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 ЛАБОРАТОРНОЙ РАБОТЕ №2

По дисциплине «Языки программирования»

Вариант 9

Выполнил: ст. гр. ТКИ - 141

Сивунов Пётр Сергеевич

Проверил: к.т.н., доц. Васильева М. А.

(Проверил: к.т.н, доц. Балакина Е. П.)

Москва 2024

2.1.1 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 – Исходные данные

|  |  |
| --- | --- |
| Вариант | Формулы |
| 9 | Заданы длина, ширина и высота параллелепипеда. Определить его:   * объём; * площадь поверхности. |

2.1.2 Блок-схема алгоритма

Блок-схема основного алгоритма представлена ниже (Рисунок 1). Блок-схемы функций представлены ниже (Рисунок 2).

Изображение выглядит как текст, снимок экрана, черно-белый, книга

Автоматически созданное описание

Рисунок 1 ­ Блок-схема основного алгоритма

Изображение выглядит как текст, диаграмма, линия, Шрифт

Автоматически созданное описаниеИзображение выглядит как текст, диаграмма, снимок экрана, круг

Автоматически созданное описание

Рисунок 2 – Блок-схема используемых функций

2.1.3 Текст программы на языке C

#include <locale.h>

#include <stdio.h>

#include <stdlib.h>

/\*\*

\* Перечисление выборов пользователя

\* VOLUME - объем

\* SURFACE\_AREA - площадь поверхности

\* EXIT - выход

\*/

typedef enum

{

VOLUME = 1,

SURFACE\_AREA = 2,

EXIT = 3

} UserChoice;

/\*\*

\* @brief Вычисляет объем параллелепипеда

\* @param length длина

\* @param width ширина

\* @param height высота

\* @return double - Объем параллелепипеда

\*/

double get\_volume(const double length, const double width, const double height);

/\*\*

\* @brief Вычисляет площадь поверхности параллелепипеда

\* @param length длина

\* @param width ширина

\* @param height высота

\* @return double - Площадь поверхности параллелепипеда

\*/

double get\_square(const double length, const double width, const double height);

/\*\*

\* @ Функция ввода вещестенного числа

\* @return double

\*/

double double\_input(void);

/\*\*

\* Пытается получить выбор пользователя

\* @return UserChoice

\*/

int get\_user\_choice(void);

/\*\*

\* @brief Точка входа в программу

\* @return 0

\*/

int main(int argc, char\* argv[])

{

setlocale(LC\_ALL, "Rus");

printf("%d. Объем\n", VOLUME);

printf("%d. Площадь поверхности\n", SURFACE\_AREA);

printf("%d. Выход\n", EXIT);

puts("Введите длину");

double length = double\_input();

puts("Введите ширину");

double width = double\_input();

puts("Введите высоту");

double height = double\_input();

UserChoice choice = get\_user\_choice();

switch (choice)

{

case VOLUME:

{

double volume = get\_volume(length, width, height);

printf("Объем: %lf\n", volume);

break;

}

case SURFACE\_AREA:

{

double square = get\_square(length, width, height);

printf("Площадь поверхности: %lf\n", square);

break;

}

case EXIT:

{

puts("Завершение работы");

return 0;

}

default:

{

puts("Неверный ввод. Попробуйте снова.");

abort();

}

}

return 0;

}

int get\_choice(void)

{

UserChoice choice = EXIT;

printf("Введите %d (Объем) или %d (Площадь поверхности) или %d (Выход):", VOLUME, SURFACE\_AREA, EXIT);

if (scanf\_s("%d", &choice) != 1)

{

abort();

}

return choice;

}

double double\_input(void)

{

double value = 0;

if (scanf\_s("%lf", &value) != 1)

{

abort();

}

return value;

}

double get\_square(const double length, const double width, const double height)

{

return 2 \* (length \* width + width \* height + length \* height);

}

double get\_volume(const double length, const double width, const double height)

{

return length \* width \* height;

}

2.1.4 Результаты выполнения программы

Результаты выполнения программы представлены ниже (Рисунок 3).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Рисунок 3 – Результаты выполнения программы

