Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Оренбургский государственный университет»

Кафедра математических методов и моделей в экономике

# РАЗРАБОТКА ПРОГРАММ НА ЯЗЫКЕ C++: ОПЕРАТОРЫ И ЦИКЛЫ

Методические указания

Составители: О.Н. Яркова, О.С. Чудинова

Рекомендовано к изданию редакционно-издательским советом федерального государственного бюджетного образовательного учреждения высшего образования «Оренбургский государственный университет» для обучающихся по образовательной программе высшего образования по направлению подготовки 01.03.04 Прикладная математика

Оренбург
2020

УДК 004.42(076.5)
ББК 32.973я7
Р17

Рецензент – доцент, кандидат экономических наук А.В. Раменская

Р17 Разработка программ на языке С++: операторы и циклы: методические указания / составители О.Н. Яркова, О.С. Чудинова; Оренбургский гос. ун-т.– Оренбург: ОГУ, 2020. – 45 с.

Методические указания содержат теоретические сведения, рекомендации к выполнению и задания для лабораторных работ по курсу «Программирование» на тему «Введение в язык программирования высокого уровня С++».

Методические указания предназначены для обучающихся по образовательной программе высшего образования по направлению подготовки 01.03.04 Прикладная математика, очной формы обучения.

УДК 004.42(076.5)
ББК 32.973я7

© Яркова О.Н.,
Чудинова О.С.,
составление, 2020
© ОГУ, 2020

# Содержание

Введение ..... 4
1 Краткие теоретические сведения ..... 5
1.1 Этапы создания программы на языке C++, алфавит, идентификаторы, ключевые слова ..... 5
1.2 Типы данных ..... 7
1.3 Структура программы ..... 9
1.4 Переменные ..... 10
1.5 Функции ввода/вывода ..... 11
1.6 Условный оператор ..... 13
1.7 Оператор switch ..... 14
1.8 Операторы цикла ..... 16
1.9 Операторы передачи управления ..... 22
2 Примеры программ на языке C++ ..... 24
3 Задания к лабораторным работам, требования к оформлению и защите отчетов по лабораторным работам ..... 34
Список использованных источников ..... 36
Приложение А ..... 37
Приложение Б ..... 40
Приложение В ..... 42

# Введение

Методические указания содержат основные теоретические сведения, задания к лабораторным работам и примеры программ, реализующих алгоритмы линейной структуры, алгоритмы с ветвлениями, циклические структуры на языке программирования C++. Приведены требования к оформлению и защите отчетов по лабораторным работам.

Выполнение лабораторных работ на темы «Разработка программ линейной структуры», «Разработка программ с ветвлениями», «Разработка программ с циклами» по дисциплине «Программирование», относящейся к базовой части блока Д «Дисциплины (модули)», способствует формированию у обучающихся по направлению подготовки 01.03.04 Прикладная математика общепрофессиональной компетенции ОПК-4 Способен разрабатывать и использовать современные методы и программные средства информационно-коммуникационных технологий.

# 1 Краткие теоретические сведения

## 1.1 Этапы создания программы на языке C++, алфавит, идентификаторы, ключевые слова

В тексте на любом естественном языке можно выделить четыре основных элемента: символы, слова, словосочетания и предложения. Из аналогичных конструкций состоит и алгоритмический язык, при этом слова называют лексемами (элементарными конструкциями), словосочетания — выражениями, а предложения — операторами. Лексемы образуются из символов, выражения — из лексем и символов, операторы — из символов, выражений и лексем. Алфавит языка, или его символы — это основные неделимые знаки, с помощью которых создаются тексты на языке программирования. Лексема, или элементарная конструкция, это минимальная единица языка, имеющая самостоятельный смысл. Выражение задает правило вычисления некоторого значения. Оператор задает законченное описание некоторого действия. Для описания сложного действия требуется последовательность операторов. Операторы могут быть объединены в составной оператор, или блок. Операторы бывают исполняемые и неисполняемые. Исполняемые операторы задают действия над данными. Неисполняемые операторы служат для описания данных. Программу на алгоритмическом языке образует объединенная единым алгоритмом совокупность описаний и операторов [1].

Для того чтобы выполнить программу, требуется перевести ее на язык, понятный процессору — в машинные коды. Этот процесс состоит из нескольких этапов (рисунок 1). Сначала программа передается препроцессору, который выполняет директивы, содержащиеся в ее тексте (например, включение в текст так называемых заголовочных файлов — текстовых файлов, в которых содержатся описания используемых в программе элементов). Получившийся полный текст программы поступает на вход компилятора, который выделяет лексемы, а затем на основе грамматики языка распознает выражения и операторы, построенные из этих

лексем. При этом компилятор выявляет синтаксические ошибки и в случае их отсутствия строит объектный модуль.

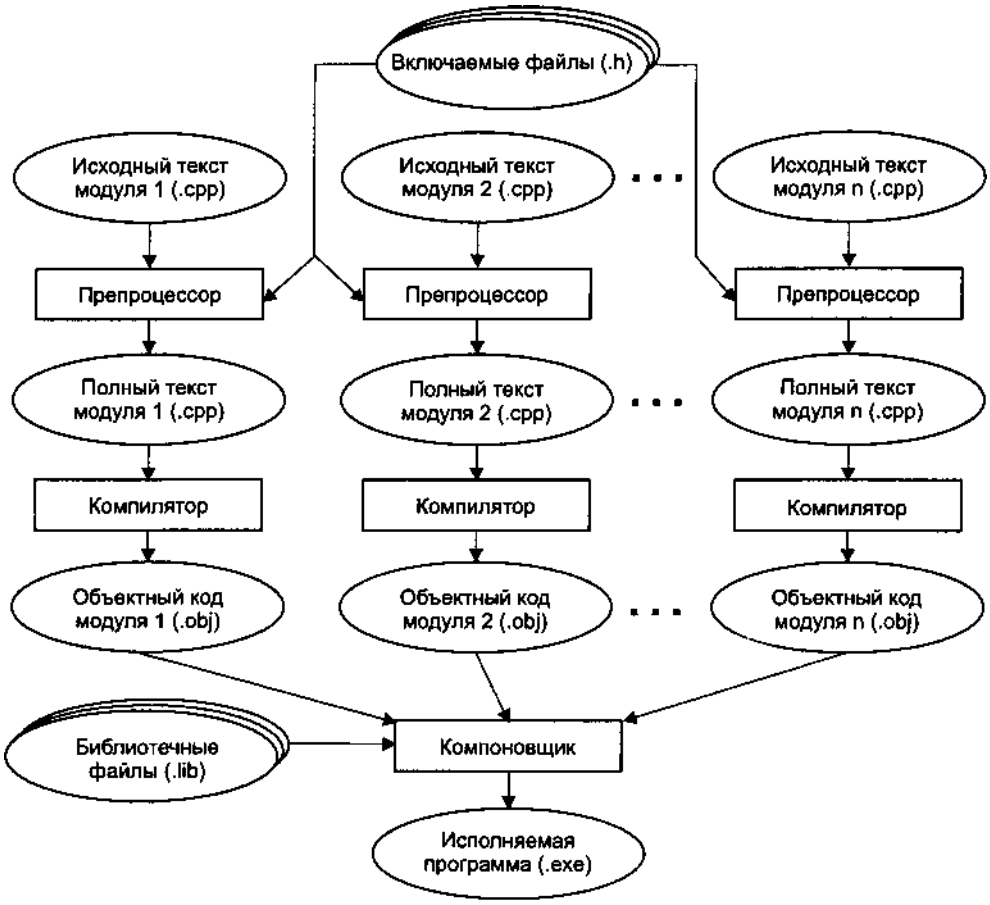



Рисунок 1 – Этапы создания исполняемой программы [1]

Компоновщик, или редактор связей, формирует исполняемый модуль программы, подключая к объектному модулю другие объектные модули, в том числе содержащие функции библиотек, обращение к которым содержится в любой программе (например, для осуществления вывода на экран). Если программа состоит из нескольких исходных файлов, они компилируются по отдельности и объединяются на этапе компоновки. Исполняемый модуль имеет расширение *.exe и запускается на выполнение обычным образом.

Алфавит C++ состоит из следующих элементов: прописные и строчные латинские буквы и знак подчеркивания; арабские цифры от 0 до 9; специальные знаки: ” { } , | [ ] ( ) + - / % * • \ ’ :

? < = > ! & # ~ ; ^; пробельные символы: пробел, символы табуляции, символы перехода на новую строку.

Из символов алфавита формируются лексемы языка: идентификаторы; ключевые (зарезервированные) слова; знаки операций; константы; разделители (скобки, точка, запятая, пробельные символы). Границы лексем определяются другими лексемами, такими, как разделители или знаки операций.

Идентификатор — это имя программного объекта. В идентификаторе могут использоваться латинские буквы, цифры и знак подчеркивания. Прописные и строчные буквы различаются. Первым символом идентификатора может быть буква или знак подчеркивания, но не цифра. Пробелы внутри имен не допускаются. Длина идентификатора по стандарту не ограничена, но некоторые компиляторы и компоновщики налагают на нее ограничения. Идентификатор создается на этапе объявления переменной, функции, типа и т. п., после этого его можно использовать в последующих операторах программы. Идентификатор не должен совпадать с ключевыми словами и именами используемых стандартных объектов языка. Не рекомендуется начинать идентификаторы с символа подчеркивания, поскольку они могут совпасть с именами системных функций или переменных.

Ключевые слова — это зарезервированные идентификаторы, которые имеют специальное значение для компилятора. Список ключевых слов C++: asm; auto; bool; break; case; catch; char; class; const; const_cast; continue; default; delete; do; double; dynamic_cast; else; enum; explicit; export; extern; false; float; for; friend; goto; if; inline; int; long; mutable; namespace; new; operator; private; protected; public; register; reinterpret_cast; return; short; signed; sizeof; static; static_cast; struct; switch; template; this; throw; true; try; typedef; typeid; typename; union; unsigned; using; virtual; void; volatile; wchar_t; while.

### 1.2 Типы данных

Цель любой программы состоит в обработке данных. Данные различного типа хранятся и обрабатываются по-разному. Тип данных определяет: внутреннее

представление данных в памяти компьютера; множество значений, которые могут принимать величины этого типа; операции и функции, которые применимы к величинам этого типа. Исходя из этих характеристик, программист выбирает тип каждой величины, используемой в программе для представления реальных объектов.

