ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»  
(РУТ (МИИТ))

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

ОТЧЁТ  
О ПРАКТИЧЕСКОЙ РАБОТЕ №2

По дисциплине «Введение в языки программирования»

Выполнил: ст. гр. ТКИ – 111

Черкас Н.В.

Проверил: к.т.н., доц.

Васильева М.А.

Москва 2021

Оглавление

[Задание 2–1 3](#_Toc85287969)

[Формулировка задания 3](#_Toc85287970)

[Блок-схема алгоритма 4](#_Toc85287971)

[Программа на языке С++ 5](#_Toc85287972)

[Решение тестовых примеров на С++ 9](#_Toc85287973)

[Решение тестовых примеров в Excel 10](#_Toc85287974)

[Задание 2–2 11](#_Toc85287975)

[Формулировка задания 11](#_Toc85287976)

[Блок-схема алгоритма 12](#_Toc85287977)

[Программа на языке С++ 13](#_Toc85287978)

[Решение тестовых примеров на С++ 14](#_Toc85287979)

[Решение тестовых примеров в Excel 15](#_Toc85287980)

[Задание 2–3 16](#_Toc85287982)

[Формулировка задания 16](#_Toc85287983)

[Блок-схема алгоритма 17](#_Toc85287984)

[Программа на языке С++ 18](#_Toc85287985)

[Решение тестовых примеров на С++ 20](#_Toc85287986)

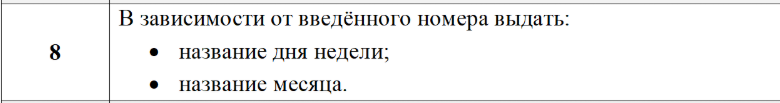
[Решение тестовых примеров в Excel 21](#_Toc85287987)

**Задание 2-1**

**Формулировка задания**

Создать консольное приложение для решения задачи, представленной в таблице, с использованием перечислимого типа. Выбор формулы вычисления зависит от пользователя. Данные для решения задачи так же вводит пользователь. Ввод необходимо проверять на правильность. Все результаты вывести на экран. Дополнить свой отчёт блок-схемой алгоритма.

**Таблица 1**

****

**Блок-схема алгоритма**

Начало

Выбор пользователя (месяц или день)

