


**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ,
СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**
**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)**


Санкт-Петербургский колледж телекоммуникаций им. Э.Т. Кренкеля

УТВЕРЖДАЮ

Заместитель директора

по учебной работе

 О.В. Колбанева

 21.09.2021 г

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ
САМОСТОЯТЕЛЬНЫХ РАБОТ**

по учебной дисциплине

ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

по специальности

10.02.04 Обеспечение информационной безопасности телекоммуникационных систем
(код и наименование специальности)

квалификация

техник по защите информации
среднего профессионального образования

Санкт-Петербург
2021

ОП. 05 Основы алгоритмизации и программирования. Методические указания по выполнению самостоятельных работ.

Составила А.А. Обудовская – Санкт-Петербург, 2021.

Методические указания содержат описания самостоятельных работ, предусмотренных рабочей программой ОП. 05 Основы алгоритмизации и программирования. Количество внеурочных самостоятельных работ 10, общий объём составляет 20 часов. Нумерация рисунков, формул и таблиц в пределах одной работы. Методические указания предназначены для обучающихся очной формы обучения по специальности 10.02.04 Обеспечение информационной безопасности телекоммуникационных систем.

Рассмотрено и одобрено предметной (цикловой) комиссией информатики и программирования в компьютерных системах Санкт-Петербургского колледжа телекоммуникаций им. Э.Т. Кренкеля.

СОДЕРЖАНИЕ

№		
1.	Пояснительная записка	4
2.	Перечень самостоятельных работ	4
3.	Самостоятельная работа №1. Составление алгоритмов разветвляющейся структуры.	5
4.	Самостоятельная работа № 2. Линейные вычислительные программы	7
5.	Самостоятельная работа № 3. Программы для решения задач с использованием условного оператора	10
6.	Самостоятельная работа № 4. Применение операторов цикла в С++ для вычисления определённых интегралов методами прямоугольников, трапеций, Симпсона	12
7.	Самостоятельная работа № 5. Работа с элементами одномерных массивов	14
8.	Самостоятельная работа № 6. Обработка двумерных массивов.	16
9.	Самостоятельная работа № 7. Функции-подпрограммы в языке С++.	19
10.	Самостоятельная работа № 8. Работа со структурами в языке С++	22
11.	Самостоятельная работа № 9. Простейшие программы с использованием визуальных компонентов	25
12.	Самостоятельная работа № 10. Обработка одномерных массивов в визуальной среде на языке Visual C++	27
13.	Список литературы и других источников	31

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Самостоятельные работы разработаны в рамках рабочей программы учебной дисциплины ОП.05. Основы алгоритмизации и программирования, являющейся частью основной профессиональной образовательной программы в соответствии с ФГОС СПО по специальности 10.02.04 Обеспечение информационной безопасности телекоммуникационных систем.

Учебная дисциплина ОП.05. Основы алгоритмизации и программирования обеспечивает формирование общих компетенций по всем видам деятельности ФГОС по специальности 10.02.04 Обеспечение информационной безопасности телекоммуникационных систем.

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности;

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие;

ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами;

ОК 09. Использовать информационные технологии в профессиональной деятельности;

2. ПЕРЕЧЕНЬ САМОСТОЯТЕЛЬНЫХ РАБОТ

№	Наименование	Часы
1	Составление алгоритмов разветвляющейся структуры.	2ч.
2	Линейные вычислительные программы	2ч.
3	Программы для решения задач с использованием условного оператора	2ч.
4	Применение операторов цикла в С++ для вычисления определённых интегралов методами прямоугольников, трапеций, Симпсона	2ч.
5	Работа с элементами одномерных массивов	2ч.
6	Обработка двумерных массивов.	2ч.
7	Функции-подпрограммы в языке С++.	2ч.
8	Работа со структурами в языке С++	2ч.
9	Простейшие программы с использованием визуальных компонентов	2ч.
10	Обработка одномерных массивов в визуальной среде на языке Visual C++	2ч.

Самостоятельная работа № 1

ТЕМА. Составление алгоритмов разветвляющейся структуры.

1. Цели работы:

- развитие коммуникативных компетенций;
- развитие общих компетенций;
- формирование умений работы с информацией;
- формирование понимания темы;
- формирование умения анализировать;
- закрепление знаний основных моделей и методов построения алгоритмов;
- освоение практических навыков разработки алгоритма вычислительного процесса разветвляющейся структуры.

2. Задача:

Составить в рабочей тетради блок-схему алгоритма в соответствии с индивидуальным заданием.

3. Подготовка к работе и порядок выполнения:

- изучить конспект лекций и методические указания для практических работ;
- начертить в рабочей тетради блок-схему.

Индивидуальные задания:

1. Даны три действительных числа. Возвести в квадрат те из них, значения которых неотрицательны, и в четвертую степень — отрицательные.

2. Даны две точки $A(x_1, y_1)$ и $B(x_2, y_2)$. Составить алгоритм, определяющий, которая из точек находится ближе к началу координат.

3. Даны два угла треугольника (в градусах). Определить, существует ли такой треугольник, и если да, то будет ли он прямоугольным.

4. Даны действительные числа x и y , не равные друг другу. Меньшее из этих двух чисел заменить половиной их суммы, а большее — их удвоенным произведением.

5. На плоскости XOY задана своими координатами точка A . Указать, где она расположена (на какой оси или в каком координатном угле).

6. Даны целые числа m, n . Если числа не равны, то заменить каждое из них одним и тем же числом, равным большему из исходных, а если равны, то заменить числа нулями.

7. Подсчитать количество отрицательных среди чисел a, b, c .

8. Подсчитать количество положительных среди чисел a, b, c .

9. Подсчитать количество целых среди чисел a, b, c .

10. Определить, делителем каких чисел a, b, c является число k .

11. Перераспределить значения переменных x и y так, чтобы в x оказалось большее из этих значений, а в y — меньшее.

12. Определить правильность даты, введенной с клавиатуры (число — от 1 до 31, месяц — от 1 до 12). Если введены некорректные данные, то сообщить об этом.

13. Написать программу — модель анализа пожарного датчика в помещении, которая выводит сообщение «Пожароопасная ситуация», если температура в комнате превысила 60°C .

14. Рис расфасован в два пакета. Масса первого — m кг, второго — n кг. Составить программу, определяющую:

- а) какой пакет тяжелее — первый или второй;
- б) массу более тяжелого пакета.

4. Критерии оценки

5 – блок-схема выполнена аккуратно, в соответствии со стандартами и верно отражает алгоритм выполнения задачи.

4 – при составлении блок-схемы допущены неточности. Например, пропущено присвоение начальных значений переменным или при использовании ветвления отсутствуют указатели направления.

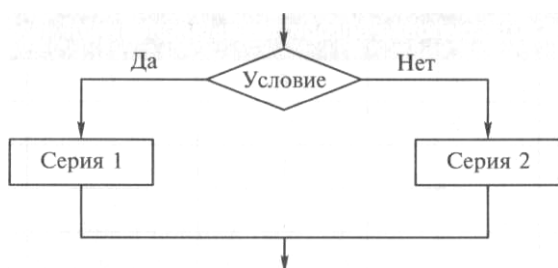
3 – блок-схема не в полной мере отражает алгоритм решения задачи.

2 – блок-схема отсутствует или не соответствует заданию.

5. Пояснения к работе

Базовая структура ВЕТВЛЕНИЕ. Обеспечивает в зависимости от результата проверки условия (**да** или **нет**) выбор одного из альтернативных путей работы алгоритма. Каждый из путей ведет к **общему выходу**, так что работа алгоритма будет продолжаться независимо от того, какой путь будет выбран.

Общий вид команды ветвления в блок-схемах и на алгоритмическом языке следующий:



если условие

то серия 1

иначе серия 2

всё

Сначала проверяется условие (вычисляется логическое выражение). Если условие истинно, то выполняется последовательность команд, на которую указывает стрелка с надписью «Да» (положительная ветвь) — «Серия 1». В противном случае выполняется отрицательная ветвь команд — «Серия 2».

На АЯ условие записывается после служебного слова «если», положительная ветвь — после слова «то», а отрицательная — после слова «иначе». Буквами «всё» в такой записи обозначают конец ветвления.

