Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный технологический университет»

**Документация**

«Таймер»

Минск 2022

**Содержание**

[1. Обратный таймер 3](#_Toc119012344)

[1.1. Алгоритм таймера 3](#_Toc119012345)

[2. Приложение c обратным таймером 5](#_Toc119012346)

[2.1 Псевдокод приложения 5](#_Toc119012347)

[2.2 Блок-схема приложения 6](#_Toc119012348)

[2.3 Тестирование приложения 7](#_Toc119012349)

[Приложение 8](#_Toc119012350)

# 1. Обратный таймер

Я рассмотрела одну из реализаций обратного таймера, где в основном используются проверки на “отсутствия” минут и секунд в нашем таймере с дальнейшей конвертацией часов в минуты и минуты в секунды.

## 1.1. Алгоритм таймера

Так как нам нужно выводить таймер в реальном времени, напишем функцию, которая будет осуществлять вывод таймера в консоль с интервалом в 1 секунду. Реализация функции приведена в коде 1.1.

void displayClock(int hours, int minutes, int seconds) {

system("cls"); // Очистка консоли

cout << setfill(' ') << setw(50) << "Обратный таймер\n";

cout << setfill(' ') << setw(55) << " ==========================\n";

cout << setfill(' ') << setw(29);

cout << "| " << setfill('0') << setw(2) << hours << " hrs / ";

cout << setfill('0') << setw(2) << minutes << " min / ";

cout << setfill('0') << setw(2) << seconds << " sec |" << endl;

cout << setfill(' ') << setw(55) << " ==========================\n";

}

Код 1.1

Для работы методов setfill и setw подключён заголовочный файл iomanip.

Реализация главной функции представлена в коде 1.2. Суть функции такова: ежесекундно уменьшается значение переменной seconds (это реализовано с помощью функции sleep(), параметром которой является время задержания выполнения программы в миллисекундах) до тех пор, пока значение не будет равным нулю. После этого таймер автоматически переведет 1 минуту в 60 секунд, уменьшив значение переменной minutes на единицу (код 1.2). Аналогично с часами: как только заканчиваются минуты, будет произведен перевод часов в минуты.

Корректность ввода часов, минут и секунд осуществляет код 1.3.

void timer(int hours, int minutes, int seconds) {

while (true) {

displayClock(hours, minutes, seconds); // Выводим время на экран

Sleep(1000); // Приостанавливаем выполнение программы на 1 секунду

seconds--; // Уменьшаем секунды на 1

if (seconds == 0) {

if (hours == 0 && minutes == 0 && seconds == 0) {// Если время закончилось

displayClock(hours, minutes, 0);

cout << "Время вышло" << endl;

return;

} else if (minutes == 0 && seconds == 0) {// Если минуты и секунды закончились

hours--; // Уменьшаем часы на 1

minutes = 59;

seconds = 59;

} else if (seconds == 0) {// Если секунды закончились

minutes--; // Уменьшаем минуты на 1

seconds = 59;

}

}

}

}

Код 1.2

int hours, minutes, seconds;

// Проверка на корректность ввода

do {

cout << "Введите часы: ";

cin >> hours;

} while (hours > 0);

do {

cout << "Введите минуты: ";

cin >> minutes;

} while (minutes > 0);

do {

cout << "Введите секунды: ";

cin >> seconds;

} while (seconds > 0);

Код 1.3

# 2. Приложение c обратным таймером

Я написала приложение, суть которого состоит в том, чтобы пользователь мог устанавливать обратный таймер на любой промежуток времени в формате ЧЧ:ММ:СС.