2.1.5 Выполнение тестовых примеров

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 4 – Тест 1

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 5 – Тест 2

Изображение выглядит как текст, снимок экрана, дисплей, Шрифт

Автоматически созданное описаниеРисунок 6 – Тест 3

* + 1. 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 1 – Исходные данные

|  |  |  |
| --- | --- | --- |
| Вариант | Функции | Константы |
| 9 |  |  |

2.2.2 БЛОК-СХЕМА АЛГОРИТМА

Изображение выглядит как диаграмма, текст, Технический чертеж, План

Автоматически созданное описание

Рисунок 7 – Блок схема функции main

Изображение выглядит как текст, диаграмма, снимок экрана, зарисовка

Автоматически созданное описание

Рисунок 8 - Блок схема функций

2.2.3 ТЕКСТ ПРОГРАММЫ НА ЯЗЫКЕ C

#include <locale.h>

#include <math.h>

#include <stdio.h>

#include <stdlib.h>

/\*\*

\* @brief Вычисляет значение функции при x > 1

\* @param a константа а

\* @param x переменная x

\* @return Возвращает значение y

\*/

double get\_y1(const double a, const double x);

/\*\*

\* @brief Вычисляет значение функции при x > 1

\* @param a константа а

\* @param x переменная x

\* @return Возвращает значение y

\*/

double get\_y2(const double a, const double x);

/\*\*

\* @ Функция ввода вещестенного числа

\* @return double

\*/

double double\_input(void);

/\*\*

\* @brief Точка входа в программу

\* @return 0

\*/

int main(int argc, char\* argv[])

{

const double a = 0.9;

setlocale(LC\_ALL, "Rus");

puts("Введите значение x:");

double x = double\_input();

if (x > 1)

{

printf("Значение y: %lf\n", get\_y1(a, x));

}

else

{

printf("Значение y: %lf\n", get\_y2(a, x));

}

return 0;

}

double get\_y1(double a, double x)

{

return a \* log10(x) + sqrt(fabs(x));

}

double get\_y2(double a, double x)

{

return 2 \* a \* cos(x) + 3 \* pow(x, 3);

}

double double\_input()

{

double value = 0;

if (scanf\_s("%lf", &value) != 1)

{

abort();

}

return value;

}

2.2.4 РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ПРОГРАММЫ

Результаты выполнения программы представлены ниже.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 9 – Результаты выполнения программы

2.2.5 ВЫПОЛНЕНИЕ ТЕСТОВЫХ ПРИМЕРОВ

Изображение выглядит как текст, снимок экрана, дисплей, Шрифт

Автоматически созданное описание

Рисунок 10 – Тест 1

Изображение выглядит как текст, снимок экрана, Шрифт, дисплей

Автоматически созданное описание

Рисунок 11 – Тест 2

* + 1. 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 – Исходные данные

|  |  |
| --- | --- |
| Вариант | Задачи |
| 9 | Записать программу, которая на ввод времени суток выводит соответствующее пожелание доброго утра, доброго дня, доброго вечера и спокойной ночи. |