В записи логических выражений, помимо арифметических операций, используются операции отношения: < (меньше), > (больше), <= (меньше или равно), >= (больше или равно),

= (равно), <> (не равно), а также логические операции – **И, ИЛИ, НЕ**.

Самостоятельная работа № 2

Тема. Линейные вычислительные программы

1. Цели работы:

- развитие коммуникативных компетенций;
- развитие общих компетенций;
- формирование умений работы с информацией;
- формирование понимания темы;
- формирование умения анализировать;
- закрепление знаний основных моделей и методов построения алгоритмов;
- освоение практических навыков разработки алгоритма и составления линейных программ.

2. Задача:

Составить в рабочей тетради блок-схему алгоритма и написать программу на языке программирования C++ в соответствии с индивидуальным заданием.

3. Подготовка к работе и порядок выполнения:

- изучить конспект лекций и методические указания для практических работ;
- начертить в рабочей тетради блок-схему и написать программу на C++ в соответствии с индивидуальным заданием.

Индивидуальные задания:

1. Вычислить периметр и площадь прямоугольного треугольника по длинам a и b двух катетов.
2. Заданы координаты трех вершин треугольника (x_1, y_1) , (x_2, y_2) , (x_3, y_3) . Найти его периметр и площадь.
3. Вычислить длину окружности и площадь круга одного и того же заданного радиуса R .
4. Найти произведение цифр заданного четырехзначного числа.
5. Даны два числа. Найти среднее арифметическое кубов этих чисел и среднее геометрическое модулей этих чисел.
6. Вычислить расстояние между двумя точками с данными координатами (x_1, y_1) и (x_2, y_2) .
7. Даны два действительных числа x и y . Вычислить их сумму, разность, произведение и частное.
8. Дана длина ребра куба. Найти площадь грани, площадь полной поверхности и объем этого куба.
9. Дана сторона равностороннего треугольника. Найти площадь этого треугольника, его высоту, радиусы вписанной и описанной окружностей.
10. Известна длина окружности. Найти площадь круга, ограниченного этой окружностью.
11. Найти площадь кольца, внутренний радиус которого равен r , а внешний — R ($R > r$).
12. Треугольник задан величинами своих углов и радиусом описанной окружности. Найти стороны треугольника.
13. Найти площадь равнобедренной трапеции с основаниями a и b и углом α при большем основании a .
14. Вычислить корни квадратного уравнения $ax^2 + bx + c = 0$ с заданными коэффициентами a , b и c (предполагается, что $a \neq 0$ и что дискриминант уравнения неотрицателен).
15. Дано действительное число x . Не пользуясь никакими другими арифметическими

операциями, кроме умножения, сложения и вычитания, вычислить за минимальное число операций

$$2x^4 - 3x^3 + 4x^2 - 5x + 6.$$

4. Критерии оценки

- 5 – Программа написана верно, блок-схема выполнена аккуратно, правильно и соответствует коду программы.
- 4 – В программе присутствуют синтаксические ошибки, имеются незначительные отклонения в оформлении блок-схемы.
- 3 – Имеются ошибки в логике алгоритма или неверна блок-схема.
- 2 – Блок-схема и код отсутствуют или не соответствуют заданию.

5. Пояснения к работе:

Линейные программы на C/C++

Структура программы. Общая структура программы на C/C++ следующая:

```
директивы_препроцессора
определение_функции_1
определение_функции_2
определение_функции_N
```

Среди функций обязательно присутствует главная функция с именем `main`. Простейшая программа содержит только главную функцию и имеет следующую структуру:

```
директивы_препроцессора
void main()
{ определения_объектов;
  исполняемые_операторы;
}
```

Математические функции (заголовочный файл `math.h`)

Обращение	Тип аргумента	Тип результата	Функция
<code>abs(x)</code>	<code>int</code>	<code>int</code>	абсолютное значение целого числа
<code>acos(x)</code>	<code>double</code>	<code>double</code>	арккосинус (радианы)
<code>asin(x)</code>	<code>double</code>	<code>double</code>	арксинус (радианы)
<code>atan(x)</code>	<code>double</code>	<code>double</code>	арктангенс (радианы)
<code>ceil(x)</code>	<code>double</code>	<code>double</code>	ближайшее целое, не меньшее x
<code>cos(x)</code>	<code>double</code>	<code>double</code>	косинус (x в радианах)
<code>exp(x)</code>	<code>double</code>	<code>double</code>	e^x – экспонента от x

<code>fabs(x)</code>	double	double	абсолютное значение вещественного x
<code>floor(x)</code>	double	double	наибольшее целое, не превышающее x
<code>fmod(x,y)</code>	double /double	double	остаток от деления нацело x / y
<code>log(x)</code>	double	double	логарифм натуральный – $\ln x$
<code>log10(x)</code>	double	double	логарифм десятичный – $\lg x$
<code>pow(x,y)</code>	double double	double	x в степени y – x^y
<code>sin(x)</code>	double	double	синус (x в радианах)
<code>sqrt(x)</code>	double	double	корень квадратный (положительное значение)
<code>tan(x)</code>	double	double	тангенс (x в радианах)

Потоковый ввод-вывод в C++. Программируя на языке C++, можно пользоваться средствами ввода-вывода стандартной библиотеки C, подключаемой с помощью заголовочного файла `stdio.h`. Однако в C++ имеются свои специфические средства ввода-вывода. Это *библиотека классов*, подключаемая к программе с помощью файла `iostream.h`. В этой библиотеке определены в качестве *объектов* стандартные символьные потоки со следующими именами:

`cin` — стандартный поток ввода с клавиатуры;
`cout` — стандартный поток вывода на экран.

Ввод данных интерпретируется как *извлечение из потока* `cin` и присваивание значений соответствующим переменным. В C++ определена операция извлечения из стандартного потока, знак которой `>>`.

Например, ввод значений в переменную x реализуется оператором
`cin>>x;`

Вывод данных интерпретируется как *помещение в стандартный поток* `cout` выводимых значений. Выводиться могут тексты, заключенные в двойные кавычки, и значения выражений. Знак операции помещения в поток `<<`. Примеры использования потокового вывода:

```
cout<<a+b;
cout<<"\nРезультат="<<Y;
cout<<"x="<<x<<" y="<<y<<" z="<<z<<endl;
```

Элемент вывода `endl` является так называемым манипулятором, определяющим перевод курсора на новую строку (действует аналогично управляющему символу `\n`).

В процессе потокового ввода-вывода происходит преобразование из формы внешнего символьного представления во внутренний формат и обратно. Тип данных и необходимый формат определяются автоматически.

Самостоятельная работа № 3

Тема: Программы для решения задач с использованием условного оператора

1. Цели работы:

- формирование умений работы с информацией;
- формирование понимания темы;
- формирование умения анализировать;
- закрепление знаний основных моделей и методов построения алгоритмов;
- освоение практических навыков разработки алгоритма и составления программ с использованием условных операторов.

2. Задача:

3. Составить в рабочей тетради блок-схему алгоритма и написать программу на языке программирования C++ в соответствии с индивидуальным заданием.

4. Подготовка к работе и порядок выполнения:

- Получение у преподавателя варианта задания для предварительной проработки материала дома.
- Предварительная проработка материала (подготовка необходимых формул и изучение рекомендованного теоретического материала).
- Составление блок-схемы и написание программы.

Список индивидуальных заданий.