В C++ определены следующие основные типы данных: int (целый); char (символьный); wchar_t (расширенный символьный); bool (логический); float (вещественный); double (вещественный с двойной точностью). Первые четыре тина называют целочисленными (целыми), последние два — типами с плавающей точкой.

Существует четыре спецификатора типа, уточняющих внутреннее представление и диапазон значений стандартных типов: short (короткий); long(длинный); signed (знаковый); unsigned (беззнаковый). В таблице 1 приведены диапазоны значений стандартных типов данных. Для вещественных типов в таблице приведены абсолютные величины минимальных и максимальных значений.

Таблица 1 - Диапазоны значений стандартных типов данных

| Тип | Диапазон значений | Размер (байт) |
|---|---|---|
| bool | true и false | 1 |
| signed char | -128 ... 127 | 1 |
| unsigned char | 0 ... 255 | 1 |
| signed short int | -32 768 ... 32 767 | 2 |
| unsigned short int | 0 ... 65 535 | 2 |
| signed long int | -2 147 483 648 ... 2 147 483 647 | 4 |
| unsigned long int | 0 ... 4 294 967 295 | 4 |
| float | 3.4e-38 ... 3.4e+38 | 4 |
| double | 1.7e-308 ... 1.7e+308 | 8 |
| long double | 3.4e-4932 ... 3.4e+4932 | 10 |

Кроме перечисленных, к основным типам языка относится тип void, но множество значений этого типа пусто. Он используется для определения функций, которые не возвращают значения, для указания пустого списка аргументов функции, как базовый тип для указателей и в операции приведения типов.

### 1.3 Структура программы

Программа на языке C++ состоит из функций, описаний и директив препроцессора. Одна из функций должна иметь имя main. Выполнение программы начинается с первого оператора этой функции. Простейшее определение функции имеет формат, приведенный на рисунке 2.

```
<тип_возвращаемого_значения> <имя> ([ <параметры> ]) {
операторы, составляющие тело функции;
}
```

Рисунок 2 – Простейшее определение функции

Здесь и далее указание квадратных скобок означает, что стоящие в них параметры, могут отсутствовать. Как правило, функция используется для вычисления какого-либо значения, поэтому перед именем функции указывается его тип. Если функция не должна возвращать значение, указывается тип void. Тело функции является блоком и, как и любой другой блок в программах C++, заключается в фигурные скобки. Функции не могут быть вложенными. Каждый оператор в программах C++ заканчивается точкой с запятой (кроме составного оператора). Структура программы на языке C++ приведена на рисунке 3.

```
<Подключение заголовочных файлов>
<Объявление глобальных переменных>
<Объявление функций, классов и др.>
int main( [ int argc, char *argv[] ]) {
<Инструкции>
return 0;
}
<Определение функций и классов>
```

Рисунок 3 – Структура программы на языке C++

В программе могут присутствовать комментарии (неисполняемая часть программы, служит для пояснений, исключения блоков при отладке и т.п.).

Комментарий либо начинается с двух символов «прямая косая черта» (//) и заканчивается символом перехода на новую строку, либо заключается между символами-скобками /* и */. Внутри комментария можно использовать любые допустимые на данном компьютере символы.

### 1.4 Переменные

Переменная — это именованная область памяти, в которой хранятся данные определенного типа. У переменной есть имя и значение. Имя служит для обращения к области памяти, в которой хранится значение. Во время выполнения программы значение переменной можно изменять. Перед использованием любая переменная должна быть описана.

Пример описания целой переменной с именем a и вещественной переменной x:

```
int a;
float x;
```

Общий вид *оператора описания* переменных:

[класс памяти] [const] тип имя [инициализатор];

Необязательный параметр «класс памяти» может принимать одно из значений auto, extern, static и register (подробнее можно ознакомиться в источнике [1]).

При описании переменной можно присвоить начальное значение (инициализировать переменную), указав справа от имени переменной знак равенства и значение или, указав значение в круглых скобках. Если при определении начальное значение переменных явным образом не задается, компилятор присваивает глобальным и статическим переменным нулевое значение соответствующего типа. Имя переменной должно быть уникальным в своей области действия.

Модификатор const показывает, что значение переменной изменять нельзя. Такую переменную называют константой. Константа должна быть инициализирована при объявлении обязательно. В одном операторе можно описать

несколько переменных одного тина, разделяя их запятыми:

```
short int b = 1;      // целая переменная a
const char C = 'C'; // символьная константа C
char s, sf = 'f';      // инициализация относится только к sf
int t (54);
float d = 0.22, x(3), sum;
```

Описание переменной задает ее область действия (та часть программы, в которой идентификатор можно использовать для доступа к связанной с ним области памяти). Класс памяти и область действия зависят от места размещения идентификатора в тексте программы. В зависимости от области действия переменная может быть локальной или глобальной. Если переменная определена внутри блока, она является локальной, область ее действия — от точки описания до конца блока, включая вложенные блоки. Если переменная определена вне любого блока, она называется *глобальной* и областью ее действия считается весь файл с программой от точки описания до его конца.

### 1.5 Функции ввода/вывода

В языке C++ ввод/вывод осуществляется с помощью функций, типов и объектов, содержащихся в стандартных библиотеках. Используется два способа: функции, унаследованные из языка C, и объекты C++.

Основные функции ввода/вывода с использованием библиотеки классов C++:

```
#include <iostream> // подключаем файл-библиотеку функций
std::cin>>a; // ввод переменной a
std::cout << "a="<<a<<endl; // вывод, здесь «endl» – оператор для перевода строки.
```

Все объекты будут доступны через пространство имен std, для того чтобы при каждом вызове объекта не указывать пространство имен «std», необходимо после подключения модулей указать инструкцию using namespace std (рисунок 4).

Пример программы, использующей функции ввода/вывода в стиле С приведен на рисунке 4.

```
#include <iostream.h>
using namespace std;
int main(){
int  a;
cout << "Введите целое число\n";
cin>> a;
cout <<"Вы ввели число: " << a << ". Спасибо!";
cin.ignore(200,'\n').get(); /* предотвращает автоматическое закрытие окна
консоли до нажатия любой клавиши*/
return 0;
}
```

Рисунок 4 – Пример программы с функциями ввода/вывода в стиле С++

В примере на рисунке 4 присутствует оператор «cin.ignore(200,'\n').get();», который позволяет предотвратить автоматическое закрытие окна консоли до нажатия любой клавиши.  Если в программе нет операторов чтения данных, для этих целей можно использовать сокращенную форму оператора: «cin.get();».

Основные функции ввода/вывода в стиле С:

#include <stdio.h> // подключаем файл – библиотеку функций

using namespace std; // подключаем пространство имен «std»

int scanf (const char* format, ... ) // функция ввода

int printf(const char* format, ... ) // функция вывода

Они выполняют форматированный ввод и вывод произвольного количества величин в соответствии со строкой формата format [2]. Внутри этой строки можно указать обычные символы и спецификаторы формата, начинающиеся с символа %. Вместо спецификаторов формата подставляются значения, указанные в качестве параметров. Количество спецификаторов должно совпадать с количеством переданных параметров. Спецификаторы имеют следующий синтаксис:

%[<Модификаторы>][<Ширина>][.<Точность>]<Тип>

В параметре <Тип> могут быть указаны следующие символы:

c — символ, пример: printf("%c", 'w');  // w

s — строка, пример:  printf("%s", "String"); // String

d или i — десятичное целое со знаком, пример: printf("%d Si", 10, 30);//10 30

u — десятичное целое без знака, пример: printf("%u", 10);  // 10

o — восьмеричное число без знака, пример: printf("%o", 10); // 12

x/X — шестнадцатеричное без знака в нижнем (верхнем) регистре, пример: printf("%x",10); //a

f — вещественное в десятичном представлении, пример: printf("%f ",12.5); //12.500000

e/E—вещественное число в экспоненциальной форме (буква"e" в нижнем (верхнем) регистре), пример: printf("%e", 18657.81452); // 1.865781e+004

Пример программы, использующей функции ввода/вывода в стиле C приведен на рисунке 5.

```
#include <stdio.h>
using namespace std;
int main(){
    int a;
    printf("Введите целое число\n");// «\n» - управляющий символ перевода строки
    scanf("%d", &a);
    printf("Вы ввели число %d . Спасибо!", a);
    cin.ignore(200,'\n').get();
    return 0;
}
```

Рисунок 5 – Пример программы с функциями ввода/вывода в стиле C

### 1.6 Условный оператор

Условный оператор if используется для разветвления процесса вычислений на два направления (рисунок 6).

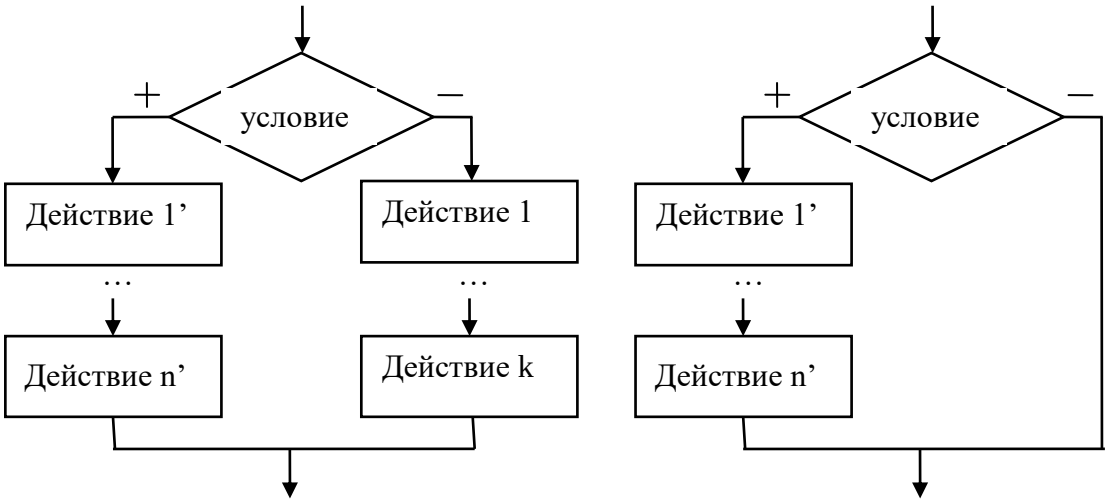



Рисунок 6 – Блок-схемы полного и неполного оператора ветвления

Формат оператора ветвления:

if ( выражение ) оператор_1; [else оператор_2;]

Сначала вычисляется выражение. Если оно не равно нулю (имеет значение true), выполняется первый оператор, иначе — второй. После этого управление передается па оператор, следующий за условным.

