МИНИСТЕРСТВО ТРАНСПОРТА РФ

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

Кафедра АСУ

**Отчет**

По лабораторной работе

по дисциплине   
«Языки программирования высокого уровня»

Тема: «МЕХАНИЗМ ОБРАБОТКИ ИСКЛЮЧЕНИЙ»

Задание № 7

Вариант № 4

Выполнил:   
 ст. гр. УВА-111:

Букин А. А.

Проверил:

Савушкин С.А.

Москва 2020 г.

**Содержание:**

[**Постановка задачи 3**](#_heading=h.30j0zll)

[**Общие требования 3**](#_heading=h.1fob9te)

[**Задание 3**](#_heading=h.3znysh7)

[**Авторское понимание и формализация задачи**](#_heading=h.2et92p0) **3**

[**Текст программы**](#_heading=h.tyjcwt) **4**

[**Результаты тестирования**](#_heading=h.3dy6vkm) **7**

[**Вывод**](#_heading=h.1t3h5sf) **7**

**Постановка задачи**

**Общие требования**

Во всех заданиях реализуемые функции должны генерировать подходящие исключения. Обработку исключений нужно выполнять главной функцией, которая должна демонстрировать обработку всех перехватываемых исключений.

Функции, реализуемые в заданиях, обязаны выполнять проверку передаваемых параметров и генерировать исключения в случае ошибочных. Все функции реализуются в четырех вариантах:

- без спецификации исключений;

- со спецификацией throw();

- с конкретной спецификацией с подходящим стандартным исключением;

- спецификация с собственным реализованным исключением.

Собственное исключение должно быть реализовано в трех вариантах: как пустой класс, как независимый класс с полями-параметрами функции, как наследник от стандартного исключения с полями.

**Задание**

Функция переводит часы и минуты в секунды.

**Авторское понимание и формализация задачи**

Мы создаем структуру MyTime для работы с ней. Реализуем методы вычисления секунд и демонстрируем работу обработки исключений. Получившийся результат выводим на экран.