1. Написать программу нахождения суммы большего и меньшего из трех чисел.
2. Написать программу, по длинам сторон распознающую среди всех треугольников ABC прямоугольные. Если таковых нет, то вычислить величину угла C .
3. Найти $\max\{\min(a, b), \min(c, d)\}$.
4. Даны три числа a, b, c . Определить, какое из них равно d . Если ни одно не равно d , то найти $\max(d - a, d - b, d - c)$.
5. Даны четыре точки $A_1(x_1, y_1), A_2(x_2, y_2), A_3(x_3, y_3), A_4(x_4, y_4)$. Определить, будут ли они вершинами параллелограмма.
6. Даны три точки $A(x_1, y_1), B(x_2, y_2)$ и $C(x_3, y_3)$. Определить, будут ли они расположены на одной прямой. Если нет, то вычислить $\angle ABC$.
7. Даны действительные числа a, b, c . Удвоить эти числа, если $a \geq b \geq c$, и заменить их абсолютными значениями, если это не так.
8. На оси OX расположены три точки a, b, c . Определить, какая из точек b или c расположена ближе к a .
9. Даны три положительных числа a, b, c . Проверить, будут ли они сторонами треугольника. Если да, то вычислить площадь этого треугольника.
10. Написать программу решения уравнения $ax^3 + bx = 0$ для произвольных a, b .
11. Дан круг радиуса R . Определить, поместится ли правильный треугольник со стороной a в этом круге.
12. Дано число x . Напечатать в порядке возрастания числа: $\sin x, \cos x, \ln x$. Если при каком-либо x некоторые из выражений не имеют смысла, вывести сообщение об этом и сравнивать значения только тех, которые имеют смысл.
13. Составить программу, осуществляющую перевод величин из радианной меры в градусную и наоборот. Программа должна запрашивать, какой перевод нужно осуществить, и выполнять указанное действие.
14. Написать программу, которая по заданным трем числам определяет, является ли сумма каких-либо двух из них положительной.
15. Известно, что из четырех чисел a_1, a_2, a_3 и a_4 одно отлично от трех других, равных между собой; присвоить номер этого числа переменной n .

5. Критерии оценки

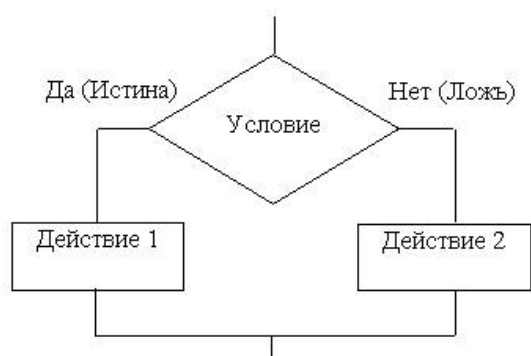
- 5 – Программа написана верно, блок-схема выполнена аккуратно, правильно и соответствует коду программы.
- 4 – В программе присутствуют синтаксические ошибки, имеются незначительные отклонения в оформлении блок-схемы.
- 3 – Имеются ошибки в логике алгоритма или неверна блок-схема.
- 2 – Блок-схема и код отсутствуют или не соответствуют заданию.

6. Пояснения к работе

Условный оператор. Формат условного оператора следующий:

```
if (выражение) оператор1; else оператор2;
```

Это полная форма оператора, программирующая структуру полного ветвления. Обычно *выражение* — это некоторое условие, содержащее операции отношения и логические операции. Значение выражения приводится к целому и интерпретируется в соответствии с правилом: равно нулю — ложь, не равно нулю — истина. Если выражение истинно, выполняется оператор1, если ложно — оператор2. Блок-схема полной формы условного оператора выглядит следующим образом:



Необходимо обратить внимание на следующие особенности синтаксиса условного оператора:

- выражение записывается в круглых скобках;
 - точка с запятой после оператора 1 ставится обязательно.
- Возможно использование неполной формы условного оператора

```
if (выражение) оператор;
```

Блок-схема в этом случае имеет вид:



Вот пример использования полной формы условного оператора для нахождения

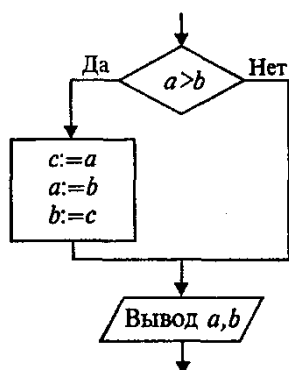
большого значения из двух переменных a и b :

```
if(a>b) max=a; else max=b;
```

Та же самая задача может быть решена с использованием неполного ветвления следующим образом:

```
max=a; if(b>a) max=b;
```

Пример 1. Упорядочить по возрастанию значения в двух переменных a , b :



```
if(a>b)
{c=a; a=b; b=c;}
cout<<"a="<<a<<"b="<<b;
```

В данном примере использован *составной оператор* — последовательность операторов, заключенная в фигурные скобки. Обратите внимание на то, что перед закрывающей фигурной скобкой точку с запятой надо ставить обязательно, а после скобки точка с запятой не ставится.

Самостоятельная работа № 4

Тема: Применение операторов цикла в С++ для вычисления определённых интегралов методами прямоугольников, трапеций, Симпсона

1. Цели работы:

- формирование умений работы с информацией;
- формирование понимания темы;
- формирование умения анализировать;
- закрепление знаний основных моделей и методов построения алгоритмов;
- освоение практических навыков разработки алгоритма и составления программ с использованием операторов цикла.

2. Задача:

Составить блок-схему алгоритма и написать программу приближённого вычисления методами прямоугольников, трапеций и Симпсона при $n = 100$ для следующих интегралов:

$$\int_{-2}^0 (x^2 - 4) \cos 3x dx$$

$$\int_1^{e^2} \frac{\ln^2 x dx}{\sqrt{x}}$$

$$\int_0^1 x^2 e^{3x} dx$$

$$\int_0^1 \frac{(x^2 + 1) dx}{(x^3 + 3x + 1)^2}$$

$$\int_0^{\pi/4} \frac{2 \cos x + 3 \sin x}{(2 \sin x - 3 \cos x)^3} dx$$

$$\int_0^{1/2} \frac{8x - \operatorname{arctg} 2x}{1 + 4x^2} dx$$

$$\int_1^e \frac{x^2 + \ln x^2}{x} dx$$

$$\int_{-1}^0 \frac{\operatorname{tg}(x+1)}{\cos^2(x+1)} dx$$

$$\int_0^1 \frac{x dx}{x^4 + 1}$$

$$\int_0^1 \frac{x dx}{\sqrt{x^4 + x^2 + 1}}$$

3. Подготовка к работе и порядок выполнения:

- Получение у преподавателя варианта задания для предварительной проработки материала дома.
- Предварительная проработка материала (подготовка необходимых формул и изучение рекомендованного теоретического материала).
- Составление блок-схемы и написание программы.

4. Критерии оценки

- 5 – Программа написана верно, блок-схема выполнена аккуратно, правильно и соответствует коду программы.
- 4 – В программе присутствуют синтаксические ошибки, имеются незначительные отклонения в оформлении блок-схемы.
- 3 – Имеются ошибки в логике алгоритма или неверна блок-схема.
- 2 – Блок-схема и код отсутствуют или не соответствуют заданию.

5. Пояснения к работе

В тех случаях, когда аналитическое вычисление интеграла затруднено, интеграл можно вычислить приближенно, воспользовавшись одной из следующих формул:

Формула прямоугольников:
$$\int_a^b f(x) dx \approx \frac{b-a}{n} (y_0 + y_1 + \dots + y_{n-1}).$$

Формула трапеций:
$$\int_a^b f(x) dx \approx \frac{b-a}{n} \left(\frac{y_0 + y_n}{2} + y_1 + y_2 + \dots + y_{n-1} \right).$$

Формула Симпсона:

$$\int_a^b f(x) dx \approx \frac{b-a}{6n} (y_0 + y_{2n} + 2(y_2 + y_4 + \dots + y_{2n-2}) + 4(y_1 + y_3 + \dots + y_{2n-1})).$$

Самостоятельная работа № 5

Тема: Работа с элементами одномерных массивов

1. Цели работы:

- формирование умений работы с информацией;
- формирование понимания темы;
- формирование умения анализировать;
- закрепление знаний основных моделей и методов построения алгоритмов;
- освоение практических навыков разработки алгоритма и составления программ с использованием одномерных массивов.

2. Задача:

Составить в рабочей тетради блок-схему алгоритма и написать программу на языке программирования C++ в соответствии с индивидуальным заданием.

3. Подготовка к работе и порядок выполнения:

- Получение у преподавателя варианта задания для предварительной проработки материала дома.
- Предварительная проработка материала (подготовка необходимых формул и изучение рекомендованного теоретического материала).
- Составление блок-схемы и написание программы.

Список индивидуальных заданий.