Одна из ветвей может отсутствовать, при этом логичнее опускать вторую ветвь вместе с ключевым словом else. Если в какой-либо ветви требуется выполнить несколько операторов, их необходимо заключить в фигурные скобки, эти операторы образуют блок. Блок может содержать любые операторы, в том числе описания и другие условные операторы (но не может состоять из одних описаний). Пример программы, использующей условный оператор приведен на рисунке 7.

### 1.7 Оператор switch

Оператор switch (переключатель) предназначен для разветвления процесса вычислений на несколько направлений.

Формат оператора:

switch ( выражение ){

case константное_выражение_1: [список__операторов_1]; [break;]

…

case константное_выражение_n: [список_операторов_n]; [break;]

[default: операторы ];

}

```
/* Программа проверяет на четность(нечетность) целое положительное число*/
#include < iostream.h >
using namespace std;
int main(){
    int a;
    cout<<"Введите целое положительное число\n";
    cin>>a;
    if (a<0) cout<<"Вы ввели отрицательное число\n";
    else {
        if (a % 2==0) cout<<"Вы ввели четное число\n";
                else cout<<"Вы ввели нечетное число\n";
    }
    cin.ignore(200,'\n').get();
    return 0;
}
```

Рисунок 7 – Пример программы с условным оператором

Вместо условия оператор switch принимает выражение. В зависимости от значения выражения выполняется один из блоков case, в котором указано это значение. Значением выражения должно быть целое число или символ. Если ни одно из значений не описано в блоках case, то выполняется блок default (если он указан). Значения в блоках case не могут иметь одинаковые константы. Оператор break позволяет досрочно выйти из оператора выбора switch. Если не указать оператор break в блоке case, то будет выполняться следующий блок case вне зависимости от указанного значения. В некоторых случаях это бывает полезно. Пример программы с оператором выбора представлен на рисунке 8.

```
/* Программа вычисления значения заданного выражения (суммы, разности,
   произведения, частного, остатка от деления для двух элементов*/
#include < iostream.h >
using namespace std;
int main(){
   int a, b, res;
   char op;
   cout<<"Введите первый операнд\n";        cin>>a;
   cout<<"Введите операцию (+,-,*,/,%)\n";   cin>>op;
   cout<<"Введите второй операнд\n";         cin>>b;
   bool f;
   switch (op)
   case ‘+’: res=a+b; break;
   case ‘*’: res=a*b; break;
   case ‘-’: res=a-b; break;
   case ‘/’: res=a/b; break;
   case ‘%’: res=a%b; break;
   default cout<<"Вы ввели  неизвестную операцию\n"; f=false;
   if (f) cout<<"Результат выполнения операции: "<<a<<op<<b<<"="<<res<<"\n";
   cin.ignore(200,'\n').get();
   return 0;
}
```

Рисунок 8 – Пример программы с оператором выбора

### 1.8 Операторы цикла

Операторы цикла используются для организации многократно повторяющихся вычислений. Цикл состоит из тела цикла, то есть операторов, которые выполняются несколько раз, начальных установок, модификации параметра цикла и проверки условия продолжения выполнения цикла. Один проход цикла называется итерацией. Проверка условия выполняется на каждой итерации либо до тела цикла (тогда говорят о цикле с предусловием), либо после тела цикла (цикл с постусловием). Тело цикла с постусловием всегда выполняется хотя бы один раз, затем проверяется, нужно ли его выполнять еще раз. Проверка необходимости выполнения цикла с предусловием делается до тела цикла, поэтому возможно, что он не выполнится ни разу. Переменные, изменяющиеся в теле цикла и используемые при проверке условия продолжения, называются параметрами цикла.

Целочисленные параметры цикла, изменяющиеся с постоянным шагом на каждой итерации, называются счетчиками цикла.

Начальные установки могут явно не присутствовать в программе, их смысл состоит в том, чтобы до входа в цикл задать значения переменным, которые в нем используются.

Цикл завершается, если условие его продолжения не выполняется. Возможно принудительное завершение как текущей итерации, так и цикла в целом. Для этого служат операторы break, continue, return и goto. Передавать управление извне внутрь цикла не рекомендуется.

Для удобства, а не по необходимости, в C++ есть три разных оператора цикла — while, do while и for. Рассмотрим цикл с предусловием (while). Блок-схема цикла с предусловием (цикла «пока») представлена на рисунке 9.

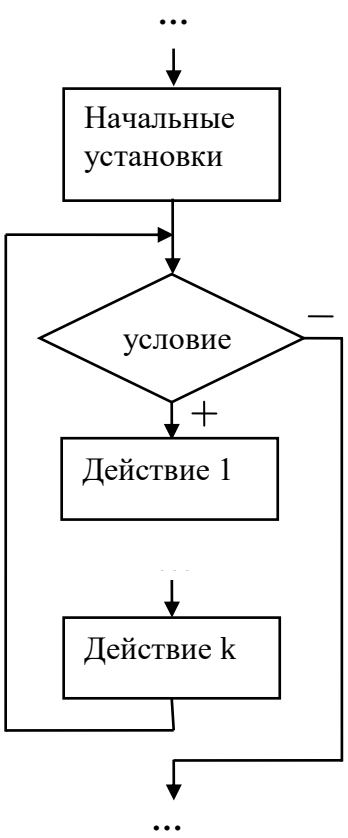



Рисунок 9 – Блок-схема цикла с предусловием (цикл «пока»)

Формат оператора цикла с предусловием (while):

<Начальные установки>;

```
while (<Условие>) {
<Инструкции>;
}
```

Последовательность работы цикла while:

1. Переменным, входящим в условие цикла присваиваются начальные значения.
2. Проверяется условие, если оно истинно, выполняются инструкции внутри цикла, иначе выполнение цикла завершается.
3. Инструкции внутри цикла обязательно должны содержать действия изменяющие переменные, входящие в условие цикла.
4. Переход к пункту 2.

Пример программы с циклом while представлен на рисунке 10.

```
/* Программа выводит на экран консоли значения от 1 до 100 */
#include < iostream.h >
using namespace std;
int main(){
cout<<"Числа 1 до 100"<<endl;
int i=1;
while (i<=100) {
  cout<<i<<" ";
  i++; //модификация параметра, входящего в условие цикла
}
cout<<endl;
cin.ignore(200,'\n').get();
return 0;
}
```

Рисунок 10 – Пример программы с циклом while

Рассмотрим цикл с постусловием (do while). Блок-схема цикла с постусловием (цикла «до») представлена на рисунке 11.

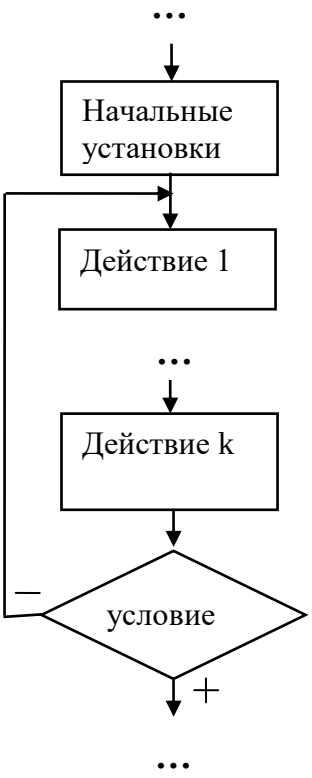



Рисунок 11– Блок-схема цикла с постусловием (цикл «до»)

Формат оператора цикла с постусловием (do while):

<Начальные установки>;

do {

<Инструкции>;

} while (<Условие>);

Последовательность работы цикла do...while.

1. Переменным, входящим в условие цикла присваиваются начальные значения.

2. Выполняются инструкции внутри цикла.

3. Инструкции внутри цикла обязательно должны содержать действия изменяющие переменные, входящие в условие цикла.

4. Проверяется условие, если оно истинно, происходит переход к п. 2, а если нет — выполнение цикла завершается.

Пример программы с циклом while представлен на рисунке 12.

```
/* Программа выводит на экран консоли значения от 1 до 100 */
#include < iostream.h >
using namespace std;
int main(){
cout<<"Числа 1 до 100"<<endl;
int i=1;
do {
  cout<<i<<" ";
  i++; //модификация параметра, входящего в условие цикла
} while (i>100);
cout<<endl;
cin.ignore(200,'\n').get();
return 0;
}
```

Рисунок 12 – Пример программы с циклом do while

Рассмотрим цикл с параметром (for). Блок-схема цикла с параметром представлено на рисунке 13.

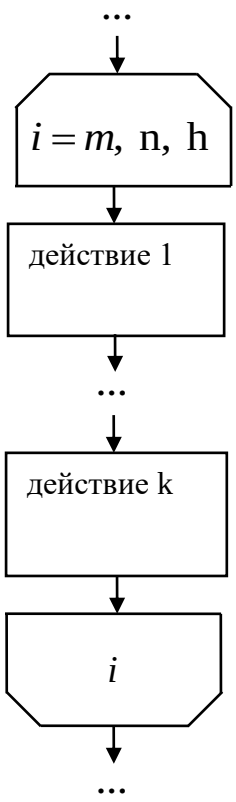



Рисунок 13 – Блок-схема цикла с параметром

В блоке модификации указывается закон изменения переменной параметра: *m* – начальное значение параметра цикла; *h* – шаг; *n* – последнее значение параметра цикла. Формат оператора цикла с параметром имеет вид:

for (<Начальное значение>; <Условие>; <Приращение>) {

<Инструкции>

}

Параметры имеют следующие значения: <Начальное значение> — присваивает переменной-счетчику начальное значение; <Условие> — содержит логическое выражение. Пока логическое выражение возвращает значение true, выполняются инструкции внутри цикла; <Приращение> — задает изменение переменной-счетчика на каждой итерации. Областью действия переменных, объявленных в части инициализации цикла, является цикл.

Последовательность работы цикла for:

1. Переменной-счетчику присваивается начальное значение.
2. Проверяется условие, если оно истинно, выполняются выражения внутри цикла, а в противном случае выполнение цикла завершается.
3. Переменная-счетчик изменяется на величину, указанную в блоке <Приращение>.
4. Переход к п. 2.