2.3.2 БЛОК-СХЕМА АЛГОРИТМА

Изображение выглядит как текст, снимок экрана, графический дизайн, Шрифт

Автоматически созданное описание

Рисунок 12 – Блок схема main

Изображение выглядит как текст, снимок экрана, черно-белый, дизайн

Автоматически созданное описание

Рисунок 13 – Блок схема determine\_time\_of\_day

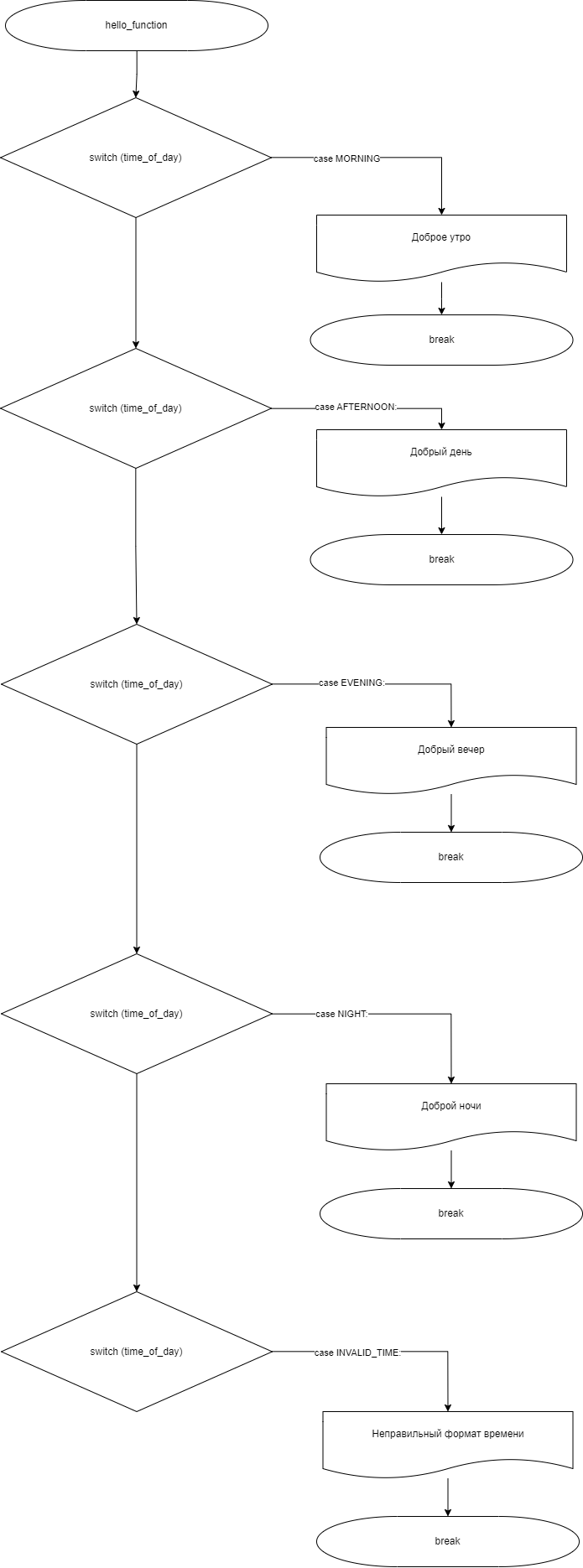


Рисунок 14 – Блок схема hello\_function

2.3.3 ТЕКСТ ПРОГРАММЫ НА ЯЗЫКЕ C

#include <locale.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

/\*\*

\* @brief Перечисление для времени суток

\*/

typedef enum {

MORNING,

AFTERNOON,

EVENING,

NIGHT,

INVALID\_TIME

} TimeOfDay;

/\*\*

\* @brief Определяет время суток на основе введенного времени

\* @param input\_time Структура tm с заполненными параметрами часы и минуты

\* @return Значение типа TimeOfDay (утро, день, вечер, ночь)

\*/

TimeOfDay determine\_time\_of\_day(const struct tm\* input\_time);

/\*\*

\* @brief Выводит приветствие соответствующее времени суток

\* @param time\_of\_day Время суток, определённое функцией determine\_time\_of\_day

\*/

void hello\_function(TimeOfDay time\_of\_day);

/\*\*

\* @brief Точка входа в программу

\* @return 0

\*/

int main(int argc, char\* argv[])

{

setlocale(LC\_ALL, "Rus");

struct tm time\_info = {0};

int is\_running = 1;

char input[10];

while (is\_running)

{

printf("Введите время в формате часы:минуты (или 'q' для выхода): ");

scanf\_s("%s", input, (unsigned)\_countof(input));

if (strcmp(input, "q") == 0)

{

is\_running = 0;

puts("Выход из программы...");

continue;

}

if (sscanf\_s(input, "%d:%d", &time\_info.tm\_hour, &time\_info.tm\_min) != 2)

{

puts("Неправильный формат ввода.");

continue;

}

TimeOfDay time\_of\_day = determine\_time\_of\_day(&time\_info);

hello\_function(time\_of\_day);

}

return 0;

}

/\*\*

\* @brief Определяет время суток на основе введённого времени

\* @param input\_time Структура tm с заполненными параметрами часы и минуты

\* @return Время суток (утро, день, вечер, ночь)

\*/

TimeOfDay determine\_time\_of\_day(const struct tm\* input\_time)

{

if (input\_time->tm\_hour > 23 || input\_time->tm\_min > 59)

{

return INVALID\_TIME;

}

if (input\_time->tm\_hour >= 6 && input\_time->tm\_hour <= 12)

{

return MORNING;

}

if (input\_time->tm\_hour > 12 && input\_time->tm\_hour < 17)

{

return AFTERNOON;

}

if (input\_time->tm\_hour >= 17 && input\_time->tm\_hour < 23)

{

return EVENING;

}

return NIGHT;

}

/\*\*

\* @brief Выводит приветствие на основе времени суток

\* @param time\_of\_day Время суток

\*/

void hello\_function(TimeOfDay time\_of\_day)

{

switch (time\_of\_day)

{

case MORNING:

puts("Доброе утро");

break;

case AFTERNOON:

puts("Добрый день");

break;

case EVENING:

puts("Добрый вечер");

break;

case NIGHT:

puts("Доброй ночи");

break;

case INVALID\_TIME:

default:

puts("Неправильный формат времени.");

break;

}

}

* + 1. РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ПРОГРАММЫ

Результаты выполнения программы представлены ниже.

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 12 – Результаты выполнения программы

**Approve скриншоты**

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, снимок экрана

Автоматически созданное описание