1. В массиве $A[N]$ занесены натуральные числа. Найти сумму тех элементов, которые кратны данному K .
2. В целочисленной последовательности есть нулевые элементы. Создать массив из номеров этих элементов.
3. Дана последовательность целых чисел a_1, a_2, \dots, a_n . Выяснить, какое число встречается раньше — положительное или отрицательное.
4. Дана последовательность действительных чисел a_1, a_2, \dots, a_n . Выяснить, будет ли она возрастающей.
5. 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 об этом факте.
6. Дана последовательность чисел a_1, a_2, \dots, a_n . Указать наименьшую длину числовой оси, содержащую все эти числа.
7. Дана последовательность действительных чисел a_1, a_2, \dots, a_n . Заменить все ее члены, большие данного Z , этим числом. Подсчитать количество замен.
8. Последовательность действительных чисел оканчивается нулем. Найти количество членов этой последовательности.
9. Дан массив действительных чисел, размерность которого N . Подсчитать, сколько в нем отрицательных, положительных и нулевых элементов.
10. Даны действительные числа a_1, a_2, \dots, a_n . Поменять местами наибольший и наименьший элементы.

4. Критерии оценки

- 5 — Программа написана верно, блок-схема выполнена аккуратно, правильно и соответствует коду программы.
- 4 — В программе присутствуют синтаксические ошибки, имеются незначительные отклонения в оформлении блок-схемы.
- 3 — Имеются ошибки в логике алгоритма или неверна блок-схема.
- 2 — Блок-схема и код отсутствуют или не соответствуют заданию.

5. Пояснения к работе

Массив — это структура однотипных элементов, занимающих непрерывную область памяти. С массивом связаны следующие его свойства: имя, тип, размерность, размер. Формат описания массива следующий:

тип элементов имя [константное_выражение]

Константное выражение определяет размер массива, т. е. число элементов этого массива. Например, согласно описанию

```
int A[10];
```

объявлен массив с именем А, содержащий 10 элементов целого типа. Элементы массива обозначаются индексированными именами. Нижнее значение индекса равно 0:

A[0], A[1], A[2], A[3], A[4], A[5], A[6], A[7], A[8], A[9]

В отличие от Паскаля в C++ нельзя определять произвольные диапазоны для индексов. Размер массива, указанный в описании, всегда на единицу больше максимального значения индекса.

Размер массива может явно не указываться, если при его объявлении производится инициализация значений элементов. Например:

```
int p[]={2, 4, 6, 10, 1};
```

В этом случае создается массив из пяти элементов со следующими значениями:

p[0]=2, p[1]=4, p[2]=6, p[3]=10, p[4]=1

В результате следующего объявления массива

```
int M[6]={5, 3, 2};
```

будет создан массив из шести элементов. Первые три элемента получают инициализированные значения. Значения остальных будут либо неопределенными, либо равны нулю, если массив внешний или статический.

Рассмотрим несколько примеров программ обработки одномерных массивов.

Пример 1. Ввод с клавиатуры и вывод на экран одномерного массива.

```
//Ввод и вывод массива
#include <iostream>
using namespace std;
void main ()
{ int i, a[5];
  for(i=0; i<5;i++)
  {cout<<"a["<<i<<"=";
    cin>>a[i];
  }
  for(i=0; i<5; i++)
    cout<<"a["<<i<<"]="<<a[i]<<"    ";
}
```

Пример2. Ввод вещественного массива и вычисление среднего значения.

```
#include <iostream>
using namespace std;
void main ()
{ const n=10;
  int i;
  double a[n],sa;
      for(i=0; i<n;i++)
      {   cout<<"a["<<i<<"]="";
          cin>>a[i];
      }
  for(i=0; i<n; i++) sa+=a[i];
      sa/=n;
      cout<<"\Среднее арифметическое массива="<<sa;
}
```

Самостоятельная работа № 6 Тема: Работа с элементами двумерных массивов

1. Цели работы:

- формирование умений работы с информацией;
- формирование понимания темы;
- формирование умения анализировать;
- закрепление знаний основных моделей и методов построения алгоритмов;
- освоение практических навыков разработки алгоритма и составления программ с использованием двумерных массивов.

2. Задача:

Составить в рабочей тетради блок-схему алгоритма и написать программу на языке программирования C++ в соответствии с индивидуальным заданием.

3. Подготовка к работе и порядок выполнения:

- Получение у преподавателя варианта задания для предварительной проработки материала дома.
- Предварительная проработка материала (подготовка необходимых формул и изучение рекомендованного теоретического материала).
- Составление блок-схемы и написание программы.

Список индивидуальных заданий.

1. Вычислить сумму и число положительных элементов матрицы $A[N, N]$, находящихся над главной диагональю.
2. Дана матрица A размером $n \times m$. Определить k — количество особых элементов массива A , считая его элемент особым, если он больше суммы остальных элементов его столбца.
3. Задана квадратная матрица. Поменять местами строку с максимальным элементом на главной диагонали со строкой с заданным номером m .
4. Дана матрица $B[N, M]$. Найти в каждой строке матрицы максимальный и минимальный элементы и поменять их местами с первым и последним элементом строки соответственно.
5. Дана целая квадратная матрица n -го порядка. Определить, является ли она магическим квадратом, т.е. такой, в которой суммы элементов во всех строках и столбцах одинаковы.
6. Элемент матрицы назовем *седловой точкой*, если он является наименьшим в своей

строке и одновременно наибольшим в своем столбце или, наоборот, является наибольшим в своей строке и наименьшим в своем столбце. Для заданной целой матрицы размером $n \times m$ напечатать индексы всех ее седловых точек.

7. Дана матрица размером $n \times m$. Переставляя ее строки и столбцы, добиться того, чтобы наибольший элемент (или один из них) оказался в верхнем левом углу.

8. Определить, является ли заданная целая квадратная матрица n -го порядка симметричной (относительно главной диагонали).

9. Дана целочисленная квадратная матрица. Найти в каждой строке наибольший элемент и поменять его местами с элементом главной диагонали.

10. Упорядочить по возрастанию элементы каждой строки матрицы размером $n \times m$.

11. Задана матрица размером $n \times m$. Найти максимальный по модулю элемент матрицы. Переставить строки и столбцы матрицы таким образом, чтобы максимальный по модулю элемент был расположен на пересечении k -й строки и k -го столбца.

12. Дана квадратная матрица $A[N, N]$. Записать на место отрицательных элементов матрицы нули, а на место положительных — единицы. Вывести на печать нижнюю треугольную матрицу в общепринятом виде.

13. Дана действительная матрица размером $n \times m$, все элементы которой различны. В каждой строке выбирается элемент с наименьшим значением, затем среди этих чисел выбирается наибольшее. Указать индексы элемента с найденным значением.

14. Дана действительная квадратная матрица порядка N (N — нечетное), все элементы которой различны. Найти наибольший элемент среди стоящих на главной и побочной диагоналях и поменять его местами с элементом, стоящим на пересечении этих диагоналей.

15. Для заданной квадратной матрицы сформировать одномерный массив из ее диагональных элементов. Найти след матрицы, суммируя элементы одномерного массива. Преобразовать исходную матрицу по правилу: четные строки разделить на полученное значение, нечетные оставить без изменения.

4. Критерии оценки

5 — Программа написана верно, блок-схема выполнена аккуратно, правильно и соответствует коду программы.

4 — В программе присутствуют синтаксические ошибки, имеются незначительные отклонения в оформлении блок-схемы.

3 — Имеются ошибки в логике алгоритма или неверна блок-схема.

2 — Блок-схема и код отсутствуют или не соответствуют заданию.

5. Пояснения к работе:

Многомерные массивы

Двумерный массив трактуется как одномерный массив, элементами которого являются массивы с указанным в описании типом элементов. Например, оператор

```
Float R[5][10];
```

объявляет массив из пяти элементов, каждый из которых есть массив из десяти вещественных чисел. Отдельные величины этого массива обозначаются именами с двумя индексами: $R[0][0]$, $R[0][1]$, ..., $R[4][9]$. Объединять индексы в одну пару скобок нельзя, т. е. запись $R[2, 3]$ ошибочна. Пример описания трехмерного массива:

```
double X[3][7][20];
```

Порядок расположения элементов многомерного массива в памяти такой, что прежде всего

меняется последний индекс, затем предпоследний и т.д., и лишь один раз пробегает свои значения первый индекс.

