	<p align="center"> Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана) </p>
---	--

ФАКУЛЬТЕТ

Информатика и системы управления

КАФЕДРА

Программное обеспечение ЭВМ и информационные технологии

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

ПО ДИСЦИПЛИНЕ

«ОСНОВЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ»

Студент

Сергеев Михаил Алексеевич

Группа

ИУ7-58Б(В)

Студент

подпись, дата

Сергеев М.А.

фамилия, и.о.

Преподаватель

подпись, дата

Солодовников В.И.

фамилия, и.о.

Оценка _____

2021 г.

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Написать программу для посекундного расчёта текущей скорости, высоты и текущей кинетической энергии падающего тела. У пользователя должна быть возможность ввода массы тела и начальной высоты.

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Необходимо разработать приложение, позволяющее выполнять расчёты координаты Y и текущей кинетической энергии свободно падающего тела (без учёта сопротивления воздуха). Начальным значением вертикальной координаты ($Y=0$) считается уровень поверхности, на которую падает тело. Программа должна запрашивать у пользователя значение начальной высоты (высоты падения тела) в метрах, а также массу падающего тела (в граммах). Для каждой секунды падения тела программа должна выводить значение высоты, на которой оно в данный момент находится, а также текущую кинетическую энергию.

Программа должна предотвращать ввод не положительных значений массы тела и начальной высоты, повторно запрашивая в таком случае ввод данных значений. Использование дробных единиц высоты (метров) и массы (граммов) не допускается.

Значение величины ускорение свободного падения необходимо считать равным $9,8 \text{ м/с}^2$.

ИСХОДНЫЙ КОД ПРОГРАММЫ

Программа компилируется из двух файлов – заголовочного файла constants.h, содержащего значение величины ускорения свободного падения g (что позволяет при необходимости изменить его без редактирования исходного файла программы), а также исходного файла Practice.cpp, содержащего основной код консольного приложения.

Содержимое обоих файлов представлено ниже.

constants.h:

```
#ifndef CONSTANTS_H
#define CONSTANTS_H

// определяется пространство имён с именем myConstants
namespace myConstants
{
    // содержащее константу g (ускорение свободного падения)
    const double g (9.8);
}
#endif
```

Для повышения удобства чтения содержимое файла Practice.cpp представлено, начиная со следующей страницы.

В начале файла подключаются заголовочные файлы iostream (для организации ввода-вывода), constants.h (содержащий значение физической константы – величины ускорения свободного падения), а также math.h (он необходим для использования функции возведения в степень `pow`).

Исходный файл Practice.cpp содержит следующие функции:

`funGetHeightValue` – запрашивает у пользователя высоту h_0 , с которой падает тело

`funGetMassValue` – запрашивает у пользователя массу падающего тела m

`funCurrentHeight` – рассчитывает текущую высоту тела в зависимости от времени падения, используя формулу $h=h_0 - (gt^2)/2$

`funCurrentSpeed` – рассчитывает текущую скорость падения тела v в зависимости от времени падения, используя формулу $v=gt$

`funCurrentEnergy` – рассчитывает текущую кинетическую энергию тела в зависимости от текущей скорости, используя формулу $E=(mv^2)/2$

```

#include <iostream>
#include "constants.h"
#include <math.h>

// запрашиваем высоту
int funGetHeightValue()
{
    int h;
    std::cout << "What is the height of the tower (in
meters)?" << std::endl;
    std::cin >> h;
    return h;
}

// запрашиваем массу тела
int funGetMassValue()
{
    int m;
    std::cout << "What is the mass of the ball (in grams)?"
<< std::endl;
    std::cin >> m;
    return m;
}

// функция расчёта текущей высоты (в зависимости от времени
падения)
double funCurrentHeight(double h, int t)
{
    return h - ((myConstants::g * pow(t, 2)) / 2);
}

// функция расчёта текущей скорости (в зависимости от времени
падения)
double funCurrentSpeed(int t)
{
    return myConstants::g * t;
}

// функция расчёта текущей кинетической энергии (в зависимости
от времени падения)
double funCurrentEnergy(int m, double v)
{
    return (m * pow(v, 2)) / 2;
}