Пример программы с циклом for представлен на рисунке 14.

```
/* Программа выводит на экран консоли значения от 1 до 100 */
#include < iostream.h >
using namespace std;
int main(){
cout<<"Числа 1 до 100"<<endl;
for (int i=1; i<=100; i++){
  cout<<i<<" ";
}
cout<<endl;
cin.ignore(200,'\n').get();
return 0;
}
```

Рисунок 14 – Пример программы с циклом for

Все параметры цикла for и инструкции внутри цикла являются необязательными. Хотя параметры можно не указывать, точки с запятой обязательно должны быть. Если все параметры не указаны, то цикл будет бесконечным. Чтобы выйти из бесконечного цикла следует использовать оператор break.

### 1.9 Операторы передачи управления

В C++ есть четыре оператора управления: оператор безусловного перехода goto; оператор выхода из цикла break; оператор перехода к следующей итерации цикла continue; оператор возврата из функции return. Они позволяют изменить естественный порядок выполнения вычислений.

Оператор break используется внутри операторов цикла или switch для обеспечения перехода в точку программы, находящуюся непосредственно за оператором, внутри которого находится break (пример приведен выше).

Оператор перехода к следующей итерации цикла continue пропускает все операторы, оставшиеся до конца тела цикла, и передает управление на начало следующей итерации.

Оператор возврата из функции return завершает выполнение функции и передает управление в точку ее вызова. Вид оператора:

return [ выражение ];

Выражение должно иметь скалярный тип. Если тип возвращаемого функцией значения описан как void, выражение должно отсутствовать.

Оператор безусловного перехода goto передает управление на помеченный оператор. Имеет формат:

goto метка;

В теле той же функции должна присутствовать ровно одна конструкция вида:

метка: оператор;

Метка — это идентификатор, областью видимости которого является функция.

Применение goto нарушает принципы структурного и модульного программирования, по которым все блоки, из которых состоит программа, должны иметь только один вход и один выход. Использование оператора безусловного перехода оправдано только в двух случаях: принудительный выход вниз по тексту программы из нескольких вложенных циклов или переключателей; переход из нескольких мест функции в одно (например, если перед выходом из функции всегда необходимо выполнять какие-либо действия). В остальных случаях для записи любого алгоритма существуют более подходящие средства и goto не используют! В любом случае не следует передавать управление внутрь операторов if, switch и циклов, а так же внутрь блоков, содержащих инициализацию переменных[1].

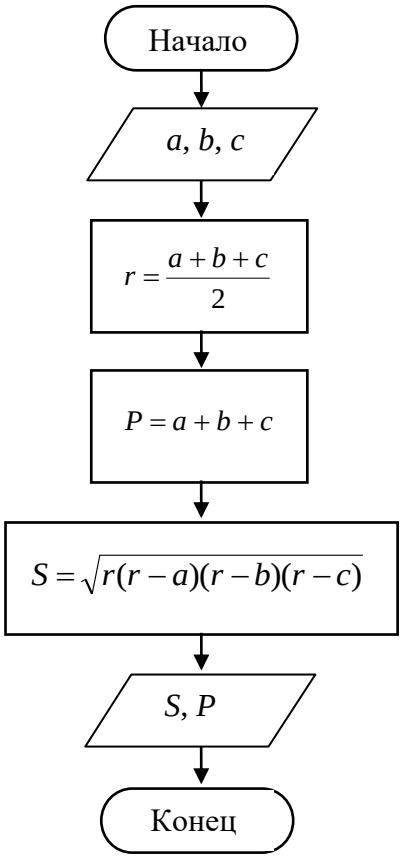
## 2 Примеры программ на языке C++

*Задача 1 (Программа линейной структуры)*

Написать программу для вычисления площади и периметра треугольника, зная длины сторон треугольника.

Рассмотрим теоретическое обоснование решения задачи. Пусть $a$, $b$, $c$ – длины сторон треугольника. Необходимо найти $S$ – площадь треугольника, $P$ – периметр. Для нахождения площади треугольника можно воспользоваться формулой Герона: $S = \sqrt{r(r-a)(r-b)(r-c)}$, где $r$ – полупериметр треугольника, $r = \frac{P}{2}$, $P = a + b + c$ – периметр треугольника.

Блок-схема алгоритма решения задачи и текст программы на языке C++ представлены на рисунке 15.




```
#include <iostream>
#include <math.h>
using namespace std;
int main() {
      float a, b, c;
      cout <<"Input a = ";
      cin >> a;
      cout <<"Input b = ";
      cin >> b;
      cout <<"Input c = ";
      cin >> c;
      float r, S;
      r=(a+b+c)/2.0;
      float P;
      P=a+b+c;
      S=sqrt(r*(r-a)*(r-b)*(r-c));
      cout << "S = " << S << endl;
      cout << "P = " << P << endl;
      cin.ignore(200,'\n').get();
      return 0;
}
```

Рисунок 15 – Блок-схема алгоритма решения задачи 1 и текст программы на языке C++

Проведем тестирование программы. На рисунке 16 представлены входные данные и результат работы программы по решению задачи 1.

```
Input a = 10
Input b = 15
Input c = 10
S = 49.6078
P = 35
```

Рисунок 16 – Входные данные и результат работы программы по решению задачи 1

Для поверки правильности работы программы по указанным значениям входных параметров выполнены расчеты в пакете Excel. Результаты представлены на рисунке 17.

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Входная информация: | | | Результаты расчетов: | |
| 2 | a= | 10 | | Периметр P= | =B2+B3+B4 |
| 3 | b= | 15 | | Полупериметр r= | =E2/2 |
| 4 | c= | 10 | | Площадь S= | =КОРЕНЬ(E3*(E3-B2)*(E3-B3)*(E3-B4)) |
| 5 | | | | | |

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Входная информация: | | | Результаты расчетов: | |
| 2 | a= | 10 | | Периметр P= | 35 |
| 3 | b= | 15 | | Полупериметр r= | 17,5 |
| 4 | c= | 10 | | Площадь S= | 49,6078 |
| 5 | | | | | |

Рисунок 17 – Результаты решения задачи 1 в пакете Excel (в режимах вывода формул и вывода результатов)

Результаты работы программы и расчетов в пакете Excel совпали.

*Задача 2 (Программа с ветвлениями)*

Написать программу для нахождения наименьшего из трех чисел.

Пусть $a$, $b$, $c$ – три числа, вводимые пользователем.

Блок-схема алгоритма решения задачи представлена на рисунке 18. Текст программы, реализующей этот алгоритм, представлен на рисунке 19.

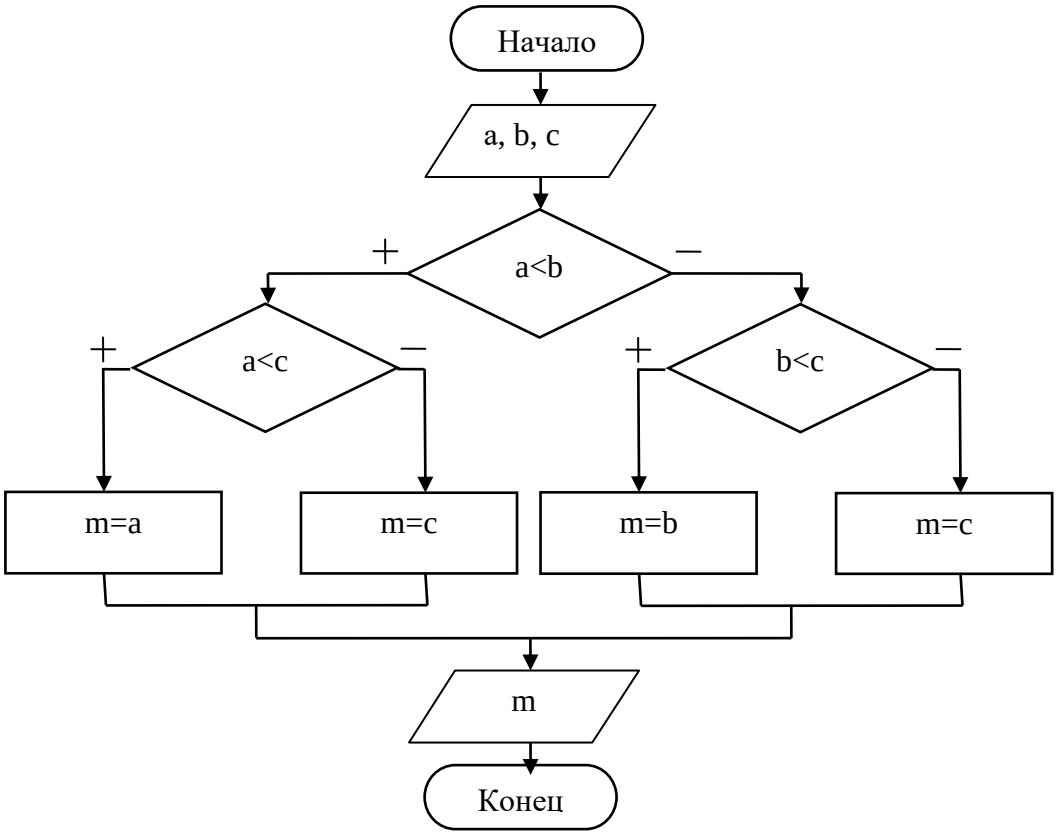



Рисунок 18 – Блок-схема алгоритма решения задачи 2

```
#include <iostream>
using namespace std;
int main() {
      float a, b, c, m;
      cout <<"Input a = ";
      cin >> a;
      cout <<"Input b = ";
      cin >> b;
      cout <<"Input c = ";
      cin >> c;
      if (a<b){
            if (a<c)
                  m=a;
            else m=c;}
      else {if (b<c)
                  m=b;
            else m=c;}
      cout << "min = " << m << endl;
      cin.ignore(200,'\n').get();
      return 0;
}
```

Рисунок 19 – Текст программы для решения задачи 2

Проведем тестирование программы. На рисунке 20 представлены входные данные и результат работы программы по решению задачи 2.