При описании многомерных массивов их также можно инициализировать. Делать это удобно так:

```
int M[3][3]={ 11,12,13,  
              21,22,23,  
              31,32,33};
```

Рассмотрим примеры программ обработки матриц — числовых двумерных массивов.

Пример 1. Вычисление и вывод на экран таблицы умножения в форме матрицы Пифагора.

```
#include <iostream>  
using namespace std;  
void main ()  
{ int i,j, a[10][10];  
  for(i=1; i<=9; i++)  
    {for(j=1; j<=9; j++)  
      { a[i][j]=i*j;  
        cout<<a[i][j]<<"  ";  
      }  
      cout<<endl;  
    }  
}
```

Пример 2. Заполнение матрицы случайными числами в диапазоне от 0 до 99 и поиск в ней максимального значения.

```
#include <iostream>
#include <iomanip>
#include <ctime>
#define n 5
using namespace std;
void main ()
{ int i, j , ImaxA, JmaxA, A[n][n];
  srand(time(0));
  //Установка датчика случайных чисел
  for(i=0; i< n; i++)
  { for(j=0; j<n; j++)
    { A[i][j]=rand()%100;
      cout<<setw(6)<<A[i][j];
    }
    cout<<endl;
  }
  ImaxA=JmaxA=0;
  for(i=0; i<n; i++)
  { for(j=0; j<n; j++)
    if (A[i][j]>A[ImaxA][JmaxA])
    { ImaxA=i; JmaxA=j; }
  }
  cout<<"\nМаксимальное_значение="<<A[ImaxA][JmaxA]<<endl;
  system("pause");
}
```

В результате тестирования этой программы получен следующий результат:

46	23	57	35	18
8	48	68	4	70
56	98	16	71	40
70	84	66	67	11
20	44	37	57	38

Максимальное значение: A[2][1]=98

Самостоятельная работа № 7

Тема: Функции-подпрограммы в языке C++

1. Цели работы:

- формирование умений работы с информацией;
- формирование понимания темы;
- формирование умения анализировать;
- закрепление знаний основных моделей, методов построения алгоритмов и синтаксических конструкций языка программирования;
- освоение практических навыков разработки алгоритма и составления программ с использованием функций-подпрограмм.

2. Задача:

Составить в рабочей тетради блок-схему алгоритма и написать программу на языке программирования C++ в соответствии с индивидуальным заданием.

3. Подготовка к работе и порядок выполнения:

- Получение у преподавателя варианта задания для предварительной проработки материала дома.
- Предварительная проработка материала (подготовка необходимых формул и изучение рекомендованного теоретического материала).
- Составление блок-схемы и написание программы.

Список индивидуальных заданий.

1. Вычислить количество отрицательных элементов каждой строки для матриц A(10,10) и B(15,20).
2. Вычислить $Z = (S1 + S2) / (K1 * K2)$, где S1 и K1 – сумма и количество положительных элементов массива X(N); S2 и K2 – сумма и количество положительных элементов массива Y(M). (M, N ≤ 100).
3. Вычислить суммы положительных элементов каждой строки для матриц A(10,12) и B(15,10).
4. Даны три квадратные матрицы n - го порядка. Вывести на печать ту из них, норма которой наименьшая. (Нормой матрицы назовем максимум из абсолютных величин ее элементов).
5. Вычислить количество положительных элементов каждой строки для матриц A(10,10) и B(15,20).
6. Даны три квадратные матрицы n - го порядка. Вывести на печать ту из них, норма которой наибольшая. (Нормой матрицы назовем максимум из абсолютных величин ее элементов).
7. Вычислить суммы отрицательных элементов каждой строки для матриц A(10,12) и B(15,10).
8. Вычислить $Z = (S1 + S2) / 2$, где S1 – сумма положительных элементов массива X(60), S2 – сумма положительных элементов массива Y(50).
9. Вычислить $Z = (S1 + S2) / (K1 * K2)$, где S1 и K1 – сумма и количество отрицательных элементов массива X(N); S2 и K2 – сумма и количество отрицательных элементов массива Y(M). (M, N ≤ 100).
10. Вычислить $Z = (S1 + S2) / 2$, где S1 – количество отрицательных элементов массива X(60), S2 – количество отрицательных элементов массива Y(50).

4. Критерии оценки

- 5 – Программа написана верно, блок-схема выполнена аккуратно, правильно и соответствует коду программы.
- 4 – В программе присутствуют синтаксические ошибки, имеются незначительные отклонения в оформлении блок-схемы.
- 3 – Имеются ошибки в логике алгоритма или неверна блок-схема.
- 2 – Блок-схема и код отсутствуют или не соответствуют заданию.

5. Пояснения к работе

Определение функции. Обращение к функции. В C++ используется лишь один тип подпрограмм — функция. Здесь вообще не принято употреблять термин «подпрограмма», потому что *функция является основной программной единицей в C++*, минимальным исполняемым программным модулем. Всякая программа обязательно включает в себя

основную функцию с именем **main**. Если в программе используются и другие функции, то они выполняют роль подпрограмм.

Рассмотрим пример. Требуется составить программу нахождения наибольшего значения из трех величин — $\max(a, b, c)$. Для ее решения можно использовать вспомогательный алгоритм нахождения максимального значения из двух, поскольку справедливо равенство: $\max(a, b, c) = \max(\max(a, b), c)$.

Вот программа решения этой задачи с использованием вспомогательной функции.

Пример 1.

```
#include <iostream.h>
//Определение вспомогательной функции
int MAX(int x, int y)
{ if (x>y) return x;
  else return y;
}
//Основная функция
void main()
{ int a,b,c,d;
  cout<<"Введите a,b,c:";
  cin>>a>>b>>c;
  d=MAX(MAX(a,b),c);
  cout<<"\nmax(a,b,c)="<<d;
}
```

Формат определения функции следующий:

```
тип имя_функции (спецификация_параметров)
{
  тело_функции
}
```

Тип функции — это тип возвращаемого функцией результата. Если функция не возвращает никакого результата, то для нее указывается тип `void`.

Имя функции — идентификатор, задаваемый программистом или `main` для основной функции.

Спецификации параметров — это либо «пусто», либо список имен формальных параметров функции с указанием типа для каждого из них.

Тело функции — это либо составной оператор, либо *блок*.

Здесь мы впервые встречаемся с понятием блока. Признаком блока является наличие описаний программных объектов (переменных, массивов и т.д.), которые действуют в пределах этого блока. Блок, как и составной оператор, ограничивается фигурными скобками.

В C++ действует правило: *тело функции не может содержать в себе определения других функций*. Иначе говоря, недопустимы внутренние функции, как это делается в Паскале. Из всякой функции возможно обращение к другим функциям, однако они всегда являются внешними по отношению к вызывающей.

Оператором возврата из функции в точку ее вызова является оператор `return`. Он может использоваться в функциях в двух формах:

return; или **return** выражение;

В первом случае функция не возвращает никакого значения в качестве своего результата. Во втором случае результатом функции является значение указанного выражения. Тип этого выражения должен либо совпадать с типом функции, либо относиться к числу типов, допускающих автоматическое преобразование к типу функции.

Оператор `return` может в явном виде отсутствовать в теле функции. В таком случае его присутствие подразумевается перед закрывающей тело функции фигурной скобкой. Такая подстановка производится компилятором.

Формат обращения к функции (вызова функции) традиционный:

имя_функции(список_фактических_параметров)

Однако в C++ обращение к функции имеет своеобразную трактовку: обращение к функции — это выражение. В этом выражении круглые скобки играют роль знака операции, для которой функция и фактические параметры (аргументы) являются операндами. Приоритет операции «скобки» самый высокий, поэтому вычисление функции в выражениях производится раньше других операций.

Между формальными и фактическими параметрами при вызове функции должны соблюдаться правила соответствия *по последовательности* и *по типам*. Фактический параметр — это выражение того же типа, что и у соответствующего ему формального параметра. Стандарт языка Си допускает автоматическое преобразование значений фактических параметров к типу формальных параметров. В Си++ такое преобразование не предусмотрено. Поэтому в дальнейшем мы будем строго следовать принципу соответствия типов.