eNum

day

month

Название дня недели

Название месяца

Конец

Рисунок 1 – Блок-схема алгоритма

**Программа на языке C++**

|  |  |
| --- | --- |
|  | #include <iostream> |
|  | /\*\* |
|  | \* \brief Константы дней недели. |
|  | \*/ |
|  | enum class WEEK |
|  | { |
|  | /\*\* |
|  | \* \brief Понедельник. |
|  | \*/ |
|  | Monday, |
|  |  |
|  | /\*\* |
|  | \* \brief Вторник. |
|  | \*/ |
|  | Tuesday, |
|  |  |
|  | /\*\* |
|  | \* \brief Среда. |
|  | \*/ |
|  | Wednesday, |
|  | /\*\* |
|  | \* \brief Четверг. |
|  | \*/ |
|  | Thursday, |
|  | /\*\* |
|  | \* \brief Пятница. |
|  | \*/ |
|  | Friday, |
|  | /\*\* |
|  | \* \brief Суббота. |
|  | \*/ |
|  | Saturday, |
|  | /\*\* |
|  | \* \brief Воскресенье. |
|  | \*/ |
|  | Sunday |
|  | }; |
|  | /\*\* |
|  | \* \brief Константы месяцов. |
|  | \*/ |
|  | enum class MONTH |
|  | { |
|  | /\*\* |
|  | \* \brief Январь. |
|  | \*/ |
|  | January, |
|  |  |
|  | /\*\* |
|  | \* \brief Февраль. |
|  | \*/ |
|  | February, |
|  |  |
|  | /\*\* |
|  | \* \brief Март. |
|  | \*/ |
|  | March, |
|  | /\*\* |
|  | \* \brief Апрель. |
|  | \*/ |
|  | April, |
|  | /\*\* |
|  | \* \brief Май |
|  | \*/ |
|  | May, |
|  | /\*\* |
|  | \* \brief Июнь. |
|  | \*/ |
|  | June, |
|  | /\*\* |
|  | \* \brief Июль. |
|  | \*/ |
|  | July, |
|  | /\*\* |
|  | \* \brief Август. |
|  | \*/ |
|  | August, |
|  | /\*\* |
|  | \* \brief Сентябрь. |
|  | \*/ |
|  | September, |
|  | /\*\* |
|  | \* \brief Октябрь. |
|  | \*/ |
|  | October, |
|  | /\*\* |
|  | \* \brief Ноябрь. |
|  | \*/ |
|  | November, |
|  | /\*\* |
|  | \* \brief Декабрь. |
|  | \*/ |
|  | December |
|  | }; |
|  |  |
|  | /\*\* |
|  | \* \brief Точка входа в программу. |
|  | \* \return Возвращает 0 в случае успеха. |
|  | \*/ |
|  | int main() |
|  | { |
|  | setlocale(LC\_ALL, "Russian"); |
|  | int a = 0; |
|  | std::cout << "Что вы хотите узнать? \nДень недели - 1\nНазвание месяца - 2\n"; |
|  | std::cin >> a; |
|  | if (a == 1) { |
|  | std::cout << "Введите день недели " |
|  | << "\n1й день - " << static\_cast<int>(WEEK::Monday) |
|  | << "\n2й день - " << static\_cast<int>(WEEK::Tuesday) |
|  | << "\n3й день - " << static\_cast<int>(WEEK::Wednesday) |
|  | << "\n4й день - " << static\_cast<int>(WEEK::Thursday) |
|  | << "\n5й день - " << static\_cast<int>(WEEK::Friday) |
|  | << "\n6й день - " << static\_cast<int>(WEEK::Saturday) |
|  | << "\n7й день - " << static\_cast<int>(WEEK::Sunday)<<std::endl; |
|  |  |
|  | int day; |
|  | std::cin >> day; |
|  |  |
|  |  |
|  | const auto week = static\_cast<WEEK>(day); |
|  | switch (week) |
|  | { |
|  | case WEEK::Monday: |
|  | std::cout << "Понедельник\n"; |
|  | break; |
|  | case WEEK::Tuesday: |
|  | std::cout << "Вторник\n"; |
|  | break; |
|  | case WEEK::Wednesday: |
|  | std::cout << "Среда\n"; |
|  | break; |
|  | case WEEK::Thursday: |
|  | std::cout << "Четверг\n"; |
|  | break; |
|  | case WEEK::Friday: |
|  | std::cout << "Пятница\n"; |
|  | break; |
|  | case WEEK::Saturday: |
|  | std::cout << "Суббота\n"; |
|  | break; |
|  | case WEEK::Sunday: |
|  | std::cout << "Воскресенье\n"; |
|  | break; |
|  | } |
|  | } |
|  | else { |
|  | std::cout << "Введите месяц" |
|  | << "\n1й месяц - " << static\_cast<int>(MONTH::January) |
|  | << "\n2й месяц - " << static\_cast<int>(MONTH::February) |
|  | << "\n3й месяц - " << static\_cast<int>(MONTH::March) |
|  | << "\n4й месяц - " << static\_cast<int>(MONTH::April) |
|  | << "\n5й месяц - " << static\_cast<int>(MONTH::May) |
|  | << "\n6й месяц - " << static\_cast<int>(MONTH::June) |
|  | << "\n7й месяц - " << static\_cast<int>(MONTH::July) |
|  | << "\n8й месяц - " << static\_cast<int>(MONTH::August) |
|  | << "\n9й месяц - " << static\_cast<int>(MONTH::September) |
|  | << "\n10й месяц - " << static\_cast<int>(MONTH::October) |
|  | << "\n11й месяц - " << static\_cast<int>(MONTH::November) |
|  | << "\n12й месяц - " << static\_cast<int>(MONTH::December)<<std::endl; |
|  |  |
|  | int month; |
|  | std::cin >> month; |
|  |  |
|  |  |
|  | const auto Month = static\_cast<MONTH>(month); |
|  | switch (Month) |
|  | { |
|  | case MONTH::January: |
|  | std::cout << "Январь\n"; |
|  | break; |
|  | case MONTH::February: |
|  | std::cout << "Февраль\n"; |
|  | break; |
|  | case MONTH::March: |
|  | std::cout << "Март\n"; |
|  | break; |
|  | case MONTH::April: |
|  | std::cout << "Апрель\n"; |
|  | break; |
|  | case MONTH::May: |
|  | std::cout << "Май\n"; |
|  | break; |
|  | case MONTH::June: |
|  | std::cout << "Июнь\n"; |
|  | break; |
|  | case MONTH::July: |
|  | std::cout << "Июль\n"; |
|  | break; |
|  | case MONTH::August: |
|  | std::cout << "Август\n"; |
|  | break; |
|  | case MONTH::September: |
|  | std::cout << "Сентябрь\n"; |
|  | break; |
|  | case MONTH::October: |
|  | std::cout << "Октябрь\n"; |
|  | break; |
|  | case MONTH::November: |
|  | std::cout << "Ноябрь\n"; |
|  | break; |
|  | case MONTH::December: |
|  | std::cout << "Декабрь\n"; |
|  | break; |
|  | } |
|  | } |
|  | return 0; |
|  | } |