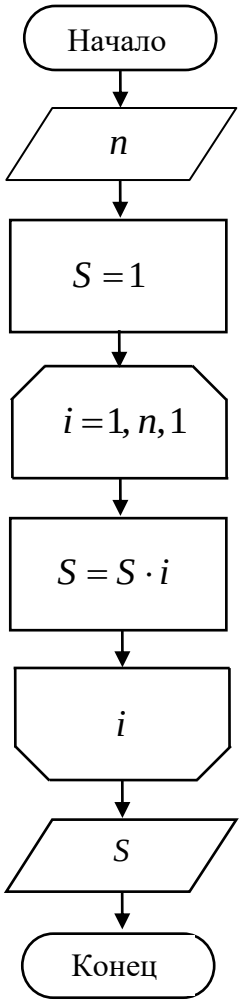
```
Input a = 150
Input b = 3
Input c = -1
min = -1
```

Рисунок 20 – Входные данные и результат работы программы по решению задачи 2

*Задача 3 (программа с циклом со счетчиком)*

Написать программу для вычисления факториала натурального числа *n*.

Блок-схема алгоритма решения задачи и текст программы на языке C++ представлены на рисунке 21.




```
#include <iostream>
using namespace std;
int main() {
      int n,S;
      cout <<"Input n = ";
      cin >> n;
      S=1;
      for (int i=1; i<=n; ++i) S=S*i;
      cout << "n! = " << S << endl;
      cin.ignore(200,'\n').get();
      return 0;
}
```

Рисунок 21 – Блок-схема алгоритма решения задачи 3 и текст программы на языке C++

Проведем тестирование программы. На рисунке 22 представлены входные данные и результат работы программы по решению задачи 3.

```
Input n = 5
n! = 120
```

Рисунок 22 – Входные данные и результат работы программы по решению задачи 3

Результаты расчетов в пакете Excel представлены на рисунке 23.

| | A | B |
|---|---|---|
| 1 | | |
| 2 | n= | 5 |
| 3 | n!= | =ФАКТР(B2) |

| | A | B |
|---|---|---|
| 1 | | |
| 2 | n= | 5 |
| 3 | n!= | 120 |

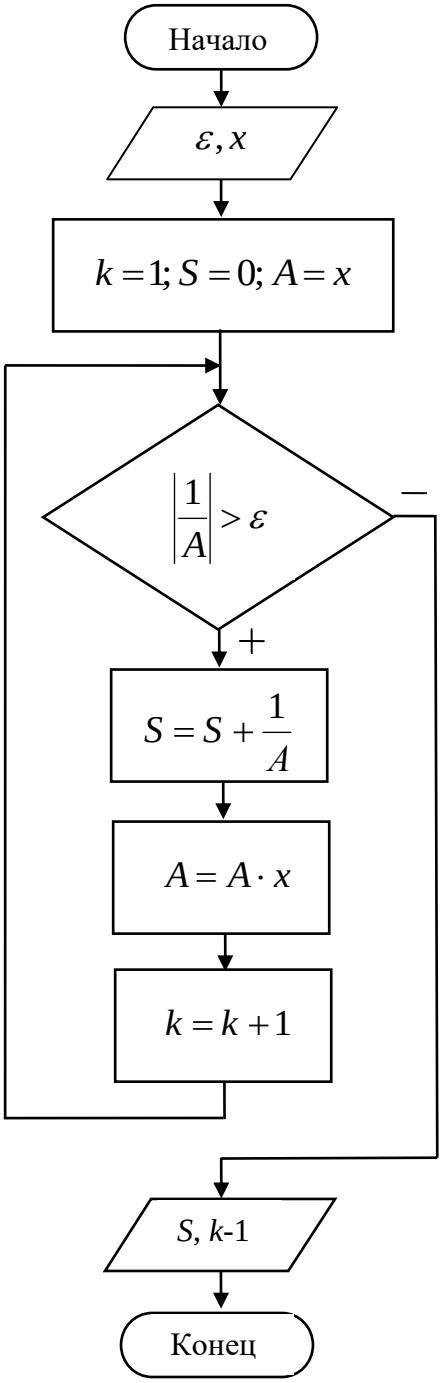
Рисунок 23 – Результаты решения задачи 3 в пакете Excel (в режимах вывода формул и вывода результатов)

*Задача 4 (программа с циклом с предусловием)*

Написать программу для вычисления с точностью до $\varepsilon$, $\varepsilon > 0$ значения выражения $\sum_{k=1}^{\infty} \frac{1}{x^k}$, $x \neq 0$.

Блок-схема алгоритма решения задачи с использованием цикла с предусловием и текст программы на языке C++ представлены на рисунке 24.

Проведем тестирование программы. На рисунке 25 представлены входные данные и результаты работы программы по решению задачи 4.




```
#include <cmath>
using namespace std;
int main() {
      float e,x,S,A;
      int k;
      cout <<"Input e = ";
      cin >> e;
      cout <<"Input x = ";
      cin >> x;
      k=1;
      S=0;
      A=x;
      while (abs(1/A)>e) {
            S=S+1/A;
            A=A*x;
            k=k+1;
      }
      cout <<"S=" << S << endl;
      cout <<"k=" << k-1 << endl;
      cin.ignore(200,'\n').get();
      return 0;
}
```

Рисунок 24 – Блок-схема алгоритма решения задачи 4 с использованием цикла с предусловием и текст программы на языке C++

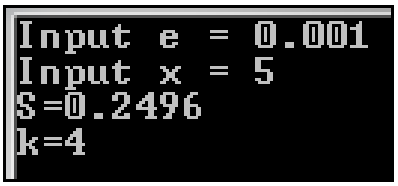



Рисунок 25 – Входные данные и результаты работы программы по решению задачи 4

Результаты расчетов в пакете Excel представлены на рисунке 26.

| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | k | k=1 | k=2 | k=3 | k=4 | k=5 |
| 2 | e= | 0,001 | | 1 | =1/(B3^D2) | =1/($B$3^$D2) | =1/($B$3^$D2) | =1/($B$3^$D2) | =1/($B$3^$D2) |
| 3 | x= | 5 | | 2 | | =1/($B$3^D3) | =1/($B$3^$D3) | =1/($B$3^$D3) | =1/($B$3^$D3) |
| 4 | | | | 3 | | | =1/($B$3^$D4) | =1/($B$3^$D4) | =1/($B$3^$D4) |
| 5 | | | | 4 | | | | =1/($B$3^$D5) | =1/($B$3^$D5) |
| 6 | | | | 5 | | | | | =1/($B$3^$D6) |
| 7 | | | | 6 | | | | | |
| 8 | | | | | | | | | |
| 9 | | | | Сумма= | =СУММ(E2:E7) | =СУММ(F2:F7) | =СУММ(G2:G7) | =СУММ(H2:H7) | =СУММ(I2:I7) |

| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | k | k=1 | k=2 | k=3 | k=4 | k=5 |
| 2 | e= | 0,001 | | 1 | 0,2 | 0,2 | 0,2 | 0,2 | 0,2 |
| 3 | x= | 5 | | 2 | | 0,04 | 0,04 | 0,04 | 0,04 |
| 4 | | | | 3 | | | 0,008 | 0,008 | 0,008 |
| 5 | | | | 4 | | | | 0,0016 | 0,0016 |
| 6 | | | | 5 | | | | | 0,00032 |
| 7 | | | | 6 | | | | | |
| 8 | | | | | | | | | |
| 9 | | | | Сумма= | 0,2 | 0,24 | 0,248 | 0,2496 | 0,24992 |

Рисунок 26 – Результаты решения задачи 4 в пакете Excel (в режимах вывода формул и вывода результатов)

Решим эту задачу с использованием цикла с постусловием. Текст программы представлен на рисунке 27.

```
#include <iostream>
#include <cmath>
using namespace std;
int main() {
	float e,x,S,A;
	int k;
	cout <<"Input e = ";
	cin >> e;
	cout <<"Input x = ";
	cin >> x;
	k=0;
	S=0;
	A=x;
	do {
		S=S+1/A;
		A=A*x;
		k=k+1;
	} while (abs(1/A)>e);
	cout <<"S=" << S << endl;
	cout <<"k=" << k << endl;
	cin.ignore(200,'\n').get();
	return 0;
}
```

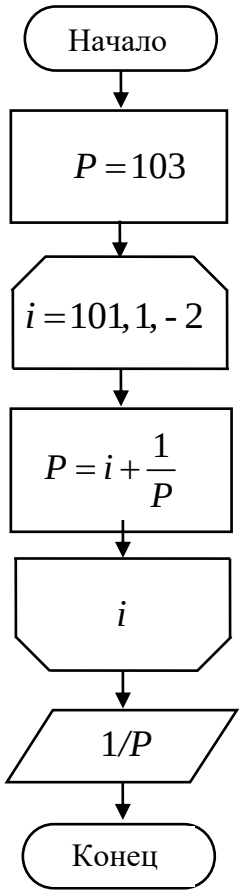
Рисунок 27 – Текст программы на языке C++ для решения задачи 4 с использованием цикла с постусловием

*Задача 5*

Написать программу для вычисления выражения вида:

$$P = \cfrac{1}{1 + \cfrac{1}{3 + \cfrac{1}{5 + \cfrac{1}{\cdots\cdots\cdots\cdots \; 101 + \cfrac{1}{103}}}}}.$$

Блок-схема алгоритма решения задачи с использованием цикла с параметром и текст программы на языке С++ представлены на рисунке 28.




```
#include <iostream>
using namespace std;
int main() {
        float P;
        int i;
        P=103;
        for (i=101; i>0; i-=2) P=i+1/P;
        cout << "P = " << 1/P;
        cin.ignore(200,'\n').get();
        return 0;
}
```

Рисунок 28 – Блок-схема алгоритма решения задачи 5 с использованием цикла с параметром и текст программы на языке С++

Проведем тестирование программы. На рисунке 29 представлен результат работы программы по решению задачи 5.