Самостоятельная работа № 8

Тема: Работа со структурами в языке C++

1. Цели работы:

- формирование умений работы с информацией;
- формирование понимания темы;
- формирование умения анализировать;
- закрепление знаний основных моделей и методов построения алгоритмов;
- освоение практических навыков разработки алгоритма и составления программ с использованием условных операторов.

2. Задача:

Составить в рабочей тетради блок-схему алгоритма и написать программу на языке программирования C++ в соответствии с индивидуальным заданием.

3. Подготовка к работе и порядок выполнения:

- Получение у преподавателя варианта задания для предварительной проработки материала дома.
- Предварительная проработка материала (подготовка необходимых формул и изучение рекомендованного теоретического материала).
- Составление блок-схемы и написание программы.

Список индивидуальных заданий.

1. В столовой предлагается N комплексных обедов, состоящих из Q блюд. Известна стоимость и калорийность каждого блюда. Сколько стоит самый дешевый и самый дорогой обед? Сколько калорий включает в себя самое калорийное блюдо?
2. Торговый склад производит уценку хранящейся продукции. Если продукция хранится на складе дольше n месяцев, то она уценивается в два раза, а если срок хранения превысил m ($m < n$) месяцев, но не достиг n , то – в 1,5 раза. Ведомость уценки товаров должна содержать следующую информацию: наименование товара, количество товара, цена товара до уценки, срок хранения товара, цена товара после уценки, общая стоимость товара до уценки, общая стоимость товара после уценки. Выяснить максимальный и минимальный сроки хранения товаров на складе; максимальную и минимальную цену товаров до уценки и после уценки.
3. N спортсменов – многоборцев принимают участие в соревнованиях по M видам спорта. По каждому виду спорта спортсмен набирает определенное количество очков. Вычислить, сколько очков в сумме набрал каждый спортсмен после окончания

- соревнований. Вычислить разницу в очках для спортсменов, занявших первое и последнее места.
4. N учеников проходили тестирование, выполнив M тестов по какому – либо предмету. Сколько очков набрал каждый ученик по всем темам? Вычислить средний балл, полученный учениками, и разницу между лучшим результатом и средним баллом.
 5. Описать переменную «служащий», состоящую из имени, фамилии, отчества служащего, даты рождения, образования, домашнего адреса, профессии. Определить имена людей с высшим образованием. Выдать данные о служащем, который имеет ту или иную профессию.
 6. Описать переменную «круг», в которой содержатся все данные для построения круга в декартовой системе координат. Определить координаты центра, радиус, площадь и длину окружности круга минимального радиуса, который будет содержать внутри себя все заданные круги.
 7. Описать переменную «экзаменационная ведомость» (предмет, номер группы, номер зачетной книжки, фамилия, имя, отчество студента, его оценки по итогам текущей сессии). Определить отличников, хорошистов, троечников и двоечников.
 8. Описать переменную «расписание», содержащую:
 - день недели;
 - количество пар в этот день;
 - время начала и конца пары;
 - название предмета;
 - фамилию преподавателя.Вывести полную информацию о занятиях, относящихся к предметной области «информатика».
 9. В библиотеке имеются книги, газеты, журналы. Для каждого печатного издания указать:
 - название;
 - год выпуска (для книги), дату выпуска (для газет и журналов);
 - автора (для книги), редактора (для газеты), состав редколлегии (для журнала);
 - объем.Вывести информацию об изданиях, вышедших в заданном году.
 10. Описать переменную «кондиционер» (марка, производительность, мощность, габариты). Вывести на экран данные о кондиционерах максимальных и минимальных габаритов.
 11. Имеются следующие данные о планетах: название, радиус, масса, в какой системе находится. Вывести на экран сведения о планетах, находящихся в Солнечной системе.
 12. Описать переменную «транспорт» (стоимость проезда, количество единиц в парке, рентабельность, название). Выдать данные о наиболее рентабельном виде транспорта.
 13. Дан массив структур, содержащих дату (число, месяц, год). Упорядочить этот массив в порядке «возрастания», т.е. от более ранних значений к более поздним.
 14. Из ассортимента конфет, выпускаемых пермской кондитерской фабрикой, выбрать те, стоимость которых от 30 до 55 рублей за 1 кг. Указать срок их годности и номера магазинов, в которых они имеются в продаже.
 15. Распечатать фамилии рабочих бригады, начинающиеся с букв А и С, с указанием их месячной зарплаты.

4. Критерии оценки

- 5 – Программа написана верно, блок-схема выполнена аккуратно, правильно и соответствует коду программы.
- 4 – В программе присутствуют синтаксические ошибки, имеются незначительные отклонения в оформлении блок-схемы.

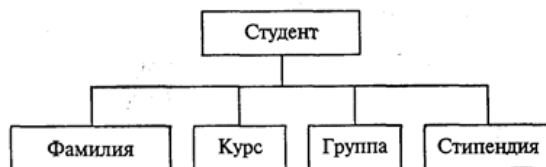
- 3 – Имеются ошибки в логике алгоритма или неверна блок-схема.
2 – Блок-схема и код отсутствуют или не соответствуют заданию.

5. Пояснения к работе:

Структура - это структурированный тип данных, представляющий собой поименованную совокупность разнотипных элементов. Тип структура обычно используется при разработке информационных систем, баз данных.

Правила использования структур рассмотрим на примере.

Сведения о выплате студентам стипендии требуется организовать в виде, показанном на рис.



Элементы такой структуры (фамилия, курс, группа, стипендия) называются полями. Каждому полю должно быть поставлено в соответствие имя и тип.

Формат описания структурного типа следующий:

```
struct имя_типа
    {определения_элементов};
```

В конце обязательно ставится точка с запятой (это оператор). Для рассмотренного примера определение соответствующего структурного типа может быть следующим:

```
struct student{    char fam[30];
                  int kurs;
                  char grup[3];
                  float stip;
                };
```

После этого, `student` становится именем структурного типа, который может быть назначен некоторым переменным. В соответствии со стандартом C++ это нужно делать так:

```
struct student stud1, stud2;
```

Правила C++ разрешают в этом случае служебное слово `struct` опускать и писать

```
student stud1, stud2;
```

Здесь `stud1` и `stud2` — переменные структурного типа. Допускаются и другие варианты описания структурных переменных. Можно вообще не задавать имя типа, а описывать сразу переменные:

```
struct {char fam[30];
        int kurs;
        char grup[3];
        float stip;
      }
stud1, stud2, *pst;
```

В этом примере кроме двух переменных структурного типа объявлен указатель `pst` на такую структуру. В данном описании можно было сохранить имя структурного типа `student`.

Обращение к элементам (полям) структурной величины производится с помощью *уточненного* имени следующего формата:

имя_структуры.имя_элемента

Примеры уточненных имен для описанных выше переменных:

```
stud1.fam;    stud1.stip
```

Значения элементов структуры могут определяться вводом, присваиванием, инициализацией. Пример инициализации в описании:


```
student stud1={"Кротов", 3, "Ф32", 350};
```

Пусть в программе определен указатель на структуру

```
student *pst, stud1;
```

Тогда после выполнения оператора присваивания

```
pst=&stud1;
```

к каждому элементу структурной переменной `stud1` можно обращаться тремя способами.

Например, для поля `fam`

```
stud1.fam или (*pst).fam или pst->fam
```

В последнем варианте используется *знак операции доступа к элементу структуры*: —

>. Аналогично можно обращаться и к другим элементам этой переменной:

```
pst->FIO, pst->grup, pst->stip.
```

Поля структуры могут сами иметь структурный тип. Такие величины представляют многоуровневые деревья.

Допускается использование массивов структур. Например, сведения о 100 студентах могут храниться в массиве, описанном следующим образом:

```
student stud[100];
```

Тогда сведения об отдельных студентах будут обозначаться, например, так: `stud[1].fam`, `stud [5].kurs` и т.п. Если нужно взять первую букву фамилии 25-го студента, то следует писать: `stud[25].fam[0]`.

Самостоятельная работа № 9

Тема: Простейшие программы с использованием визуальных компонентов

1. Цели работы:

- формирование умений работы с информацией;
- формирование понимания темы;
- формирование умения анализировать;
- закрепление знаний основных моделей и методов построения алгоритмов;
- освоение практических навыков разработки алгоритма и составления программ с использованием визуальных компонентов.

