 200);
            cout << str << endl;
        }
    }
    file.close();
}

void ChoiceRemove(char path[]) {
    cout.fill('-');
    ifstream fin(path);
    if (!fin.is_open()) {
        cout << setw(63) << "\nERROR! FILE IS NOT FOUND!!!\n" << setw(35) <<
"\n";
        fin.close();
    }
    else {
        fin.close();
        remove(path);
        cout << setw(56) << "\nФайл успешно удален!\n" << setw(35) << "\n";
    }
}
}

```

6. Построение проекта: после ввода программного кода нужно скомпилировать и отладить программу. Для этого: Построение – Компилировать (Ctrl + F7) – Программа скомпилирована. Если в программном коде имеются ошибки, они отобразятся в нижней части окна. Чтобы увидеть, в каком месте ошибка, нужно дважды кликнуть на сообщении об ошибке. Синий маркер укажет на строку в программном коде, где имеется ошибка. Далее создаем исполняемый exe-файл: Построение – Построить решение (F7). Чтобы просмотреть результат выполнения программы, нужно выполнить: Отладка – Запуск без отладки (Ctrl+F5).

Результат выполнения программы:

```
Введите кол-во квартир: 5
Введите данные 1 квартиры:
Кол-во комнат: 1
Площадь: 26.4
Этаж: 2
Улица: Нахимова
Дом: 84
Стоимость (тыс. руб.): 850
-----
Введите данные 2 квартиры:
Кол-во комнат: 4
Площадь: 144.8
Этаж: 22
Улица: Современная
Дом: 101
Стоимость (тыс. руб.): 30000
-----
Введите данные 3 квартиры:
Кол-во комнат: 3
Площадь: 66.2
Этаж: 8
Улица: Коммунистическая
Дом: 12
Стоимость (тыс. руб.): 3000
-----
Введите данные 4 квартиры:
Кол-во комнат: 3
Площадь: 74.9
Этаж: 4
Улица: Интернациональная
Дом: 43
Стоимость (тыс. руб.): 3360
-----
Введите данные 5 квартиры:
Кол-во комнат: 2
Площадь: 42.5
Этаж: 9
Улица: Луганская
Дом: 30
Стоимость (тыс. руб.): 1950
```

```

1 - создание файла;
2 - просмотр содержимого файла;
3 - удаление файла;
4 - выход;
Выберите действие: 1
Введите название файла: счета
Введите кол-во материалов: 3