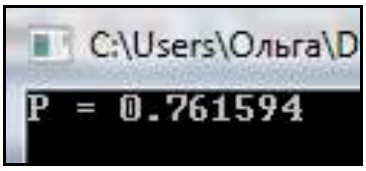



Рисунок 29 – Результат работы программы по решению задачи 5

Для проверки работы программы в пакете Excel организован ввод с клавиатуры начального значения параметра цикла *i*. Текст программ и результаты её работы при разных значениях вводимого параметра представлены на рисунке 30.

```
#include <iostream>
using namespace std;
int main() {
       float P;
       int i;
       cout <<"Input i = ";
       cin >> i;
       P=i+2;
       for (; i>0; i-=2) P=i+1/P;
       cout << "P = " << 1/P;
       cin.ignore(200,'\n').get();
       return 0;
}
```

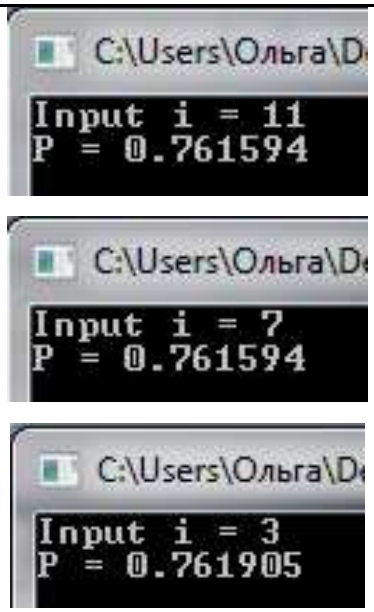



Рисунок 30 – Текст программы по решению задачи 5 и результаты работы программы при разных значениях вводимого параметра

Результаты расчетов в пакете Excel представлены на рисунке 31.

| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | | | 13 | | | 9 | | | 5 |
| 2 | i= | 11 | =B2+1/C1 | i= | 7 | =E2+1/F1 | i= | 3 | =H2+1/I1 |
| 3 | | 9 | =B3+1/C2 | | 5 | =E3+1/F2 | | 1 | =H3+1/I2 |
| 4 | | 7 | =B4+1/C3 | | 3 | =E4+1/F3 | | | |
| 5 | | 5 | =B5+1/C4 | | 1 | =E5+1/F4 | | P= | =1/I3 |
| 6 | | 3 | =B6+1/C5 | | | | | | |
| 7 | | 1 | =B7+1/C6 | | P= | =1/F5 | | | |
| 8 | | | | | | | | | |
| 9 | | P= | =1/C7 | | | | | | |

| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | | | 13 | | | 9 | | | 5 |
| 2 | i= | 11 | 11,07692308 | i= | 7 | 7,111111111 | i= | 3 | 3,2 |
| 3 | | 9 | 9,090277778 | | 5 | 5,140625 | | 1 | 1,3125 |
| 4 | | 7 | 7,110007639 | | 3 | 3,194528875 | | | |
| 5 | | 5 | 5,140646825 | | 1 | 1,313035205 | | P= | 0,761904762 |
| 6 | | 3 | 3,194528049 | | | | | | |
| 7 | | 1 | 1,313035285 | | P= | 0,761594203 | | | |
| 8 | | | | | | | | | |
| 9 | | P= | 0,761594156 | | | | | | |

Рисунок 31 – Результаты решения задачи 5 в пакете Excel (в режимах вывода формул и вывода результатов)

## 3 Задания к лабораторным работам, требования к оформлению и защите отчетов по лабораторным работам

Задания к лабораторным работам по темам «Разработка программ линейной структуры», «Разработка программ с ветвлениями», «Разработка программ с циклами» приведены в приложениях А-В.

Выполнение лабораторных работ по перечисленным темам состоит из следующих этапов:

1) ознакомление с формулировкой задания к лабораторной работе и порядком её выполнения;
2) разработка алгоритмов;
3) написание программы на языке C++;
4) тестирование разработанной программы;
5) оформление результатов;
6) подготовка письменного отчета по лабораторной работе;
7) защита отчета по лабораторной работе.

Отчет по лабораторной работе оформляется в соответствии с требованиями стандарта организации СТО 02069024.101–2015 РАБОТЫ СТУДЕНЧЕСКИЕ. Общие требования и правила оформления [3]. Отчет должен сдержать:

– титульный лист;

– задание к лабораторной работе (формулы набирать в редакторе MS Equetion 3.0);

– блок-схему алгоритма решения задачи (схемы алгоритмов могут быть оформлены с использованием редактора изображений MS Word, MS Visio, графических редакторов, к примеру, редактор Gimp);

– текст программы на языке C++;

– тестовые примеры (минимум два примера). Тестовый пример включает скриншот работы программы и проверку в стандартных пакетах прикладных программ, например, MS Excel или MathCad или расчеты выполненные вручную.

Для защиты отчетов по лабораторным работам необходимо подготовиться к ответу на вопросы, приведенные ниже.

1 Опишите алфавит языка C++.

2 Охарактеризуйте типы данных C++.

3 Что такое переменные?

4 Что такое область действия переменной, область видимости.

5 Как описать переменную в C++.

6 Что такое класс памяти переменной в C++, какие значения он может принимать?

7 Приведите структуру программы C++.

8 Операции C++.

9 Приведите примеры унарных, бинарных, тернарных операций.

10 Выражения в C++, их правила записи.

11 Организация ввода-вывода в стиле C. Правила, примеры.

12 Организация ввода-вывода в стиле C++. Правила, примеры.

13 Операторы ветвления, форма записи, примеры.

14 Оператор выбора, форма записи, примеры.

15 Оператор цикла с предусловием, форма записи, примеры.

16 Оператор цикла с постусловием, форма записи, примеры.

17 Оператор счетного цикла, форма записи, примеры.

18 Операторы передачи управления.

# Список использованных источников

1 Павловская, Т. А. C/C++. Программирование на языке высокого уровня [Текст] : для магистров и бакалавров: учебник для студентов высших учебных заведений, обучающихся по направлению подготовки дипломированных специалистов "Информатика и информационная техника" / Т. А. Павловская. - Санкт-Петербург : Питер, 2008, 2013. - 461 с.

2 Прохоренок, Н.А. Язык С. Самое необходимое / Н.А. Прохоренок. – СПб.: БХВ-Петербург, 2020. – 480 с.

3 СТО 02069024.101–2015 Работы студенческие. Общие требования и правила оформления. – Оренбург: ОГУ, 2015. – Режим доступа: http://www.osu.ru/docs/official/standart/standart_101-2015_.pdf.

4 Яркова, О. Н. Основы алгоритмизации [Электронный ресурс]: методические указания для обучающихся по образовательной программе высшего образования по направлению подготовки 01.03.04 Прикладная математика / О. Н. Яркова, О. С. Чудинова; М-во науки и высш. образования Рос. Федерации, Федер. гос. бюджет. образоват. учреждение высш. образования "Оренбург. гос. ун-т", Каф. мат. методов и моделей в экономике. - Электрон. текстовые дан. (1 файл: 1.36 Мб). - Оренбург : ОГУ, 2019. - 73 с.

5 Панова, Н. Ф. Сборник задач по программированию на языках Pascal, C++ [Электронный ресурс] : метод. указания к лаб. работам по курсу "Программирование" / Н. Ф. Панова; М-во образования и науки Рос. Федерации, Федер. агентство по образованию, Гос. образоват. учреждение высш. проф. образования "Оренбург. гос. ун-т", Каф. приклад. информатики. – Электрон. текстовые дан. (1 файл: 0.33 Мб). - Оренбург : ГОУ ОГУ, 2005. – 52 с.

# Приложение А

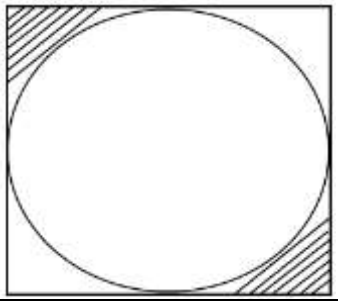
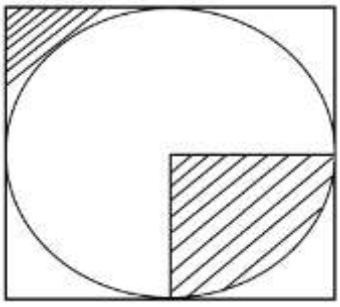
**(*обязательное*)**

**Задание к лабораторной работе на тему «Разработка программ линейной структуры»**

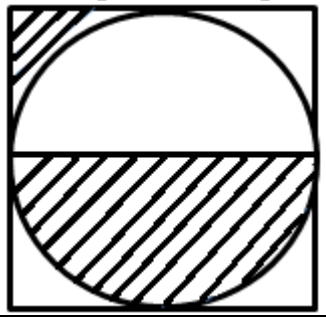
**Задание 1**

На языке C++ разработать программу для решения задачи, соответствующей варианту. Задачи по вариантам приведены в таблице А.1 [4].

Таблица А.1 – Задачи по вариантам на тему «Разработка программ линейной структуры»

| Вариант | Задача |
|---|---|
| (1) | (2) |
| 1 | Значениями переменных A, B, C являются цифры. Присвоить целой переменной L число, составленное из целых цифр (например, A=1, B=3, C=5, тогда L=135). |
| 2 | Дано трехзначное число L. Трем переменным присвоить значения каждой цифры числа L (например, L=135, тогда A=1, B=3, C=5). |
| 3 | Определить площадь заштрихованной части фигуры, если известно длина стороны квадрата.  |
| 4 | В квадрат вписана окружность. Определить площадь заштрихованной фигуры, если известна длина стороны квадрата.  |
| 5 | Целой переменной X присвоить значение суммы цифр заданного трехзначного числа. |

Продолжение таблицы А.1

| (1) | (2) |
|---|---|
| 6 | Дано ребро куба *a*. Вычислить площадь одной грани куба, его объем и площадь полной поверхности. |
| 7 | Дано положительное число a. Вычислить: а) площадь равностороннего треугольника со стороной a; б) площадь квадрата со стороной a; в) площадь круга, радиус которого равен a. |
| 8 | Даны координаты вершин треугольника. Вычислить его площадь. |
| 9 | Пусть идет k-ая секунда суток. Определить, сколько целых часов h и целых минут m прошло к этому моменту. |
| 10 | Присвоить целой переменной L сумму последних цифр целой части положительного вещественного числа X (так, если X=143,57, то L=4+3=7). |
| 11 | Найти произведение P двух первых цифр из дробной части положительного вещественного числа X (так если X=31,956, то P=9*5=45). |
| 12 | Присвоить двум переменным A и B крайнюю левую и крайнюю правую цифры трехзначного числа K, переменной C присвоить сумму цифр числа K (например, если K=135, тогда A=1, B=5, C=1+3+5=9). |
| 13 | Определить число, полученное выписыванием в обратном порядке цифр заданного трехзначного целого числа. |
| 14 | В квадрат вписана окружность. Определить площадь заштрихованной фигуры, если известна длина стороны квадрата.  |
| 15 | Даны катеты *a* и *b* прямоугольного треугольника. Вычислить длину гипотенузы и площадь треугольника. |
| 16 | Целой переменной X присвоить значение произведения цифр заданного четырехзначного числа. |

**Задание 2**

На языке C++ разработать программу для вычисления значений выражений *S*, *S*1, *S*2. Значения параметров *a*, *b*, *c*, *d* вводятся пользователем. Вид выражений по вариантам приведен в таблице А.2.