2. Задача:

Составить в рабочей тетради блок-схему алгоритма, начертить схему формы проекта и написать код обработчика события в соответствии с индивидуальным заданием.

3. Подготовка к работе и порядок выполнения:

- Получение у преподавателя варианта задания для предварительной проработки материала дома.
- Предварительная проработка материала (подготовка необходимых формул и изучение рекомендованного теоретического материала).
- Составление блок-схемы, выполнение чертежа формы и написание программы.

Список индивидуальных заданий.

1. Найти произведение цифр заданного четырехзначного числа.
2. Дана сторона равностороннего треугольника. Найти радиус описанной окружности.
3. Дано действительное число x . Не пользуясь никакими другими арифметическими операциями, кроме умножения, сложения и вычитания, вычислить за минимальное число операций $2x^4 - 3x^3 + 4x^2 - 5x + 6$.
4. Найти сумму членов арифметической прогрессии, если известны ее первый член,

- знаменатель и число членов прогрессии.
5. Найти (в радианах и в градусах) все углы треугольника со сторонами a , b , c .
 6. Даны натуральные числа M и N . Вывести младшую цифру целой части числа M/N .
 7. Дано натуральное число T , которое представляет длительность прошедшего времени в минутах. Вывести данное значение длительности в часах и минутах в следующей форме: *HH ч MM мин*.
 8. Заданы два вектора с координатами (X_1, Y_1, Z_1) и (X_2, Y_2, Z_2) . Определить угол между векторами.
 9. Задан вектор с координатами (x, y, z) . Найти углы наклона этого вектора к координатным осям.
 10. Даны два комплексных числа в алгебраической форме. Найти их произведение.

4. Критерии оценки

- 5 – Программа написана верно, блок-схема и чертеж формы выполнены аккуратно, правильно и соответствуют коду программы.
- 4 – В программе присутствуют синтаксические ошибки, имеются незначительные отклонения в оформлении блок-схемы или формы приложения.
- 3 – Имеются ошибки в логике алгоритма, неверна блок-схема или отсутствует чертеж формы.
- 2 – Блок-схема, чертёж формы и код отсутствуют или не соответствуют заданию.

5. Пояснения к работе

Основной частью визуального приложения является форма. Поэтому при программировании очень важно уделять внимание внешнему виду форм и функциям. Итак, форма представляет собой пустую доску, которую разработчик оснащает элементами управления, формируя интерфейс пользователя, и кодом для управления данными. Для этого Visual Studio обеспечивает интегрированную среду разработки, способствующую написанию кодов, а также расширенный набор элементов управления .NET Framework. Дополняя функциональными возможностями этих элементов управления свои коды, пользователь может легко и быстро разработать необходимое приложение.

При создании проекта в Visual C++ создается объект, называемый **формой**.

Форма – это главный контейнер, на котором размещаются компоненты среды разработки. Как и любой объект, форма имеет свойства, методы и события.

Элемент управления TextBox

Компонент находится в списке All Windows Forms палитры компонентов. Он задает многострочное или однострочное редактируемое поле, через которые вводят (выводят) строчные данные.

Текст, отображаемый в элементе управления, содержится в свойстве Text. По умолчанию в текстовом поле можно ввести до 2048 знаков. Свойство Text может быть установлено в окне «Свойства» во время разработки, программными средствами во время выполнения или в результате ввода данных пользователем во время выполнения.

Текущее содержимое текстового поля может быть получено во время выполнения путем считывания значения свойства Text.

В следующем примере кода текст помещается в элемент управления во время выполнения.

Пример записи текста в элемент управления TextBox, имеющего имя TextBox1.

```
textBox1->Text = "Это элемент управления TextBox.";
```

Элемент управления Button

Элемент управления Windows Forms Button (кнопка) служит для выполнения действия с помощью мыши. На элементе управления Button может отображаться как текст, так и рисунок. Если нажать кнопку, она выглядит так, словно она нажата и отпущена. При нажатии на кнопку вызывается обработчик событий Click. В обработчик событий Click помещается код, отвечающий за выполнение нужного действия.

Текст, отображающийся на кнопке, содержится в свойстве Text. Внешний вид текста управляется свойством Font и свойством TextAlign.

Чаще всего элемент управления Button в Windows Forms используется для выполнения какой-либо программы при нажатии кнопки.

Элемент управления Label

Элементы управления Windows Forms Label предназначены для отображения текста или изображений, которые пользователь не может изменить с клавиатуры. Они используются для идентификации объектов в форме, например, для описания, что произойдет с элементом управления после выполнения на нем щелчка мышью, или для отображения сведений в ответ на процесс или событие времени выполнения в приложении. Например, имеется возможность использовать надписи для добавления описательных заголовков в текстовые поля, списки, поля со списком и т. д. Кроме того, возможно написание кода, который изменяет текст, отображаемый в надписи, в ответ на события во время выполнения.

Самостоятельная работа № 10

Тема: Обработка одномерных массивов в визуальной среде на языке Visual C++

1. Цели работы:

- формирование умений работы с информацией;
- формирование понимания темы;
- формирование умения анализировать;
- закрепление знаний основных моделей и методов построения алгоритмов;
- освоение практических навыков разработки алгоритма и составления программ с использованием визуальных компонентов для работы с одномерными массивами.

2. Задача:

Составить в рабочей тетради блок-схему алгоритма, начертить схему формы проекта и написать код обработчика события в соответствии с индивидуальным заданием.

3. Подготовка к работе и порядок выполнения:

- Получение у преподавателя варианта задания для предварительной проработки материала дома.
- Предварительная проработка материала (подготовка необходимых формул и изучение рекомендованного теоретического материала).
- Составление блок-схемы, выполнение чертежа формы и написание программы.

Список индивидуальных заданий.

1. Дана последовательность действительных чисел a_1, a_2, \dots, a_n . Указать те ее элементы, которые принадлежат отрезку $[c, d]$.

2. Дана последовательность целых положительных чисел. Найти произведение только тех из них, которые больше заданного числа M . Если таких чисел нет, то выдать сообщение об этом.

3. Последовательность a_1, a_2, \dots, a_n , состоит из нулей и единиц. Поставить в начало этой

последовательности нули, а затем единицы.

4. Даны действительные числа a_1, a_2, \dots, a_n . Среди них есть положительные и отрицательные. Заменить нулями те числа, величина которых по модулю больше максимального числа.

5. В последовательности действительных чисел a_1, a_2, \dots, a_n , есть только положительные и отрицательные элементы. Вычислить произведение отрицательных элементов P_1 и произведение положительных элементов P_2 . Сравнить модуль P_2 с модулем P_1 , указать, какое из произведений по модулю больше.

6. Дан массив действительных чисел. Среди них есть равные. Найти его первый максимальный элемент и заменить его нулем.

7. Дана последовательность действительных чисел $a_1 \leq a_2 \leq \dots \leq a_n$. Вставить в нее действительное число b так, чтобы последовательность осталась неубывающей.

8. Даны целые положительные числа a_1, a_2, \dots, a_n . Найти среди них те, которые являются квадратами некоторого числа m .

9. Дана последовательность целых чисел a_1, a_2, \dots, a_n . Образовать новую последовательность, выбросив из исходной те члены, которые равны $\min(a_1, a_2, \dots, a_n)$.

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

11. Дан целочисленный массив с количеством элементов n . Сжать массив, выбросив из него каждый второй элемент.

12. Задан массив, содержащий несколько нулевых элементов. Сжать его, выбросив эти элементы.

13. Задан массив с количеством элементов N . Сформировать два массива: в первый включить элементы исходного массива с четными номерами, а во второй — с нечетными.

14. Дана последовательность целых чисел a_1, a_2, \dots, a_n . Наименьший член этой последовательности заменить целой частью среднего арифметического всех членов, остальные члены оставить без изменения. Если в последовательности несколько наименьших членов, то заменить последний по порядку.

15. Даны две последовательности целых чисел a_1, a_2, \dots, a_n и b_1, b_2, \dots, b_n . Преобразовать последовательность b_1, b_2, \dots, b_n по следующему правилу: если $a_i \leq 0$, то b_i увеличить в 10 раз, в противном случае b_i заменить нулем ($i = 1, 2, \dots, n$).