**Решение тестовых примеров на языке C++**

****

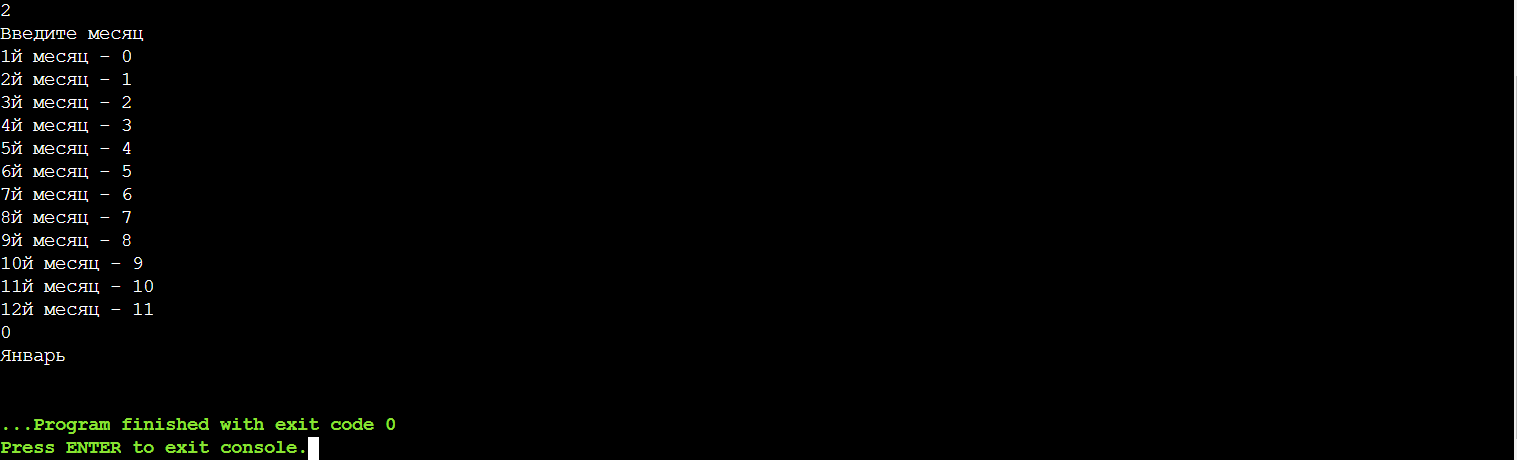
****

Рисунок 2 – Решение тестовых примеров на C++

**Решение тестовых примеров в Excel**

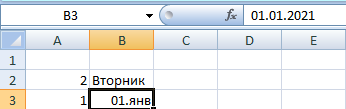
****

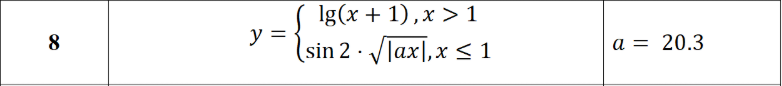
Рисунок 3 – Решение тестовых примеров в Excel

**Задание 2-2**

**Формулировка задания**

Создать консольное приложение с организацией разветвлённого процесса для решения заданной в таблице функции. Данные для решения вводит пользователь. Ввод необходимо проверять на правильность. Все результаты вывести на экран. Дополнить свой отчёт блок-схемой алгоритма.

**Таблица 2**

****

**Блок-схема алгоритма**

y= lg(x+1)

y= sin(2)\*sqrt(a\*x)

x>1

x

a = 20.3

Начало

**True False**

Конец

y

Рисунок 4 – Блок-схема алгоритма

**Программа на языке C++**

|  |  |
| --- | --- |
|  | #include <iostream> |
|  | #include <cmath> |
|  |
|  |
|  |  |
|  | using namespace std; |
|  |  |
|  | /\*\* |
|  | \* \brief вычисление переменной y по заданному x |
|  | \* \param сonst double x - переменная, которую вводит пользователь |
|  | \* \param сonst double y - искомая переменная |
|  | \* \param const double a - переменная, данная по условию |
|  | \* \return - значение y |
|  | \*/ |
|  | double getY(const double x, const double a); |
|  |  |
|  | int main() |
|  | { |
|  | const double a=20.3; |
|  | double x=0; |
|  | double y=0; |
|  | cout<<"Введите значение переменной x"<<endl; |
|  | cin>>x; |
|  | y=getY(x,a); |
|  | cout<<"Y = " << y <<endl; |
|  | return 0; |
|  | } |
|  | double getY(const double x, const double a){ |
|  | if (x>1) |
|  | return log10(x+1); |
|  | else |
|  | return sin(2)\*sqrt(abs(a\*x)); |
|  | } |