```

```

int main()
{
    // t - время, m - масса, h - высота
    int t{ 0 }, m{ 0 }, h{ 0 };

    // запрашиваем значение высоты
    h = funGetHeightValue();
    // проверка ввода на положительное значение
    while (h <= 0)
    {
        // просьба повторного ввода значения высоты
        std::cout << "I'm sorry, but I can't understand how
high the tower is. Would you be please so kind to enter it
again?" << std::endl;
        h = funGetHeightValue();
    }

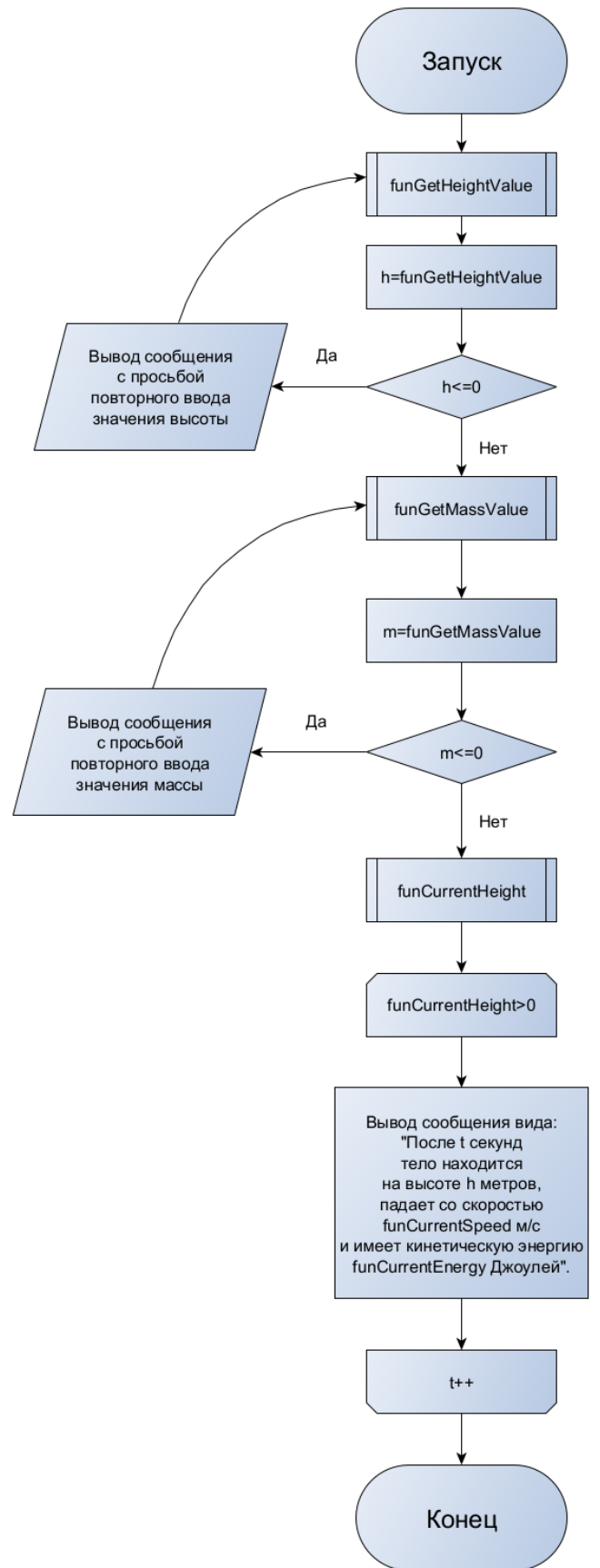
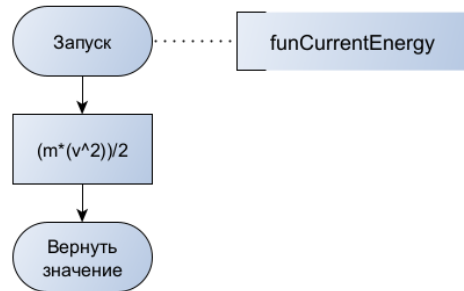
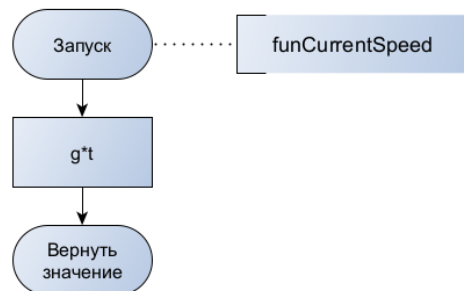
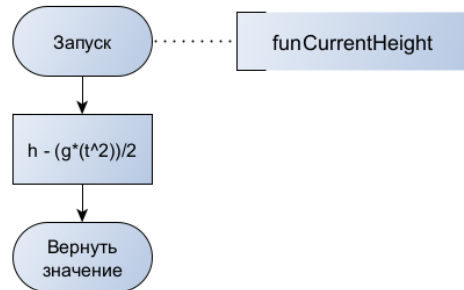
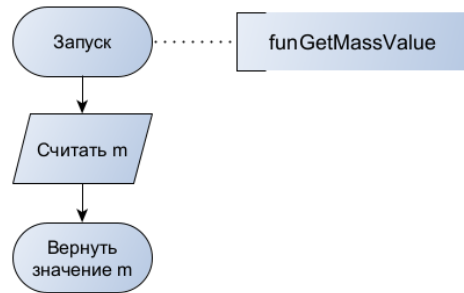
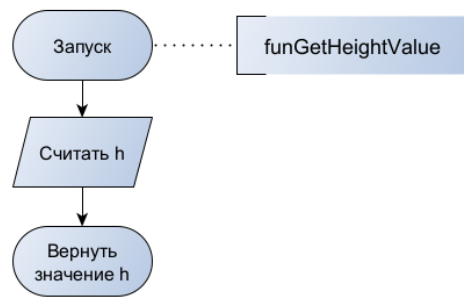
    // запрашиваем значение массы тела
    m = funGetMassValue();