Введите данные 1 абонента:
Введите лицевой счет: 1211
Введите плату за газ: 30.33
Введите плату за мусор: 15.05
-----
Введите данные 2 абонента:
Введите лицевой счет: 1415
Введите плату за газ: 42.17
Введите плату за мусор: 14.03
-----
Введите данные 3 абонента:
Введите лицевой счет: 8721
Введите плату за газ: 135.70
Введите плату за мусор: 67.22
-----
Данные успешно записаны!
-----
Для продолжения нажмите любую клавишу . . .

```

Оплата за коммунальные услуги за _____ месяц				
Лицевой счет	Оплата за газ(руб.)	Оплата за мусор (руб.)	Всего	
1211	30.33	15.05	45.38	
1415	42.17	14.03	56.20	
8721	135.70	67.22	202.92	
Итого	208.20	96.30	304.50	

Задания для самостоятельного выполнения:

Задание 12.1. Работа с файлами.

Создание файла. Выбор данных из файла. В задаче предусмотреть пункты меню:

- 1 – создание файла**
2- просмотр содержимого файла
3 – удаление файла
4 - выход

1. Создать на МД файл, записи которого имеют следующую структуру:

- дата;
- наименование материала;
- количество (шт).

Вывести следующий документ:

Поступление материалов на склад _____

Дата	Наименование материала	Количество (шт.)
15.03	Гвоздь	100
16.04	Шуруп	200
.....

2. Создать на МД файл, записи которого имеют следующую структуру:

- лицевой счёт;
- оплата за газ (руб);
- оплата за мусор (руб).

Вывести следующий документ:

Оплата за коммунальные услуги за _____ месяц
 ФИО, адрес

Лицевой счёт	Оплата за газ (руб)	Оплата за мусор (руб)	Всего
1211	30.33	15.05	<input type="checkbox"/>
1415	42.17	14.03	<input type="checkbox"/>
.....	
Итого	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

3. Создать на МД файл, записи которого имеют следующую структуру:

- наименование материала;
- цена (руб) за единицу;
- количество (шт).

Вывести следующий документ:

Ведомость стоимости материалов

фирма, дата

Наименование материала	Цена за единицу (руб)	Количество (шт)	Сумма
Шуруп	10	5	<input type="checkbox"/>
Гвоздь	20	10	<input type="checkbox"/>
.....	
Итого			<input type="checkbox"/>

4. Создать на МД файл, записи которого имеют следующую структуру:

- код изделия;
- издержки производства (руб);
- выпущено изделий (шт);
- себестоимость единицы изделия (руб).

Вывести следующий документ:

Ведомость издержек производства

Фирма, дата

Код изделия	Издержки производства (руб)	Выпущено изделий (руб)	Себестоимость единицы изделия (руб)
3245	231.54	453	<input type="checkbox"/>
6578	235.00	1265	<input type="checkbox"/>
.....	

5. Создать на МД файл, записи которого имеют следующую структуру:

- код культуры;
- валовый сбор урожая;
- посевная площадь (га);
- средняя урожайность с 1 га (ц).

Вывести следующий документ:

Урожайность культур по хозяйствам

Район, дата

Код культуры	Валовый сбор (ц)	Посевная площадь (га)	Средняя урожайность с 1 га (ц)
Ячмень	6574.00	129.50	<input type="checkbox"/>
Кукуруза	9800.50	100.00	<input type="checkbox"/>
.....	

6. Создать на МД файл, записи которого имеют следующую структуру:

- виды оплаты;
- ИТР (руб);
- рабочие (руб);
- всего (руб).

Вывести следующий документ:

Ведомость начислений по зарплате
Фирма, дата

Виды оплаты	ИТР (руб)	Рабочие (руб);	Всего (руб).
Основная зарплата	500.00	785.50	<input type="checkbox"/>
Премия	50.75	20.00	<input type="checkbox"/>
.....			
Итого	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

7. Создать на МД файл, записи которого имеют следующую структуру:

- артикул;
- цена (руб);
- количество;
- сумма (руб).

Вывести следующий документ:

Ведомость реализации товаров
Фирма, дата

Артикул	Цена (руб)	Количество	Сумма (руб)
434	3.62	1600	<input type="checkbox"/>
121	4.12	2400	<input type="checkbox"/>
.....	
Итого			<input type="checkbox"/>

8. Создать на МД файл, записи которого имеют следующую структуру:

- номенклатурный номер детали;
- количество деталей на комплект;
- количество комплектов;
- общая потребность в деталях.

Вывести следующий документ:

Комплектовочная ведомость
Фирма, дата

Номенклатурный но- мер детали	Количество деталей на комплект	Количество комплек- тов	Общая потребность в деталях
23232	342	75	<input type="checkbox"/>
23678	654	86	<input type="checkbox"/>
.....	

9. Создать на МД файл, записи которого имеют следующую структуру:

- номер детали, узла;
- количество деталей, узлов;

- цена (руб).

Вывести следующий документ:

Инвентаризационная ведомость
Фирма, дата

Номер детали, узла	Количество деталей, узлов	Цена (руб)	Сумма (руб)
5453	324	1.54	<input type="checkbox"/>
1324	163	3.62	<input type="checkbox"/>
.....	

10. Создать на МД файл, записи которого имеют следующую структуру:

- код продукта;
- количество кг.;
- цена (руб).

Вывести следующий документ:

Ведомость отпуска продуктов
Дата____ фирма_____

Код продукта	Количество	Цена (руб)
5453	324	1.54
1324	163	3.62
.....

11. Создать на МД файл, записи которого имеют следующую структуру:

- дата;
- наименование материала;
- количество (шт).

Вывести следующий документ:

Отпуск материалов со склада _____

Дата	Наименование материала	Количество (шт.)
15.03	Гвоздь	100
16.04	Шуруп	200
.....

12. Создать на МД файл, записи которого имеют следующую структуру:

- лицевой счёт;
- оплата за домашний телефон (руб);

- оплата за мобильный телефон (руб).
Вывести следующий документ:

Оплата за телефон за __ год			
ФИО _____			
Адрес _____			
Месяц	Оплата за домаш- ний телефон (руб)	Оплата за мобильный теле- фон (руб)	Всего
январь	300.00	15.05	<input type="checkbox"/>
февраль	164.88	14.03	<input type="checkbox"/>
.....	
Итого	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

13. Создать на МД файл, записи которого имеют следующую структуру:

- наименование материала;
- цена (руб) за единицу;
- фирма.

Вывести следующий документ:

Ведомость стоимости продуктов (Прайс-лист)		
Дата, Торговая точка		
Наименование продукта	Цена за единицу (руб)	Фирма
Колбаса ливерная	340.00	ИП Петров В.А.
Сыр Российский		
....		

14. Создать на МД файл, записи которого имеют следующую структуру:

- код изделия;
- год выпуска;
- выпущено изделий (шт).

Вывести следующий документ:

Ведомость выпуска продукции		
Фирма		
Код изделия	Год выпуска	Выпущено изделий (руб.)
3245	2012	453
6578	2017	1265
.....

15. Создать на МД файл, записи которого имеют следующую структуру:

- код культуры;
- валовый сбор урожая;
- посевная площадь (га);
- наименование хозяйства.

Вывести следующий документ:

Урожайность культур по хозяйствам			
дата			
Код культуры	Валовый сбор (ц)	Посевная площадь (га)	Наименование хозяйства
Ячмень	6574.00	129.50	ООО «Колосок»
Кукуруза	9800.50	100.00	
.....	

16. Создать на МД файл, записи которого имеют следующую структуру:

- виды оплаты;
- ИТР (руб);
- рабочие (руб);
- всего (руб).

Вывести следующий документ:

Ведомость начислений по зарплате			
Фирма, дата			
Виды оплаты	ИТР (руб)	Рабочие (руб);	Всего (руб).
Основная зарплата	500.00	785.50	<input type="checkbox"/>
Премия	50.75	20.00	<input type="checkbox"/>
.....			
Итого	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

17. Создать на МД файл, записи которого имеют следующую структуру:

- артикул;
- цена (руб);
- количество;
- сумма (руб).

Вывести следующий документ:

Ведомость реализации товаров			
Фирма, дата			
Артикул	Цена (руб)	Количество	Сумма (руб)
434	3.62	1600	<input type="checkbox"/>
121	4.12	2400	<input type="checkbox"/>
.....	

Итого

☐

18. Создать на МД файл, записи которого имеют следующую структуру:

- номенклатурный номер детали;
- количество деталей на комплект;
- количество комплектов;
- общая потребность в деталях.

Вывести следующий документ:

Комплектовочная ведомость

Фирма, дата

Номенклатурный номер детали	Количество деталей на комплект	Количество комплектов	Общая потребность в деталях
23232	342	75	<input type="checkbox"/>
23678	654	86	<input type="checkbox"/>
.....	

Итого

☐

19. Создать на МД файл, записи которого имеют следующую структуру:

- номер детали, узла;
- количество деталей, узлов;
- цена (руб);
- сумма (руб).

Вывести следующий документ:

Инвентаризационная ведомость

Фирма, отдел, дата

Номер детали, узла	Количество деталей, узлов	Цена (руб)	Сумма (руб)
5453	324	1.54	<input type="checkbox"/>
1324	163	3.62	<input type="checkbox"/>
.....	

Итого

☐

20. Создать на МД файл, записи которого имеют следующую структуру:

- шифр продукта;
- количество кг.;
- цена (руб);
- сумма (руб).

Вывести следующий документ:

Ведомость отпуска продуктов

Фирма, дата

Шифр продукта	Количество кг.	Цена (руб)	Сумма (руб)
0095	75.9	1.12	<input type="checkbox"/>
0123	63.5	5.40	<input type="checkbox"/>
.....	
Итого			<input type="checkbox"/>

К лабораторной работе № 13. Классы

Пример 13.1. Создание класса (метод: нахождение площади прямоугольника; свойства класса – стороны прямоугольника).

Порядок выполнения:

1. Пуск, программы, Visual Studio 2015, Создать проект, Visual C++, Консольное приложение Win32, имя (задать имя проекта), расположение (диск, каталог), имя решения (появится автоматически), ОК, консольное приложение, пустой проект, готово.
2. Проект (пункт меню), добавить новый элемент, файл C++ (.cpp), имя (ввести имя программы), добавить.
 3. В папке Файлы исходного кода появился файл с введенным именем (имя.cpp). Набрать текст в редакторе. Выполнить: Проект, Построить решение.
 4. Отладка программы. Отладка (пункт меню), начать отладку, выполнить построение, ДА. Повторное открытие проекта осуществить с помощью функции Открыть проект.
 5. Ввод программного кода:

```
#include <iostream> //библиотеки для ввода-вывода данных  
#include <math.h> //библиотеки для работы с математическими функциями  
using namespace std; //пространство имен
```

```
//setlocale(LC_ALL, "Russian"); //библиотека для работы с русским языком  
class Pr { //начало описания класса  
public: //открытые свойства и методы класса  
    int a, b; //свойства класса  
    int S(); //метод класса  
}; //конец описания класса  
int Pr::S() //метод-функция класса  
{ return a*b; //возвращаемое значение функции  
} //конец описания метода  
int main() //главная функция  
{ //начало главной функции  
    Pr Rabadan; //описание экземпляра класса
```

```

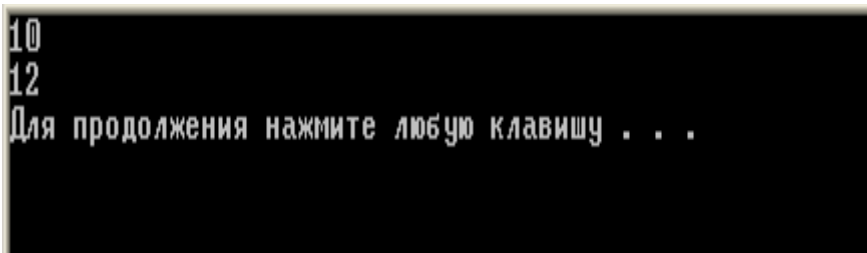
Pr Tanya;//описание экземпляра класса
Rabadan.a=5;//описание экземпляра класса
Rabadan.b=2;//описание экземпляра класса
cout<<Rabadan.S()<<endl;//вывод результата
Tanya.a=4;//описание экземпляра класса
Tanya.b=3;//
cout<<Tanya.S()<<endl;//вывод результата

system("pause");//задержка экрана
return 0;//возврат из главной процедуры
}конец главной процедуры

```

6. Построение проекта: после ввода программного кода нужно скомпилировать и отладить программу. Для этого: Построение – Компилировать (Ctrl + F7) – Программа скомпилирована. Если в программном коде имеются ошибки, они отобразятся в нижней части окна. Чтобы увидеть, в каком месте ошибка, нужно дважды кликнуть на сообщении об ошибке. Синий маркер укажет на строку в программном коде, где имеется ошибка. Далее создаем исполняемый exe-файл: Построение – Построить решение (F7). Чтобы просмотреть результат выполнения программы, нужно выполнить: Отладка – Запуск без отладки (Ctrl+F5).

Результат выполнения программы:



```

10
12
Для продолжения нажмите любую клавишу . . .

```

Пример № 13.2.

Разработайте класс «Клиент», который содержит свойства: ФИО клиента, домашний адрес покупателя, размер скидки. Выполните тестирование класса.

Код программы.