Таблица А.2 – Вид выражений $S$, $S1$, $S2$ по вариантам

| Вариант | $S$ | $S1$ | $S2$ |
|---|---|---|---|
| (1) | (2) | (3) | (4) |
| 1 | $S = a + b + \frac{c \cdot (b^2 - c)}{4}$ | $S1 = \frac{S \cdot 2 + b^2}{c^2 - 3}$ | $S2 = \frac{(S1 + S \cdot (b - c))}{a^2}$ |
| 2 | $S = \frac{a^2 + b^3 - c}{2 \cdot c \cdot (b - 1)}$ | $S1 = \frac{S \cdot (b + 2)}{(b - 4) \cdot (c + 3)}$ | $S2 = \frac{(S1 + S) \cdot (a - 2)}{(b^2 - 4) \cdot (c^3 + 3)}$ |
| 3 | $S = \frac{(a^2 + d^2) \cdot (b^2 - c)}{(d + 4) \cdot (c - d)}$ | $S1 = \frac{S \cdot (c + d - 3)}{(b - 4) \cdot (d^2 + 2)}$ | $S2 = \frac{(S + S1^2) \cdot (S1^3 - 3)}{(a^2 + b) \cdot (c - d)}$ |
| 4 | $S = \frac{(a^2 - b) \cdot (c - b^2)}{a^2}$ | $S1 = \frac{(S + b) \cdot (b - c)}{(c + a) \cdot (a + b)}$ | $S2 = \frac{(S + S1) \cdot (a - S1)}{(S1 + S^2) \cdot (b + c)}$ |
| 5 | $S = \frac{(b^2 - a \cdot c \cdot (c + 1)) \cdot (a + 2)}{(c - b) \cdot (b + 4)}$ | $S1 = \frac{(S + a \cdot b) \cdot (S + b^2 \cdot a)}{(a^2 - 1) \cdot (b^2 + 4)}$ | $S_2 = \frac{b \cdot c(2 + a)}{S_1}$ |
| 6 | $S = \frac{a + b}{(c + d) \cdot (a + b) + 2}$ | $S_1 = \frac{S + b^2}{2 \cdot c - 4 \cdot d}$ | $S_2 = \frac{S_1 + c^2 - d}{2 \cdot (3c + b) - 5 \cdot d}$ |
| 7 | $S = \frac{(a - b) \cdot 4 + c}{(c - d) \cdot 3 + b^2}$ | $S1 = \frac{S \cdot (a + b) - 4 \cdot d}{3 \cdot (c + d)^2 - 4 \cdot (b + d)}$ | $S2 = \frac{S1 \cdot (b + d) - c}{S1 \cdot (c + d - 2 \cdot b)}$ |
| 8 | $S = \frac{(a + 2 \cdot b) \cdot 4 \cdot c - d)}{25 \cdot (6 \cdot a + b) \cdot (3 \cdot d + c)}$ | $S1 = \frac{s \cdot (a + b) - d}{(6 \cdot b - c) \cdot (4 \cdot d + a)}$ | $S2 = \frac{s1 \cdot b - 4 \cdot b \cdot c \cdot d}{s \cdot s1 \cdot (b + d)^2}$ |
| 9 | $S = \frac{(b + 3 \cdot d) \cdot (4 \cdot a + c)}{2 \cdot a \cdot b \cdot (c + d)}$ | $S1 = S + \frac{2 \cdot b^2 + 4}{(3 \cdot a + b) \cdot (2 \cdot d + 7)}$ | $S2 = S1 \cdot \frac{(b + c) \cdot (4 \cdot d - 3)^2}{2 \cdot d \cdot (c - b) \cdot a + b}$ |
| 10 | $S = \frac{(b + c)^2 \cdot 4}{(2d + c) \cdot (b - 1)}$ | $S1 = S + \frac{(b + d) \cdot a^2}{27 \cdot (b + 4) \cdot d}$ | $S2 = \frac{(b + 2 \cdot a \cdot c)^2 \cdot (4 \cdot d + 1)}{S \cdot S1 \cdot (b + a) \cdot c}$ |
| 11 | $S = \frac{(b + a) \cdot (a^2 - b^2)}{(d + c)^2 - b}$ | $S1 = \frac{(S + d \cdot c) \cdot b}{2 \cdot a \cdot (d + c)}$ | $S2 = \frac{(S1 + b^2)}{S}$ |
| 12 | $S = \frac{(a + b \cdot c) \cdot d - 2 \cdot b^2}{(d + c) \cdot (b^2 + c)}$ | $S1 = \frac{(b - S) \cdot (S + a)}{2 \cdot c \cdot d}$ | $S2 = \frac{(S1 - S) \cdot (a + b^2)}{2 \cdot c \cdot d}$ |
| 13 | $S = \frac{(b + 2 \cdot a \cdot c) \cdot (d + 2 \cdot b \cdot c)}{4 \cdot d^2 - 2 \cdot a \cdot c}$ | $S1 = \frac{2 \cdot S - 4 \cdot b \cdot d}{2 \cdot a \cdot (c + d)^2}$ | $S2 = \frac{S1 - S}{(b^2 + d^2) \cdot a \cdot c}$ |
| 14 | $S = \frac{(a + 2 \cdot b \cdot c) \cdot (2 \cdot d - 4 \cdot a)}{(3 \cdot a^2 - b^3) \cdot (4 \cdot b + c)}$ | $S1 = S \cdot \frac{4 \cdot a + b}{(2 \cdot a \cdot c + 4 \cdot d^2) \cdot 2 \cdot a}$ | $S2 = \frac{(S1 + S) \cdot (a^2 + b^2)}{4 \cdot d^2}$ |
| 15 | $S = \frac{(2 \cdot b^2 - 4 \cdot a \cdot c)}{(b^2 + a^2) \cdot (c^2 - d^2)}$ | $S1 = \frac{S^2 \cdot (b + c)}{(2a^2 - d^2) \cdot c}$ | $S2 = \frac{(S1 - S)^2}{(4 \cdot b^2 - 2 \cdot a \cdot c) \cdot d}$ |
| 16 | $S = \frac{(5b^3 - a \cdot c)}{(b + 2) \cdot (c - b)}$ | $S1 = \frac{(S + S^2) \cdot a}{(2 \cdot a^2 - c) \cdot b}$ | $S2 = \frac{(S1 - S)}{(4 \cdot S1 + b^2) \cdot a^2}$ |

# Приложение Б

**(*обязательное*)**

**Задание к лабораторной работе на тему «Разработка программ с ветвлениями»**

**Задание**

На языке C++ разработать программу для решения задачи, соответствующей варианту. Задачи по вариантам приведены в таблице Б.1 [5].

Таблица Б.1 – Задачи по вариантам на тему «Разработка программ с ветвлениями»

| Вариант | Задача |
|---|---|
| (1) | (2) |
| 1 | Даны вещественные числа x, y, z. Вывести их на экран в порядке убывания. |
| 2 | Даны два значения x, y ($x \neq y$). Меньшее из этих двух чисел заменить их полусуммой, а большее – их удвоенным произведением. |
| 3 | Определить минимальное из четырех чисел: x, y, z, p. |
| 4 | Даны три действительных числа. Выбрать из них те, которые принадлежат интервалу (1; 3). |
| 5 | Определить, является ли значение целочисленной переменной X кратным трём. Если это верно, вывести на экран сообщение «число кратно трём». В противном случае вывести сообщение «число не кратно трём». |
| 6 | Даны положительные числа a, b, c, d. Выяснить, можно ли прямоугольник со сторонами a и b уместить внутри прямоугольника со сторонами c и d так, чтобы каждая из сторон одного прямоугольника была параллельна или перпендикулярна каждой стороне второго прямоугольника. Ответ получить в текстовой форме: «можно» или «нельзя». |
| 7 | Написать программу, в результате выполнения которой булевская переменная T получает значение true, если выполняется условие: введенное число N является удвоенным нечетным. В противном случае переменная принимает значение false. |
| 8 | Написать программу, в результате выполнения которой булевская переменная T получает значение true, если выполняется условие: введенное число N является квадратом нечетного числа. В противном случае переменная принимает значение false. |

Продолжение таблицы Б.1

| (1) | (2) |
|---|---|
| 9 | Вычислить значение функции: $z = \begin{cases} \sin x, \text{если } x < a; \\ \cos x, \text{если } a \le x \le b; \\ tgx, \text{если } x > b. \end{cases}$ |
| 10 | Даны действительные числа A, B, C, D, S, T, U, где $S \neq 0$ и $T \neq 0$. Определить, лежат ли точки (A, B) и (C, D) на прямой, заданной уравнением $Sx + Ty + U = 0$. Если нет, то выяснить, принадлежит ли они разным полуплоскостям. |
| 11 | Даны координаты трех точек на плоскости. Если они могут быть координатами вершин равностороннего треугольника, вычислить его площадь. |
| 12 | Даны координаты вершин четырехугольника. Определить, является ли этот четырехугольник прямоугольным. |
| 13 | Даны числа $A_1, B_1, C_1, A_2, B_2, C_2$. Напечатать координаты точки пересечения прямых: $A_1x + B_1y = C_1$ и $A_2x + B_2y = C_2$, либо сообщить, что эти прямые совпадаю, не пересекаются или вовсе отсутствуют. |
| 14 | Даны действительные числа X, Y. Если X, Y – отрицательные, то каждое значение заменить его модулем. Если отрицательное значение только одно, то оба значения увеличить на 0,5. Если оба значения неотрицательные и ни одно из них не принадлежит отрезку [0.5, 2], то оба значения уменьшить в 10 раз. В остальных случаях X и Y оставить без изменения. |
| 15 | Даны три различных действительных числа X, Y, Z. Если их сумма меньше 1, то наименьшее из них заменить полусуммой двух других чисел, в противном случае заменить меньшее из X, Y полусуммой двух оставшихся значений. |
| 16 | Вычислить значение функции: $z = \begin{cases} \ln x, \text{если } x > 1; \\ 1, \text{если } x = 1; \\ e^x, \text{если } x < 1. \end{cases}$ |

# Приложение В

**(*обязательное*)**

**Задание к лабораторной работе на тему «Разработка программ с циклами**

**Задание 1**

На языке C++ разработать программу для решения задачи, соответствующей варианту. Задачи по вариантам приведены в таблице В.1 [4].