    // проверка ввода на положительное значение
    while (m <= 0)
    {
        // просьба повторного ввода значения массы
        std::cout << "I'm sorry, but I can't understand how
heavy the ball is. Would you be please so kind to enter it
again?" << std::endl;
        m = funGetMassValue();
    }

    // посекундный вывод параметров, пока текущая высота
    больше нуля (т.е. пока тело не упало на землю)
    for (h, t; funCurrentHeight(h, t) > 0; t++)
    {
        std::cout << "In " << t << " second(s), the ball will
be at a height of " << funCurrentHeight(h, t) << " meter(s)
with the speed of " << funCurrentSpeed(t) << " m/s and the
energy of " << funCurrentEnergy(m, funCurrentSpeed(t)) << "
Joule(s)" << std::endl;
    }

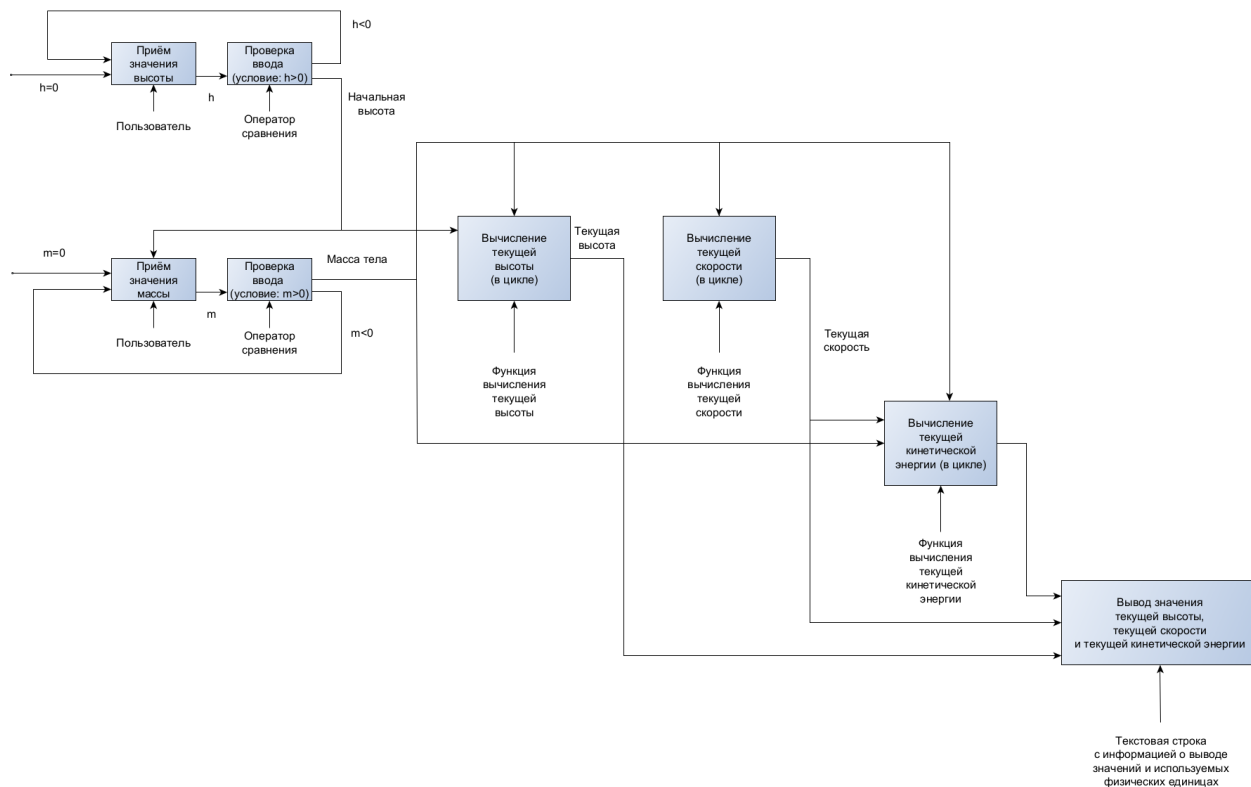
    return 0;
}