**Решение тестовых примеров на языке C++**

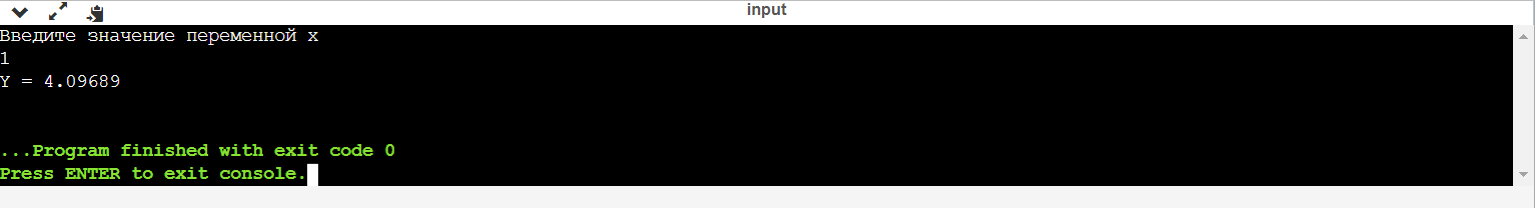
****

Рисунок 5 – Решение тестовых примеров на языке C++

**Решение тестовых примеров в Excel**

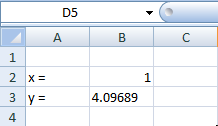
****

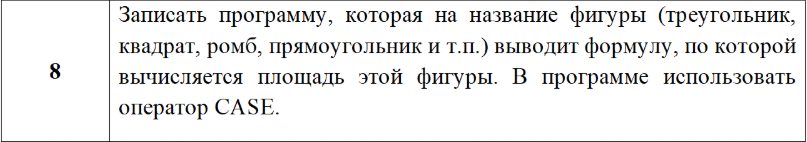
Рисунок 6 - Решение тестовых примеров в Excel

**Задание 2-3**

**Формулировка задания**

Создать консольное приложение с организацией разветвлённого процесса для решения задачи из таблицы. Данные для решения вводит пользователь. Ввод необходимо проверять на правильность. Все результаты вывести на экран. Дополнить свой отчёт блок-схемой алгоритма.

