|  |  |
| --- | --- |
|  | **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)** |

|  |  |
| --- | --- |
| ФАКУЛЬТЕТ | «Информатика и системы управления» (ИУ) |

|  |  |
| --- | --- |
| КАФЕДРА | «Информационная безопасность» (ИУ8) |

Лабораторная работа № 6

ПО КУРСУ

«Алгоритмические языки»

на тему «Исключения»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИУ8-23 |  |  |  | В. С. Ажгирей |
|  | (Группа) |  |  |  | (И. О. Фамилия) |
|  |  |  |  |  |  |
| Преподаватель: |  |  |  |  | М. В. Малахов |
|  |  |  |  |  | (И.О. Фамилия) |

2024

Введение

Цели и задачи работы

Целью задачи является изучение работы с исключениями и обработчиками исключений в программировании на C++. Подробно изучаются ключевые слова try, catch и flow для эффективной обработки ошибок в коде. Основной задачей является правильная обработка исключений внутри функции, в которой они были сгенерированы, чтобы избежать прерывания работы программы. Важным аспектом является также изучение различных типов и категорий исключений в C++ и способы их обработки, включая создание собственного класса, наследуемого от std::exception.

Условия для 1 варианта

Реализовать класс «Время». Поля: часы, минуты, секунды. Конструкторы: конструктор для инициализации полей. При необходимости добавить свои функции. При недопустимых параметрах объекта создается исключение. Для исключения создать класс, производный от out\_of\_range. Дополнительно созданный класс для исключения кроме наследуемых полей (необходимо их также использовать) содержит поля класса «Время» для недопустимого объекта и метод для печати всех данных. Продемонстрировать работу с объектами в случае возникновения исключения и его обработки и без возникновения исключения.

Основная часть

Исходный текст программы:

Файл заголовка sources.hpp:

#pragma once

#include <iostream>

class Time

{

private:

int hour;

int minute;

int second;

public:

Time();

Time(int, int, int);

void print() const noexcept;

};

class TimeException : public std::out\_of\_range

{

private:

int hour;

int minute;

int second;

public:

TimeException(const char\*, int, int, int);

void printException() const noexcept;

};

Файл описания sources.cpp:

#include "sources.hpp"

Time::Time() : hour(0), minute(0), second(0) {}

Time::Time(int hour, int minute, int second)

{

if (hour < 0 || minute < 0 || second < 0) {

throw TimeException("Time\_error: Time cannot be negative", hour, minute, second);

}

else if (hour > 23) {

throw TimeException("Time\_error: The number of hours in a day cannot be more than 24", hour, minute, second);

}

else if (minute > 59) {

throw TimeException("Time\_error: The number of minutes cannot be more than 59", hour, minute, second);

}

else if (second > 59) {

throw TimeException("Time\_error: The number of seconds cannot be more than 59", hour, minute, second);

}

this->hour = hour;

this->minute = minute;

this->second = second;

}

void Time::print() const noexcept

{

std::cout << "Time: " << hour << ":" << minute << ":" << second << std::endl;

}

TimeException::TimeException(const char\* message, int hour, int minute, int second) : std::out\_of\_range(message), hour(hour), minute(minute), second(second) {}

void TimeException::printException() const noexcept

{

std::cerr << what() << std::endl;

std::cerr << "Time: " << hour << ":" << minute << ":" << second << std::endl;

}

Исполняемый файл main.cpp:

#include "sources.hpp"

int main() {

try {

Time time\_negativ(12, -34, 65);

Time time\_incorrect\_number\_hours(26, 12, 34);

Time time\_incorrect\_number\_minutes(23, 67, 34);

Time time\_incorrect\_number\_seconds(14, 12, 78);

Time time\_corr(8, 36, 0);

time\_corr.print();

Time time\_incorrect(26, -12, 94);

time\_incorrect.print();

}

catch (const TimeException& ex) {

ex.printException();

}

return 0;

}

Снимки выполнения работы программы



Рисунок 1 – Входные данные



Рисунок 2 – Выходные данные



Рисунок 3 – Входные данные



Рисунок 4 – Выходные данные



Рисунок 5 – Входные данные



Рисунок 6 – Выходные данные



Рисунок 7 – Входные данные



Рисунок 8 – Выходные данные



Рисунок 9 – Входные данные



Рисунок 10 – Выходные данные

Заключение

Задачи лабораторной работы были решены, результаты проверены. Изучены механизмы генерации, обработки и создания собственных исключений в языке Си++.