```

БЛОК-СХЕМА ПРОГРАММЫ



SADT-ДИАГРАММА



Замечание: для более удобного чтения блок-схема и SADT-диаграмма представлены в виде отдельных файлов, содержащихся в приложении к настоящей лабораторной работе.

ПРИМЕР ТЕСТОВОГО ЗАПУСКА ПРИЛОЖЕНИЯ

Ниже представлен пример запуска программы. Наглядно видно, что приложение действительно проверяет корректность вводимых данных исходной высоты и массы тела и в случае, если они не являются положительными, вежливо просит ввести их повторно, что, в соответствии с правилами проектирования устойчивого пользовательского интерфейса, повышает удобство использования приложения. Получив корректные значения, программа посекундно выводит значение текущей координаты тела по вертикали, текущей скорости и текущей кинетической энергии, что соответствует требованиям, содержащимся в техническом задании к программе.

```
Какова высота падения (в метрах)?
-100
Извините, но я не могу понять, с какой высотой падает тело. Не могли бы Вы ввести её повторно?
Какова высота падения (в метрах)?
0
Извините, но я не могу понять, с какой высотой падает тело. Не могли бы Вы ввести её повторно?
Какова высота падения (в метрах)?
100
Какова масса тела (в граммах)?
-50
Извините, но я не могу понять, какова масса тела. Не могли бы Вы ввести её повторно?
Какова масса тела (в граммах)?
0
Извините, но я не могу понять, какова масса тела. Не могли бы Вы ввести её повторно?
Какова масса тела (в граммах)?
50
Через 0 секунд(у/ы) тело будет на высоте 100 метр(а/ов), падая со скоростью 0 м/с и кинетической энергией в 0 Джоулей
Через 1 секунд(у/ы) тело будет на высоте 95.1 метр(а/ов), падая со скоростью 9.8 м/с и кинетической энергией в 2401 Джоулей
Через 2 секунд(у/ы) тело будет на высоте 80.4 метр(а/ов), падая со скоростью 19.6 м/с и кинетической энергией в 9604 Джоулей
Через 3 секунд(у/ы) тело будет на высоте 55.9 метр(а/ов), падая со скоростью 29.4 м/с и кинетической энергией в 21609 Джоулей
Через 4 секунд(у/ы) тело будет на высоте 31.6 метр(а/ов), падая со скоростью 39.2 м/с и кинетической энергией в 38416 Джоулей
```