**Таблица 3**

****

**Блок-схема алгоритма**

Выбор фигуры пользователем

eNum, figure

Начало

Формула четырёхугольника

Формула элипса

Формула параллелограмма

Формула Ромба

Формула Трапеция

Рисунок 7 – Блок-схема алгоритма

**Программа на языке C++**

|  |  |
| --- | --- |
|  | #include <iostream> |
|  | /\*\* |
|  | \* \brief Константы фигур. |
|  | \*/ |
|  | enum class FIGURE |
|  | { |
|  | /\*\* |
|  | \* \brief Параллелограмм. |
|  | \*/ |
|  | Parallelogram, |
|  |  |
|  | /\*\* |
|  | \* \brief Ромб. |
|  | \*/ |
|  | Rhombus, |
|  |  |
|  | /\*\* |
|  | \* \brief Трапеция. |
|  | \*/ |
|  | Trapezoid, |
|  | /\*\* |
|  | \* \brief Четырёхугольник. |
|  | \*/ |
|  | Quadrilateral, |
|  | /\*\* |
|  | \* \brief Круг. |
|  | \*/ |
|  | Circle, |
|  | /\*\* |
|  | \* \brief Эллипс. |
|  | \*/ |
|  | Ellipse, |
|  | }; |
|  |  |
|  |  |
|  | /\*\* |
|  | \* \brief Точка входа в программу. |
|  | \* \return Возвращает 0 в случае успеха. |
|  | \*/ |
|  | int main() |
|  | { |
|  | setlocale(LC\_ALL, "Russian"); |
|  | std::cout << "Площадь какой фигуры вы хотите узнать? " |
|  | << "\nПараллелограмм - " << static\_cast<int>(FIGURE::Parallelogram) |
|  | << "\nРомб - "<< static\_cast<int>(FIGURE::Rhombus) |
|  | << "\nТрапеция - "<< static\_cast<int>(FIGURE::Trapezoid) |
|  | << "\nЧетырёхугольник - "<< static\_cast<int>(FIGURE::Quadrilateral) |
|  | << "\nКруг - "<< static\_cast<int>(FIGURE::Circle) |
|  | << "\nЭллипс - "<< static\_cast<int>(FIGURE::Ellipse)<<std::endl; |
|  |  |
|  | int figure; |
|  | std::cin >> figure; |
|  |  |
|  |  |
|  | const auto Figure = static\_cast<FIGURE>(figure); |
|  | switch (Figure) |
|  | { |
|  | case FIGURE::Parallelogram: |
|  | std::cout << "Параллелограмм: S = a · h\n S = a · b · sin α\nS= (d1 · d2 · sin y)/2\n"; |
|  | break; |
|  | case FIGURE::Rhombus: |
|  | std::cout << "Ромб: S = a · h\n S = a^2 · sin α'\n(d1 · d2)/2\n"; |
|  | break; |
|  | case FIGURE::Trapezoid: |
|  | std::cout << "Трапеция: S= sqrt( p\*(p-a) \* (p-b) \* (p-c) )\np=( a+b+c )/ 2\nS= ( h\* ( a+b ) )/ 2\n"; |
|  | break; |
|  | case FIGURE::Quadrilateral: |
|  | std::cout << "Четырехугольник: S = √(p - a)(p - b)(p - c)(p - d) - abcd cos2θ\nS = p · r\nS= (d1 · d2 · sin y)/2\n"; |
|  | break; |
|  | case FIGURE::Circle: |
|  | std::cout << "Круг: S = π \* r ^ 2\nS= (π \* d ^ 2)/4"; |
|  | break; |
|  | case FIGURE::Ellipse: |
|  | std::cout << "Элипс: S = π · a · b\n"; |
|  | break; |
|  | } |
|  | return 0; |
|  | } |

**Решение тестовых примеров на C++**

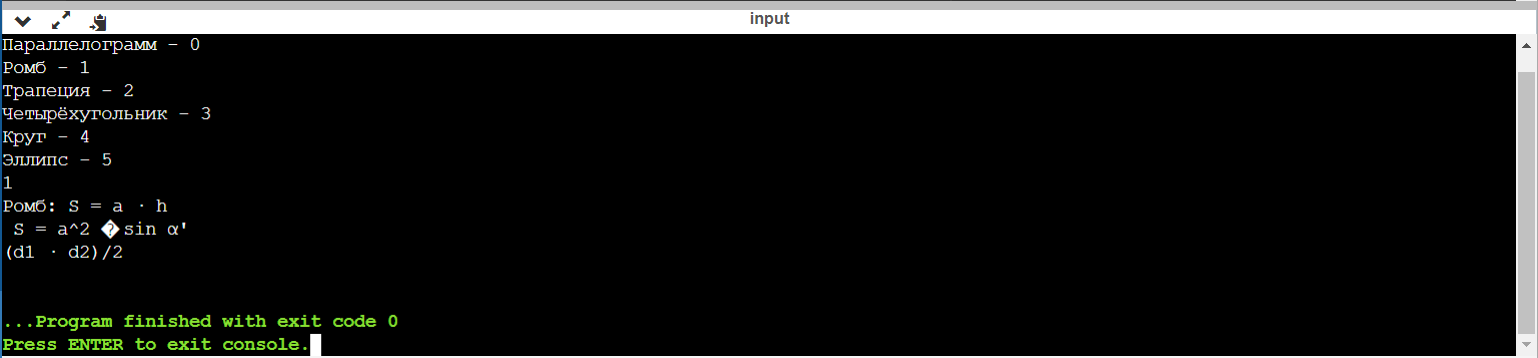
****

Рисунок 8 – Решение тестовых примеров на C++

**Решение тестовых примеров в Excel**

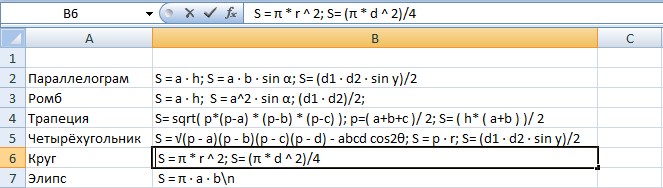
****

Рисунок 9 – Решение тестовых примеров в Excel