```

#include "stdafx.h"
#include <iostream>
#include <fstream>
#include <string>

```

```

#include <time.h>
#include <iomanip>

using namespace std;

char* getFIO(char* data) {
    if (!strcmp("name", data)) {
        static char FIO[200];
        FIO[0] = NULL;
        char Name[12][15] = { "Эдуард", "Игорь", "Дмитрий", "Антон", "Алексей",
            "Георгий", "Мария", "Анастасия", "Ольга", "Марина", "Дарья", "Виктория" };
        char secondName[12][15] = { " Эдуардович ", " Андреевич ", " Михайлович ",
            " Анатольевич ", " Павлович ", " Геннадьевич ", " Эдуардовна ", " Андреевна ", " Ми-
            хайловна ", " Анатольевна ", " Павловна ", " Геннадьевна " };
        char surname[12][15] = { "Павлов", "Бунин", "Пушкин", "Харитонов", "Тол-
            стой", "Суворов", "Павлова", "Булнина", "Пушкина", "Харитоновна", "Толстая",
            "Суворова" };
        if (rand() % 2 == 0) {
            strcat_s(FIO, 200, Name[rand() % 6]);
            strcat_s(FIO, 200, secondName[rand() % 6]);
            strcat_s(FIO, 200, surname[rand() % 6]);
        }
        else {
            strcat_s(FIO, 200, Name[6 + rand() % 6]);
            strcat_s(FIO, 200, secondName[6 + rand() % 6]);
            strcat_s(FIO, 200, surname[6 + rand() % 6]);
        }
        //cout << "INFUN FIO: " << FIO << endl;
        return FIO;
    }
    if (!strcmp("address", data)) {
        static char Address[40];
        Address[0] = NULL;
        char buffer[10];
        _itoa_s(1 + rand() % 200, buffer, 10, 10);
        char arr[12][30] = { "ул. Береговая, ", "пл. Гвардейская, ", "пер. Новый, ", "ул.
Накимова, ", "ул. Горького, ", "пл. Металлургов, ",
            "Ворошиловский пр-кт, ", "ул. Кар-
ла Маркса, ", "пер. Газетный, ", "пр-кт Шолохова, ", "ул. Коммунистов, ", "пр-кт
Ленина, " };
        strcat_s(Address, 40, arr[rand() % 12]);
        strcat_s(Address, 40, buffer);
        //cout << "INFUN ADRS: " << Address << endl;
        return Address;
    }
}

class client {
private:

```

```

char FIO[200] = "";
char address[40] = "";
int discount;
public:
    client() {
        strcat_s(FIO, 200, "NoName");
        strcat_s(address, 40, "NoAddress");
        discount = 0;
    }
    client(char* name, char* adr, int sale) {
        strcat_s(FIO, 200, name);
        strcat_s(address, 40, adr);
        discount = sale;
    }
    void printData(std::ostream &var) {
        var.fill(' ');
        var << "ФИО: " << FIO;
        if (strlen(FIO) < 26)
            var << ",\tадрес: " << address;
        else
            var << ",\tадрес: " << address;
        var << ", " << setw(34 - strlen(address)) << "скидка: " << discount << endl;
    }

    void enter_data(std::istream &var) {
        enter_FIO(var);
        enter_address(var);
        enter_discount(var); var.ignore();
    }

    void enter_FIO(std::istream &var) {
        cout << "Введите ФИО: ";
        var.getline(FIO, 200);
    }

    void enter_address(std::istream &var) {
        cout << "Введите адрес: ";
        var.getline(address, 40);
    }

    void enter_discount(std::istream &var) {
        cout << "Введите скидку: ";
        var >> discount;
    }

    void set_data(char* name, char* adr, int sale) {
        set_FIO(name);
        set_address(adr);
    }

```

```

        set_discount(sale);
    }

    void set_FIO(char* name) {
        FIO[0] = NULL;
        strcat_s(FIO, 200, name);
    }
    void set_address(char* adr) {
        address[0] = NULL;
        strcat_s(address, 40, adr);
    }

    void set_discount(int sale) {
        discount = sale;
    }

    char* get_FIO() {
        return FIO;
    }

    char* get_address() {
        return address;
    }

    int get_discount() {
        return discount;
    }
};

void main() {
    setlocale(0, "Russian");
    srand(time(NULL));
    char choice; int count;
    cout << "Введите кол-во клиентов: ";
    cin >> count;
    do {
        cout << "Ввести данные самостоятельно?(y||n): ";
        cin >> choice; cin.ignore();
    } while (choice != 'y' && choice != 'n');
    client *clnt = new client[count];
    if (choice == 'y') {
        for (int i = 0; i < count; i++) {
            cout << "Введите данные " << i + 1 << " клиента:\n";
            clnt[i].enter_data(cin);
            cout << "-----" << endl;
        }
    }
    else {
        cout << "\nСгенерирован список клиентов:\n";
    }
}

```

```

        for (int i = 0; i < count; i++)
            clnt[i].set_data(getFIO("name"), getFIO("address"), 1 + rand() % 70);
    }
    for (int i = 0; i < count; i++)
        clnt[i].printData(cout);
    delete[] clnt;
    system("pause");
}

```

Результат выполнения программы:

```

Введите кол-во клиентов: 5
Ввести данные самостоятельно?(y\n): n

Сгенерирован список клиентов:
ФИО: Георгий Павлович Толстой,      адрес: ул. Коммунистов, 121,      скидка: 23
ФИО: Ольга Андреевна Павлова,      адрес: пл. Гвардейская, 17,      скидка: 58
ФИО: Игорь Андреевич Харитонов,     адрес: пр-кт Ленина, 63,        скидка: 45
ФИО: Игорь Анатольевич Пушкин,     адрес: пер. Газетный, 6,        скидка: 29
ФИО: Антон Андреевич Суворов,       адрес: ул. Береговая, 12,       скидка: 42
Для продолжения нажмите любую клавишу . . .

```

Задания для самостоятельного выполнения:

Задание 13.1. Разработка класса.

1. Разработайте класс «Студент», который содержит свойства: ФИО студента, дата поступления, номер зачетной книжки, семестр, балл по каждому из пяти предметов. Разработанный класс должен содержать конструктор, а также метод вычисления среднего балла студента за семестр и вывода исходных данных и результата. Напишите программу, которая создает и инициализирует два объекта разработанного класса, вычисляет и выводит результат.

2. Разработайте класс «ЖКХ», который содержит свойства: лицевой счёт, оплата за газ (руб.), оплата за мусор (руб.), оплата за воду (руб.). Разработанный класс должен содержать конструктор, а также метод вычисления общей стоимости оплат и вывода исходных данных и результата. Напишите програм-

му, которая создает и инициализирует два объекта разработанного класса, вычисляет и выводит результат.

3. Разработайте класс «Коммунальные оплаты», который содержит свойства: лицевой счет абонента, год, месяц, предыдущее показание счетчика (куб. м.), текущее показание счетчика (куб. м.), тариф оплаты за воду. Разработанный класс должен содержать конструктор, а также метод вычисления стоимости оплаты за месяц и вывода исходных данных и результата. Напишите программу, которая создает и инициализирует два объекта разработанного класса, вычисляет и выводит результат.

4. Разработайте класс «Кинотеатр», который содержит свойства: дата, время сеанса, количество проданных билетов на сеанс, цена билета. Разработанный класс должен содержать конструктор, а также метод вычисления стоимости проданных билетов на каждый сеанс и вывода исходных данных и результата. Напишите программу, которая создает и инициализирует два объекта разработанного класса, вычисляет и выводит результат.

5. Разработайте класс «Заказ», который содержит свойства: дата, виды заказов, количество клиентов, цена единицы заказа. Разработанный класс должен содержать конструктор, а также метод вычисления стоимости каждого вида заказа на дату и вывода исходных данных и результата. Напишите программу, которая создает и инициализирует два объекта разработанного класса, вычисляет и выводит результат.

6. Разработайте класс «Услуги», который содержит свойства: дата, код услуги, наименование услуги, цена (руб.) услуги, количество услуг на дату. Разработанный класс должен содержать конструктор, а также метод вычисления стоимости каждого произведенного вида услуг и вывода исходных данных и результата. Напишите программу, которая создает и инициализирует два объекта разработанного класса, вычисляет и выводит результат.

7. Разработайте класс «Потребность деталей», который содержит свойства: номенклатурный номер детали, количество деталей на комплект, количество комплектов. Разработанный класс должен содержать конструктор, а также метод вычисления общей потребности в деталях и вывода исходных данных и результата. Напишите программу, которая создает и инициализирует два объекта разработанного класса, вычисляет и выводит результат.

8. Разработайте класс «Сборка», который содержит свойства: Код узла для сборки садового домика, количество узлов, цена (руб.) одного узла. Разработанный класс должен содержать метод `set_Init()` инициализации указанных переменных, а также метод вычисления стоимости каждого вида узла и вывода исходных данных и результата. Напишите программу, которая создает и инициализирует два объекта разработанного класса, вычисляет и выводит результат. Для создания и выполнения второго объекта используйте указатель.

9. Разработайте класс «Поступление продукции», который содержит свойства: код продукта; количество (кг.), цена реализации (руб). Разработанный класс должен содержать конструктор, а также метод вычисления стоимости каждого вида продукта и вывода исходных данных и результата. Напишите программу, которая создает и инициализирует два объекта разработанного класса, вычисляет и выводит результат.

10. Разработайте класс «Зарплата», который содержит свойства: год, месяц, табельный номер сотрудника, начислено (руб.), удержано (руб.). Разработанный класс должен содержать конструктор, а также метод вычисления суммы к выдаче и вывода исходных данных и результата. Напишите программу, которая создает и инициализирует два объекта разработанного класса, вычисляет и выводит результат.

11. Разработайте класс «Выращивание культур», который содержит свойства: наименования культуры, план(ц), факт (ц). Разработанный класс должен содержать конструктор, а также метод вычисления процента (%) выполнения

плана посадки культур и вывода исходных данных и результата. Напишите программу, которая создает и инициализирует два объекта разработанного класса, вычисляет и выводит результат.

12. Разработайте класс «Склад», который содержит свойства: дата, наименование товара (кг), получено товара (кг), выделено товара (кг). Разработанный класс должен содержать конструктор, а также метод вычисления остатка товара на дату и вывода исходных данных и результата. Напишите программу, которая создает и инициализирует два объекта разработанного класса, вычисляет и выводит результат.

13. Разработайте класс «Фонд зарплаты», который содержит свойства: название фирмы, адрес фирмы, число должностей, средняя годовая заработная плата (руб.). Разработанный класс должен содержать конструктор, а также метод вычисления фонда заработной платы и вывода исходных данных и результата. Напишите программу, которая создает и инициализирует два объекта разработанного класса, вычисляет и выводит результат.

14. Разработайте класс «Сдельная оплата», который содержит свойства: табельный номер, количество выполненных деталей (шт), расценка за единицу изделия (руб.). Разработанный класс должен содержать конструктор, а также метод вычисления сдельной заработной платы и вывода исходных данных и результата. Напишите программу, которая создает и инициализирует два объекта разработанного класса, вычисляет и выводит результат.

15. Разработайте класс «Отпуск продукции», который содержит свойства: номенклатурный номер, отпущено в цех № 1, отпущено в цех № 2. Разработанный класс должен содержать конструктор, а также метод вычисления итоговых данных отпущенной продукции по всем цехам и вывода исходных данных и результата. Напишите программу, которая создает и инициализирует два объекта разработанного класса, вычисляет и выводит результат.

16. Разработайте класс «Оплата труда», который содержит свойства: вид оплаты, служащие (руб.), ВОХР (руб.). Разработанный класс должен содержать конструктор, а также метод вычисления итоговых данных по виду оплат и вывода исходных данных и результата. Напишите программу, которая создает и инициализирует два объекта разработанного класса, вычисляет и выводит результат.

17. Разработайте класс «Оборот», который содержит свойства: номенклатурный номер, остаток материалов на дату (руб.), оборот материалов (руб.) за день. Разработанный класс должен содержать конструктор, а также метод вычисления остатка материала на конец дня и вывода исходных данных и результата. Напишите программу, которая создает и инициализирует два объекта разработанного класса, вычисляет и выводит результат.

18. Разработайте класс «АТС», который содержит свойства: год, месяц, общее время разговоров, тариф. Разработанный класс должен содержать конструктор, а также метод вычисления стоимости разговоров и вывода исходных данных и результата. Напишите программу, которая создает и инициализирует два объекта разработанного класса, вычисляет и выводит результат.

19. Разработайте класс «Абитуриент», который содержит свойства: ФИО абитуриента, адрес абитуриента, баллы по каждому из трех предметов. Разработанный класс должен содержать конструктор, а также метод вычисления среднего балла и вывода исходных данных и результата. Напишите программу, которая создает и инициализирует два объекта разработанного класса, вычисляет и выводит результат.

20. Разработайте класс «Товар», который содержит свойства: наименование товара, фирма - изготовитель, количество (шт), цена. Разработанный класс должен содержать конструктор, а также метод вычисления стоимости товара и вывода исходных данных и результата. Напишите программу, которая создает и инициализирует два объекта разработанного класса, вычисляет и выводит результат.

К лабораторной работе №14. Классы. Наследование

Пример 14.1. Разработать базовый и производный класс для расчета площадей квадрата и прямоугольника

Порядок выполнения:

1. Пуск, программы, Visual Studio 2015, Создать проект, Visual C++, Консольное приложение Win32, имя (задать имя проекта), расположение (диск, каталог), имя решения (появится автоматически), ОК, консольное приложение, пустой проект, готово.
2. Проект (пункт меню), добавить новый элемент, файл C++ (.cpp), имя (ввести имя программы), добавить.
3. В папке Файлы исходного кода появился файл с введенным именем (имя.cpp). Набрать текст в редакторе. Выполнить: Проект, Построить решение.
4. Отладка программы. Отладка (пункт меню), начать отладку, выполнить построение, ДА. Повторное открытие проекта осуществить с помощью функции Открыть проект.
5. Ввод программного кода:

```
#include<iostream>
using namespace std;
class Kvadrat
{
protected:
    int a;
public:
    int x;
    int f1(int x);
};
int Kvadrat::f1(int x)
    {return x*x;}

class Prymougolnik: Kvadrat
{
public:
    int y;
    int f1(int y)
```

```

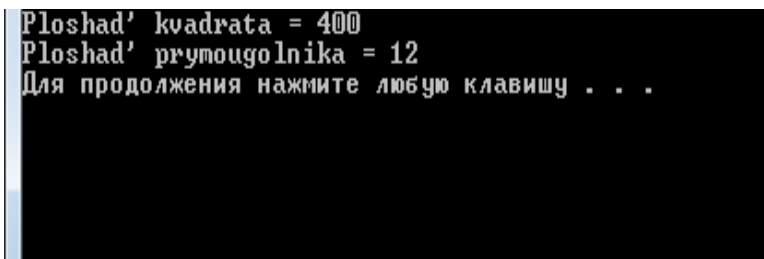
        {Kvadrat::a=4;
        return Kvadrat::a*y;}
};

int main()
{
    Kvadrat kv;
    Prymougolnik pr;
    kv.x=20;
    pr.y=5;
    int x1 = kv.f1(kv.x);
    int x2 = pr.f1(3);
    cout<<"Ploshad' kvadrata = "<<x1<<endl;
    cout<<"Ploshad' prymougolnika = "<<x2<<endl;
    system("pause");
    return 0;
}

```

6. Построение проекта: после ввода программного кода нужно скомпилировать и отладить программу. Для этого: Построение – Компилировать (Ctrl + F7) – Программа скомпилирована. Если в программном коде имеются ошибки, они отобразятся в нижней части окна. Чтобы увидеть, в каком месте ошибка, нужно дважды кликнуть на сообщении об ошибке. Синий маркер укажет на строку в программном коде, где имеется ошибка. Далее создаем исполняемый exe-файл: Построение – Построить решение (F7). Чтобы просмотреть результат выполнения программы, нужно выполнить: Отладка – Запуск без отладки (Ctrl+F5).

Результат выполнения программы:



```

Ploshad' kvadrata = 400
Ploshad' prymougolnika = 12
Для продолжения нажмите любую клавишу . . .

```

Пример 14.2. Создание базового класса зарплата и производного класса премия