Таблица В.1 – Задача 1 по вариантам на тему «Разработка программ с циклами»

| Вариант | Задача |
|---|---|
| (1) | (2) |
| 1 | Вычислить сумму $1+\frac{x}{2}+\frac{x^2}{3}+K+\frac{x^n}{n}\cdot$ Операцию возведения в степень не использовать. |
| 2 | Вычислить сумму $-1+\frac{1}{2}-\frac{1}{3}+K+\frac{(-1)^n}{n}\cdot$ Операцию возведения в степень и условный оператор не использовать. |
| 3 | Вычислить сумму $1-\frac{3}{2}+\frac{4}{3}-...+\frac{(-1)^{n+1}(n+1)}{n}$. Условный оператор не использовать. |
| 4 | Вычислить сумму $x+\frac{x^3}{3}+\frac{x^5}{5}+...+\frac{x^{11}}{11}$, при x=2. Операцию возведения в степень не использовать. |
| 5 | Вычислить сумму $1-\frac{1}{2}+\frac{1}{3}-...+\frac{(-1)^{n+1}}{n}$. Операцию возведения в степень и условный оператор не использовать. |
| 6 | Дано натуральное число n. Найти сумму $n^2-(n+1)^2+...+(-1)^{n+1}(2n)^2$. Операцию возведения в степень и условный оператор не использовать. |
| 7 | Вычислить сумму $1-\frac{x}{2}+\frac{x^2}{3}-...+(-1)^{n+1}\frac{x^n}{n}$. Условный оператор и операцию возведения в степень не использовать. |
| 8 | Вычислить сумму $\frac{2}{3}x^2+\frac{3}{4}x^3+\frac{4}{5}x^4...+\frac{10}{11}x^{10}$. Операцию возведения в степень не использовать. |

Продолжение таблицы В.1

| (1) | (2) |
|---|---|
| 9 | Последовательность чисел $\nu_0, \nu_1, \nu_2 \text{K}$ образуется по закону: $\nu_1 = \nu_2 = 0$; $\nu_3 = 1{,}5$; $\nu_i = \frac{i+1}{i^2+1} * \nu_{i-1} - \nu_{i-2} * \nu_{i-3}$; $i = 4{,}5{,}\text{K}$. Дано натуральное число $n(n \geq 4)$, получить $\nu_\text{n}$. |
| 10 | Вычислите сумму $1 + \frac{1}{3} + \frac{1}{3^2} + ... + \frac{1}{3^8}$. Операции возведения в степень не использовать. |
| 11 | Рассмотрим последовательность, образованную дробями: $\frac{1}{1}; \frac{2}{1}; \frac{3}{2}, ...$, в которой числитель (знаменатель) следующего члена последовательности получается сложением числителей (знаменателей) двух предыдущих членов. Числители первых двух дробей равны 1 и 2, знаменатели 1 и 1. Найти k-ый член последовательности. |
| 12 | Получить первые n членов последовательности $\frac{1}{1}, -\frac{5}{2}, \frac{7}{3}, ..., \frac{2\text{k}+1}{\text{k}}(-1)^{\text{k}+1}, ...$ |
| 13 | Последовательность чисел $a_0, a_1, a_2, ...$ образуется по закону: $a_0 = 1$, $\text{a}_\text{k} = \text{ka}_{\text{k}-1} + \frac{1}{\text{k}}$, $k = 1{,}2, ...$ Дано натуральное число n. Получить $a_1, a_2, ..., a_n$. |
| 14 | Вычислите сумму $1 + \frac{2}{3} + \frac{4}{3^2} + ... + \frac{16}{3^8}$. Операцию возведения в степень не использовать. |
| 15 | Вычислите сумму $1 + \frac{2 \cdot \text{x}}{3} + \frac{4 \cdot \text{x}^2}{3^2} + ... + \frac{16 \cdot \text{x}^{10}}{3^{10}}$. Операцию возведения в степень не использовать. |
| 16 | Последовательность чисел $\nu_0, \nu_1, \nu_2 \text{K}$ образуется по закону: $\nu_1 = \nu_2 = 0$; $\nu_3 = 2.5$; $\nu_i = \frac{\nu_{i-2}+1}{i^2-3} \cdot \nu_{i-1} - \nu_{i-3}$; $i = 4{,}5{,}\text{K}$. Дано натуральное число $n(n \geq 4)$, получить $\nu_\text{n}$. |

**Задание 2**

На языке С++ разработать программу для решения задачи, соответствующей варианту. Задачи по вариантам приведены в таблице В.2 [4].

Таблица В.2 – Задача 2 по вариантам на тему «Разработка программ с циклами»

| Вариант | Задача |
|---|---|
| (1) | (2) |
| 1 | Даны действительные числа $x, \varepsilon (x \neq 0, \varepsilon > 0)$. Вычислить с точностью до $\varepsilon$ бесконечную сумму и указать количество учтенных слагаемых $\sum_{k=0}^{\infty} \frac{(-1)^k \cdot (k+1) \cdot x^k}{3^k}$. Операцию возведения в степень не использовать. |
| 2 | $a_n = \frac{x^n}{2 \cdot n!}$. Получить $\sum_{n=1}^{k} a_n$, где $k$ - наименьшее число, удовлетворяющее условиям: $(k > 10) \cap (a_{k+1} < 10)$. Операцию возведения в степень не использовать. |
| 3 | Даны числа $x$, $\varepsilon$. Вычислить: $\sum_{k=0}^{\infty} \frac{(-1)^k \cdot x^{2k+1}}{k!}$. Слагаемые, меньшие $\varepsilon$, не учитывать. Операцию возведения в степень не использовать. |
| 4 | Вычислить y=$\sqrt{x}$ по итерационной формуле $y_{i+1} = \frac{1}{2}\left(y_i + \frac{x}{y_i}\right)$ с точностью $\lvert y_{i+1} - y_i \rvert < \varepsilon$. Начальное приближение $y_0 = h$. |
| 5 | Вычислить сумму бесконечного ряда: $\ln(x+1) = x - \frac{x^2}{2} + \frac{x^3}{3} + .... + (-1)^{n-1} \cdot \frac{x^n}{n} + ...., \ -1 \leq x \leq 1,$ с точностью $\varepsilon$. Результат проверить с помощью стандартной функции. Операцию возведения в степень не использовать. |
| 6 | Вычислить с точностью $\varepsilon$: $y = shx = 1 + \frac{x^3}{3!} + \frac{x^5}{5!} + .... + \frac{x^{2n+1}}{(2n+1)!} + ....$ Операцию возведения в степень не использовать. |
| 7 | Вычислить с точностью $\varepsilon$: $y = \cosh x = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + ... + \frac{x^{2n}}{(2n)!} + ....$ Операцию возведения в степень не использовать. |
| 8 | Дано действительное число $x$. Вычислить с точностью $\varepsilon$: $\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - ... + (-1)^n \cdot \frac{x^{2n}}{(2n)!} + ....$ Результат проверить с помощью стандартной функции. Операцию возведения в степень не использовать. |
| 9 | Последовательность $y_0, y_1, \ldots$, образованна по закону: $y_0 = 0$, $y_k = \frac{y_{k-1} + 1}{y_{k-1} + 2}$, k=1,2. Найти первый член $y_k$, для которого выполнится условие $\lvert y_k - y_{k-1} \rvert < \varepsilon$. |

Продолжение таблицы В.2

| (1) | (2) |
|---|---|
| 10 | Даны действительное число $x$, натуральное число $n$. С точностью до члена ряда, меньшего $\varepsilon$ вычислить сумму членов ряда $e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \ldots + \frac{x^n}{n!} + \ldots$. Операцию возведения в степень не использовать. |
| 11 | Дано действительное число $x$. Вычислить с точностью до $10^{-3}$ сумму ряда: $\sum_{k=1}^{\infty} \frac{x^2}{k^{(3/2)}}$, при $\lvert x \rvert < 1$. |
| 12 | Даны положительные натуральные числа $a$, $x$, $\varepsilon$. В последовательности $y_1, y_2, \ldots$, образованной по закону: $y_0 = a$, $y_i = \frac{1}{2}\left(y_{i-1} + \frac{x}{y_{i-1}}\right)$, $i = 1,2,\ldots$ найти первый член $y_n$, для которого выполнено неравенство: $\lvert y_n - y_{n-1} \rvert < \varepsilon$. |
| 13 | Последовательность $a_1, a_2, \ldots$ образована по закону: $a_1 = \left(1 - \frac{1}{2}\right)$, $a_2 = \left(1 - \frac{1}{2}\right)\left(1 - \frac{1}{3}\right)$, $a_n = \left(1 - \frac{1}{2}\right)\left(1 - \frac{1}{3}\right) \cdot \ldots \cdot \left(1 - \frac{1}{n+1}\right)$. Найти первый член $a_n (n \geq 2)$, для которого $\lvert a_n - a_{n-1} \rvert < \varepsilon$. |
| 14 | Пусть: $x_0 = 1$, $x_k = \frac{2 - x_{k-1}^3}{5}$, $k = 1,2,\ldots$. Найти первый член $x_n$, для которого $\lvert x_n - x_{n-1} \rvert < \varepsilon$. |
| 15 | Вычислить с точностью $\varepsilon$: $y = \sin x = 1 - \frac{x^3}{3!} + \frac{x^5}{5!} - \ldots + (-1)^{n+1} \frac{x^{2n-1}}{(2n-1)!} + \ldots$. Результат проверить с помощью стандартной функции $\sin(x)$. |
| 16 | Последовательность вычисляется по закону $a_1 = 1$, $a_2 = \frac{1}{2!}$, … $a_n = \frac{1}{n!}$. Найти первый член $a_n$, для которого выполняется условие: $\lvert a_n - a_{n-1} \rvert < \varepsilon$, $\varepsilon = 0{,}001$. |