**Текст программы**

**Файл – lab7.cpp**

**#include <iostream>**

**#include <string>**

**using namespace std;**

**// Структура, описываюшщая время**

**struct MyTime {**

**int hours;**

**int mins;**

**MyTime(int phours, int pmins) :**

**hours(phours), mins(pmins) {}**

**};**

**// собственный класс исключение**

**class MyTimeException {**

**int hours;**

**int mins;**

**public:**

**string message;**

**MyTimeException(int phours, int pmins, const string& s) :**

**hours(phours), mins(pmins), message(s) {}**

**};**

**class MyTimeException2 : public invalid\_argument {**

**int hours;**

**int mins;**

**public:**

**string message;**

**MyTimeException2(int phours, int pmins, const string& s) :**

**hours(phours), mins(pmins), invalid\_argument(s) {}**

**};**

**int TimeSfirst(MyTime Time) {**

**// не проверяются параметры**

**return (Time.hours \* Time.mins \* 60);**

**}**

**int TimeSsecond(MyTime Time) {**

**// проверяются параметры и в случае их несоответствия генерируются исключения**

**if (Time.hours < 0) throw 1;**

**if (Time.mins < 0) throw 2;**

**return (Time.hours \* Time.mins \* 60);**

**}**

**int TimeSthird(MyTime Time) {**

**// проверяются параметры и в случае их несоответствия генерируются исключения**

**if (Time.hours < 0) throw MyTimeException(Time.hours, Time.mins, "hours < 0");**

**if (Time.mins < 0) throw MyTimeException(Time.hours, Time.mins, "mins < 0");**

**return (Time.hours \* Time.mins \* 60);**

**}**

**int TimeSfour(MyTime Time) {**

**// проверяются параметры и в случае их несоответствия генерируются исключения**

**if (Time.hours < 0) throw MyTimeException2(Time.hours, Time.mins, "hours < 0");**

**if (Time.mins < 0) throw MyTimeException2(Time.hours, Time.mins, "mins < 0");**

**return (Time.hours \* Time.mins \* 60);**

**}**

**//Главная функция**

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

**{**

**int hours = 0, mins = 0;**

**cout << "Введите часы: ";**

**cin >> hours;**

**cout << "Введите минуты: ";**

**cin >> mins;**

**MyTime mytime(hours, mins);**

**// обработка исключения**

**try {**

**int sec = TimeSfirst(mytime);**

**cout << sec << endl;**

**}**

**catch (int) {**

**cout << "Error" << endl;**

**}**

**try {**

**int sec = TimeSsecond(mytime);**

**cout << sec << endl;**

**}**

**catch (int) {**

**cout << "Error" << endl;**

**}**

**try {**

**int sec = TimeSthird(mytime);**

**cout << sec << endl;**

**}**

**catch (MyTimeException e) {**

**cout << e.message.c\_str() << endl;**

**}**

**try {**

**int sec = TimeSfour(mytime);**

**cout << sec << endl;**

**}**

**catch (MyTimeException2 e) {**

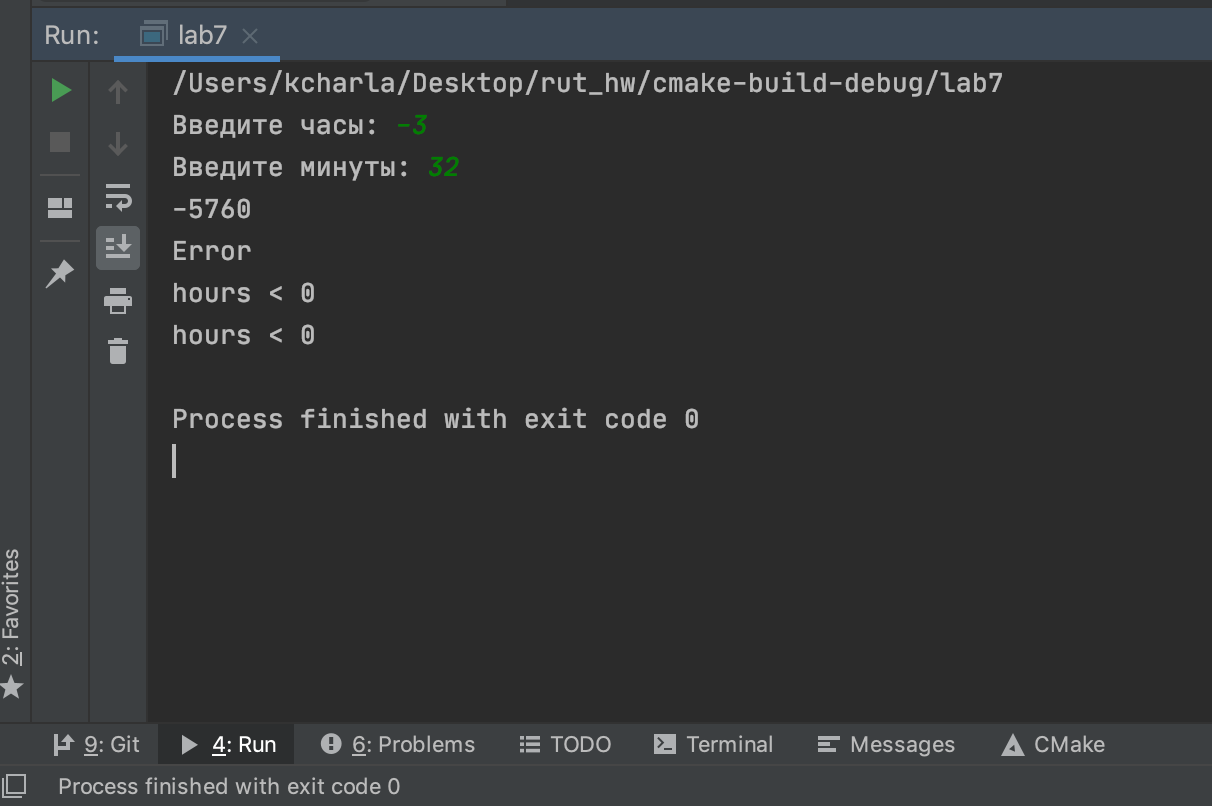
**cout << e.what() << endl;**

**}**

**return 0;**

**}**

**Результаты тестирования**

****

**Вывод**

Программа выдает ожидаемые результаты и работает корректно. Поскольку полученные результаты совпали с ожидаемыми, программу можно считать правильной. Я научился обрабатывать исключительные ситуации.