```

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

```

```

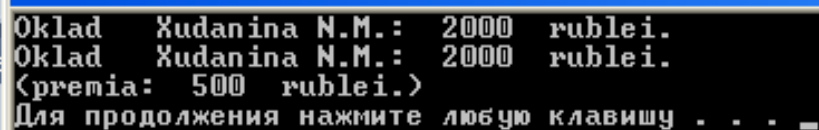
using namespace std;

class zp // определение базового класса

private: // закрытые свойства класса
    char name[64]; // фамилия сотрудника
    float sal; // оклад сотрудника
public: // открытые методы класса
    zp(char *,float); // конструктор
    void show_sal(void); // метод вывода данных
}; //
zp::zp(char *new_name, float old_sal) // описание конструктора
{
    strcpy(name,new_name);
    sal=old_sal;
}
void zp::show_sal(void) // описание метода вывода данных
{
    cout<<"Oklad " <<name<<": " <<sal<<" rublei. "<<endl; //
}
class zpp:public zp // определение производного класса
{
private: // закрытые свойства класса
    float prize; //
public: // открытые методы класса
    zpp(char *,float,float); // конструктор
    void show_salp(void); // метод вывода данных
};
zpp::zpp(char *new_name,float old_sal, float sum_prize):zp(new_name, old_sal) //
описание конструктора
{
    prize=sum_prize; //
}
void zpp::show_salp(void) // описание метода вывода данных
{
    show_sal(); // вызов метода базового класса
    cout<<"(premia: " <<prize<<" rublei. ) "<<endl; // вывод премии
}
int main()
{
    zp sotr(" Xudanina N.M.",2000.0); // объявление экземпляра класса сотрудник sotr
    zpp sotr_prize(" Xudanina N.M.",2000.0,500.0); // объявление экземпляра класса со-
трудник sotr_prize
    sotr.show_sal(); // вызов метода базового класса
    sotr_prize.show_salp(); // вызов метода производного класса
    system ("pause");
    return 0;
}

```

Результат выполнения программы:



```
Oklad Xudanina N.M.: 2000 rublei.  
Oklad Xudanina N.M.: 2000 rublei.  
(premia: 500 rublei.)  
Для продолжения нажмите любую клавишу . . . _
```

Задания для самостоятельного выполнения:

Задание 14.1. Разработка базового и производного классов.

1. Создайте программу с классом Студент, который включает в себя следующие данные – элементы о студентах университета:

- ФИО;
- курс;
- номер группы;
- семестр;
- размер стипендии;
- оценки по N предметам за семестр.

В состав класса входят следующие функции-члены класса:

- конструктор с параметрами;
- функция, которая вычисляет средний балл за семестр;
- виртуальная функция просмотра текущего объекта.

Производный класс Студент_1 содержит следующие элементы:

- общественная работа;
- процент надбавки к стипендии, если все оценки «отлично»;
- переопределенную функцию просмотра состояния объекта.

Введите пользовательские данные для работы с классом

2. Создайте программу с классом Фирма, который включает в себя следующие данные – элементы о сотрудниках фирмы:

- ФИО сотрудника;

- табельный номер;
- количество отработанных часов за месяц;
- почасовой тариф.

В состав класса входят следующие функции-члены класса:

- конструктор с параметрами;
- функция, которая позволяет вывести размер заработной платы каждого сотрудника фирмы, за вычетом подоходного налога, который составляет 13% от суммы заработка. Необходимо учесть, что рабочее время свыше 144 часов считается сверхурочным и оплачивается в двойном размере;

- виртуальная функция просмотра текущего объекта.

Производный класс Фирма _1 содержит следующие элементы:

- процент надбавки к зарплате;
- переопределенную функцию просмотра состояния объекта.

Введите пользовательские данные для работы с классом

3. Создайте программу с классом Абитуриент, который включает в себя следующие данные – элементы об абитуриентах, сдавших вступительные экзамены в университет:

- ФИО;
- адрес;
- оценки по вступительным предметам.

В состав класса входят следующие функции-члены класса:

- конструктор с параметрами;
- функция, позволяющая определить средний балл абитуриента;
- виртуальная функция просмотра текущего объекта.

Производный класс Абитуриент _1 содержит следующие данные о льготах:

- тип медали (золотая или серебряная);
- призовое место на Всероссийской олимпиаде;

- переопределенную функцию просмотра состояния объекта.

Введите пользовательские данные для работы с классом

4. Создайте программу с классом Студент, который включает в себя следующие данные о студентах, желающих получить места в общежитии. Общежитие в первую очередь предоставляется тем студентам, у кого доход на члена семьи меньше двух минимальных зарплат. Класс включает в себя следующие данные-элементы:

- ФИО студента;
- номер группы;
- средний балл;
- доход на одного члена семьи;
- количество членов семьи;

В состав класса входят следующие функции-члены класса:

- конструктор с параметрами;
- деструктор;
- функция, которая определяет возможность предоставления студенту места в общежитии;
- виртуальная функция просмотра текущего объекта.

Производный класс Студент_1 содержит следующие элементы:

- вид общественной нагрузки;
- переопределенную функцию просмотра состояния объекта.

Введите пользовательские данные для работы с классом

5. Создайте программу с классом Автобус, который включает в себя следующие данные – элементы об автобусных рейсах:

- номер рейса;
- тип автобуса;
- пункт назначения;

- цена билета;
- льготы при покупке билета;
- время отправления.

В состав класса входят следующие функции-члены класса:

- конструктор с параметрами;
- функция - расчета стоимости билета с учетом льготы;
- виртуальная функция просмотра текущего объекта.

Производный класс Автобус_1 содержит следующие данные об автобусах-экспрессах:

- время прибытия на конечный пункт.
- переопределенную функцию просмотра состояния объекта.

Введите пользовательские данные для работы с классом

6. Создайте программу с классом Спорт, который включает в себя следующие данные – элементы об участниках спортивных соревнований:

- ФИО игрока;
- игровой номер;
- возраст;
- рост;
- вес.

В состав класса входят следующие функции-члены класса:

- конструктор с параметрами;
- функция, которая выведет информацию о допустимых параметрах (вес, возраст) в команде;
- виртуальная функция просмотра текущего объекта.

Производный класс Спорт_1 содержит следующие данные:

- разряд спортсмена;
- призер (Европы, мира и т.д.).
- переопределенную функцию просмотра состояния объекта.

Введите пользовательские данные для работы с классом

7. Создайте программу с классом Телефон, который включает в себя следующие данные – элементы о разговорах на международной АТС:

- дату разговора;
- код и название города;
- продолжительность разговора;
- тариф;
- номер телефона в этом городе;
- номер телефона абонента.

В состав класса входят следующие функции-члены класса:

- конструктор с параметрами;
- функция, которая позволяет рассчитать стоимость разговора;
- виртуальная функция просмотра текущего объекта.

Производный класс Телефон _1 содержит следующие элементы:

- вид льгот;
- процент для льготного тарифа;
- переопределенную функцию просмотра состояния объекта.

Список использованных и рекомендованных источников

1. Информатика. Базовый курс: учеб. пос. / ред. С. В. Симонович. – 2-е изд. – СПб.: Питер, 2007. – 640 с.
2. Веретенникова Е.Г., Савельева Н.Г. Информатика и программирование: учеб. пос. – Ростов н/Д.: Изд-во Рост. ун-та, 2000.
3. Веретенникова Е.Г., Патрушина С.М., Савельева Н.Г. Тесты по информатике (500 вопросов). – Ростов н/Д.: Издат. центр «МарТ», 2001.
4. Веретенникова Е.Г., Патрушина С.М., Н.Г. Савельева. Информатика: учеб. пос. *Серия Учебный курс*. – Ростов н/Д.: Издательский центр «МарТ», 2002. – 406 с.
5. Веретенникова Е.Г., Патрушина С.М., Савельева Н.Г. Компьютер для пользователя: учеб. пос.. 2-е изд., перераб. и д ИКЦ «МарТ». – Ростов н/Д.: Издат. Центр «МарТ», 2003. – 304 с.
6. Веретенникова Е.Г., Нельзина О.Г., Патрушина С.М., Савельева Н.Г. Информатика: учебный курс. / под ред. С.М. Патрушиной. – М.: ИКЦ «МарТ», 2004.
7. Информатика: учеб. пос. / под ред. д-ра эконом. наук, проф. Г.Н. Хубаева: изд. 3-е, перераб. и доп. – Ростов-н/Д.: Издат. центр «МарТ»; Феникс, 2010. – 288 с.
8. Информатика. Информационные системы. Информационные технологии. Тестирование. Подготовка к Интернет-экзамену / под общ. ред. Г.Н. Хубаева. изд. 3-е, перераб. и доп. – Ростов-н/Д.: Издат. центр «МарТ»; Феникс, 2011. – 268 с.
9. Патрушина С.М., Нельзина О.Г., Аручиди Н.А., Савельева Н.Г. Информатика: учеб. пос.: изд. 2-е, перераб. и доп. – М.: ИКЦ «МарТ», Ростов н/Д.: Издат. центр «МарТ», 2004. – 400с.

10. Пахомов Б.И. С./C++ 2010 для начинающих. – СПб.: БХВ-Петербург, 2011. – 736 с.
11. Савельева Н.Г., Мирошниченко И.И., Гречкина В.Ю. Visual Basic: Практикум / РГЭУ «РИНХ». – Ростов н/Д., 2006. – 66 с.
12. Савельева Н.Г. Основы алгоритмизации и программирования: учеб. пос.: 2-е изд. доп. и перераб. – Ростов н/Д.: Изд-во РГЭУ (РИНХ), 2000 – 150 с.
13. Макарова Н.В., Волков В.Б. Информатика: учеб. для вузов. – СПб.: Питер, 2011, – 576 с.
14. Хортон, Айвор. Visual C++ 2010: полный курс: пер. с англ. – М.: ООО «И.Д. Вильямс», 2011. – 1216 с.
15. Патрушина С.М., Нельзина О.Г., Аручиди Н.А., Савельева Н.Г. Компьютер в работе бухгалтера: учеб. пос. – М.: ИКЦ «МарТ», 2003.

Приложение
(информационное)

Форма титульного листа для отчета по самостоятельным работам

ФГБОУ ВО «РОСТОВСКИЙ ГОСУДАРСТВЕННЫЙ
ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ (РИНХ)»
ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И
ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Кафедра Информационных систем и
прикладной информатики

Отчет к самостоятельной работе № ____

на тему:

По дисциплине: *название дисциплины*

Курс: *номер курса*

Группа: *номер группы*

Направление (специальность): *название направления (специальности)*

Вариант: *номер варианта задания*

Выполнил:
студент *инициалы и фамилия студента*

Проверил:
должность *ФИО*

Ростов-на-Дону

20__

Учебное издание

Мирошниченко Ирина Иосифовна,
Веретенникова Елена Григорьевна,
Савельева Наталья Григорьевна,
Гречкина Вера Юрьевна

**Методические рекомендации
для выполнения
самостоятельной работы
по дисциплинам
«Информатика и программирование»,
«Информатика»**

*Для студентов 1, 2 курса
очной, заочной, очно-заочной формы обучения*

**Направления 09.03.02 Информационные системы и технологии,
09.03.03 Прикладная информатика,
10.03.01 «Информационная безопасность»**

Корректора, верстка Н.В. Фоменко

Изд. № 100/3436. Подписано к использованию 03.07.19.

344002, Ростов-на-Дону, ул. Б. Садовая, 69, РГЭУ (РИНХ), а.152
Издательско-полиграфический комплекс РГЭУ (РИНХ)