4. Критерии оценки

5 – Программа написана верно, блок-схема и чертеж формы выполнены аккуратно, правильно и соответствуют коду программы.

4 – В программе присутствуют синтаксические ошибки, имеются незначительные отклонения в оформлении блок-схемы или формы приложения.

3 – Имеются ошибки в логике алгоритма, неверна блок-схема или отсутствует чертеж формы.

2 – Блок-схема, чертёж формы и код отсутствуют или не соответствуют заданию.

5. Пояснения к работе

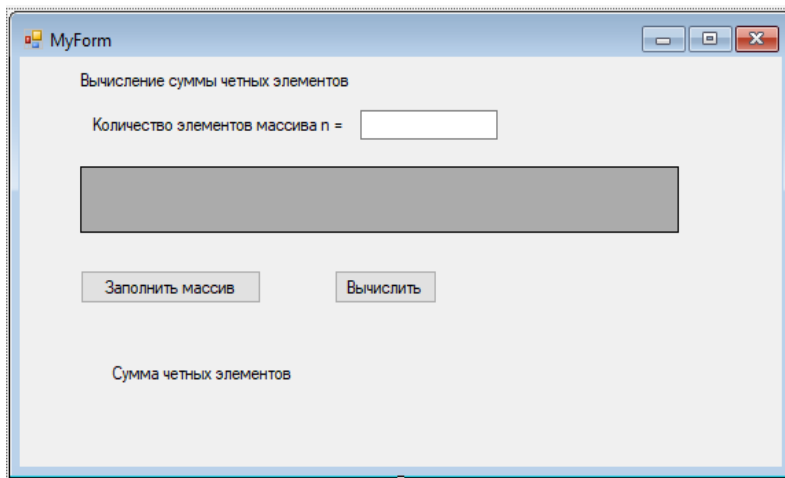
Пример работы с одномерным массивом

Заполнить одномерный массив случайными числами. Вычислить сумму четных элементов массива.

Разместим на форме следующие компоненты:

- Label – 3 метки;
- TextBox – 1 поле ввода;

- Button – 2 кнопки;
- DataGridView– 1 компонент для вывода массива.



Изменим свойство Text меток и кнопок.

Для компонента DataGridView изменим свойства ColumnHeadersVisible и RowHeadersVisible на False.

Это делается для того, чтобы не выводились пустые строка и столбец под заголовок. Свойство AutoSizeColumnsMode изменим на AllCells (для того, чтобы помещалось больше чисел)

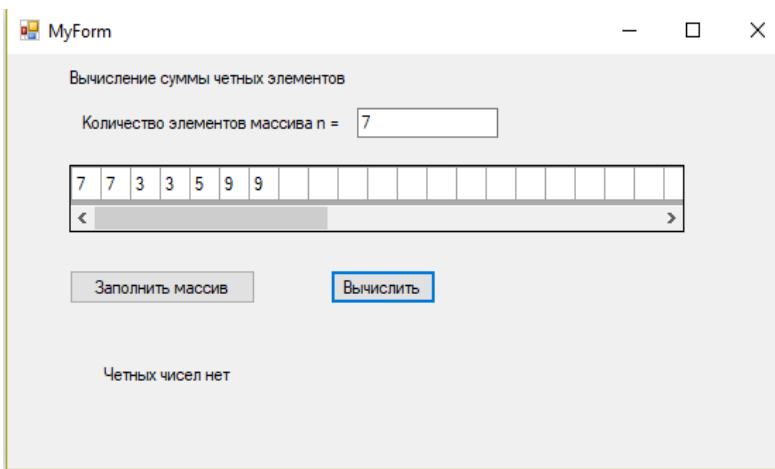
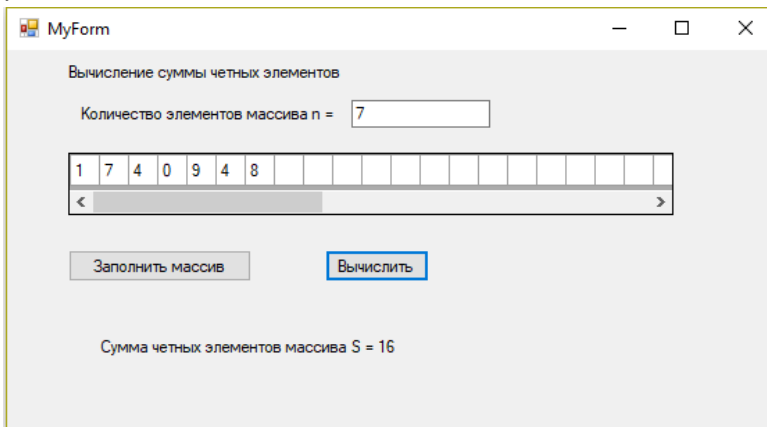
Создадим обработчики нажатия на кнопки и впишем код:

```
#include<stdlib.h>
int n;
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
    int x[50];
    dataGridView1->RowCount = 1;
    dataGridView1->ColumnCount = 50;
    label3->Text = " ";
    for (int i = 0; i < dataGridView1->ColumnCount; i++)
    {
        dataGridView1->Rows[0]->Cells[i]->Value = " ";
    }
    n = Convert::ToInt32(textBox1->Text);
    for(int i=0;i<n;i++)
    {
        x[i] = rand() % 10;
        dataGridView1->Rows[0]->Cells[i] ->Value = x[i]. ToString();
    }
}
private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e) {
    int s = 0;    int x[50];
    for (int i = 0; i < n; i++)
    {
```

```

        x[i] = Convert.ToInt32(dataGridView1-> Rows[0]->Cells[i]->Value);
        if (x[i] % 2 == 0) s+=x[i];
    }
    if (s != 0)
        label3->Text =
            "Сумма четных элементов массива S = " + s.ToString("g");
    else
        label3->Text = "Четных чисел нет ";
}

```



Информационное обеспечение

Для реализации программы библиотечный фонд образовательной организации имеет электронные издания и информационные ресурсы, рекомендуемые для использования в образовательном процессе.

Основные источники:

- Макаровских, Т.А. Языки и методы программирования. Путеводитель по языку C++ /Т.А. Макаровских, А.В. Панюков. - Москва: URSS, 2017.
- Макаровских, Т.А. Языки и методы программирования. Создание простых GUI-приложений с помощью Visual C++ /Т.А. Макаровских, А.В. Панюков. - Москва: URSS, 2017.
- Семакин, И.Г. Основы алгоритмизации и программирования: учебник для студ. учрежд. СПО/И.Г. Семакин, А.П. Шестаков. – Москва.: Академия, 2017.

- Павловская Т.А. С/С++. Программирование на языке высокого уровня: учебник для вузов / Т.А.Павловская. - Санкт-Петербург: Питер, 2020.

Дополнительные источники:

1. Голицына, О.Л. Основы алгоритмизации и программирования: учебное пособие для студ. учрежд. СПО / О.Л. Голицына, И.И. Попов. - 4-е изд., испр. и доп. – Москва: ФОРУМ: ИНФРА-М, 2020.
2. Канцедаль, С.А. Алгоритмизация и программирование: учебное пособие для студ. учрежд. СПО / С.А. Канцедаль. – Москва: ФОРУМ: ИНФРА-М, 2019.
3. Колдаев, В.Д. Основы алгоритмизации и программирования: учебное пособие для студ. учрежд. СПО / В.Д. Колдаев; под ред. Л.Г. Гагариной - Москва: ФОРУМ: ИНФРА-М, 2019.
4. Пахомов, Б.И. С/С++ и MS Visual C++ 2012 для начинающих/Б.И. Пахомов. – Санкт-Петербург: БХВ-Петербург, 2015.
5. Огнева, М.В. Программирование на языке С++: практический курс: учебное пособие для студ. учрежд. СПО. – Москва: Юрайт, 2019.

Интернет-ресурсы:

1. Интернет Университет информационных технологий-Интуит (Национальный Открытый университет). Библиотека учебных курсов. Программирование: учебные курсы. - URL: https://www.intuit.ru/studies/courses?service=0&option_id=13&service_path=1
2. Habrahabr.ru: специализированный портал. Программирование. - URL: <http://habrahabr.ru/blogs/programming>