## 2.1 Псевдокод приложения

Начало

Функция displayClock

Очистить консоль

Заполнить консоль с помощью методов setfill и setw

Вывести в консоль значения переменных hours, minutes и seconds

Конец функции

Функция timer

Пока true

Вызвать displayClock(hours, minutes, seconds)

Sleep(1000);

Уменьшить seconds на 1

Если seconds == 0

Если hours == 0 && minutes == 0 && seconds == 0

Вызвать displayClock(hours, minutes, 0)

Вывести сообщение о том, что время вышло

return;

Конец "если"

Иначе если minutes == 0 && seconds == 0

Уменьшить hours на 1

Присвоить переменным minutes и seconds новое значение — 59

Конец "иначе если"

Иначе если seconds == 0

Уменьшить minutes на 1

Присвоить переменной seconds новое значение — 59

Конец "иначе если"

Конец "если"

Конец "пока"

Конец функции

Функция Main

Объявление и инициализация переменных hours, minutes и seconds

Пока hours > 0

Ввод hours

Конец "пока"

Пока minutes > 0

Ввод minutes

Конец "пока"

Пока seconds > 0

Ввод seconds

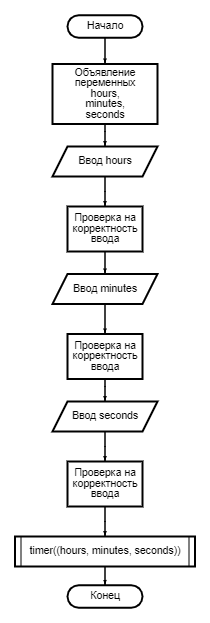
Конец "пока"

Вызов timer(hours, minutes, seconds)

Конец "main"

Конец

## 2.2 Блок-схема приложения



## 2.3 Тестирование приложения

Я произвела тестирование приложения на корректность работы таймера

Тестирование показало, что таймер работает исправно, переход с часов на минуты и с минут на секунды исправен (рисунок 2.1, 2.2). По окончании времени в консоль выводится соответствующая информация (рисунок 2.3).

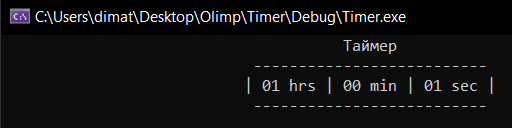


Рисунок 2.1

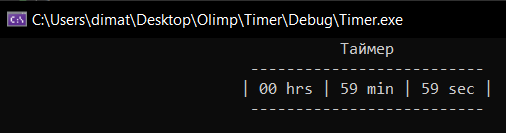


Рисунок 2.2

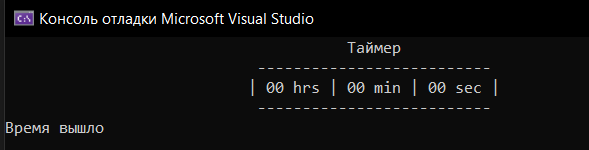


Рисунок 2.3

# Приложение

#include <iostream>

#include <Windows.h>

#include <iomanip>

using namespace std;

void displayClock(int hours, int minutes, int seconds) {

system("cls"); // Очистка консоли

cout << setfill(' ') << setw(50) << "Обратный таймер\n";

cout << setfill(' ') << setw(55) << " ==========================\n";

cout << setfill(' ') << setw(29);

cout << "| " << setfill('0') << setw(2) << hours << " hrs / ";

cout << setfill('0') << setw(2) << minutes << " min / ";

cout << setfill('0') << setw(2) << seconds << " sec |" << endl;

cout << setfill(' ') << setw(55) << " ==========================\n";

}

void timer(int hours, int minutes, int seconds) {

while (true) {

displayClock(hours, minutes, seconds); // Выводим время на экран

Sleep(1000); // Приостанавливаем выполнение программы на 1 секунду

seconds--; // Уменьшаем секунды на 1

if (seconds == 0) {

if (hours == 0 && minutes == 0 && seconds == 0) {// Если время закончилось

displayClock(hours, minutes, 0);

cout << "Время вышло" << endl;

return;

} else if (minutes == 0 && seconds == 0) {// Если минуты и секунды закончились

hours--; // Уменьшаем часы на 1

minutes = 59;

seconds = 59;

}

else if (seconds == 0) {// Если секунды закончились

minutes--; // Уменьшаем минуты на 1

seconds = 59;

}

}

}

}

int main() {

setlocale(LC\_ALL, "RUS");

int hours, minutes, seconds;

// Проверка на корректность ввода

do {

cout << "Введите часы: ";

cin >> hours;

} while (hours < 0);

do {

cout << "Введите минуты: ";

cin >> minutes;

} while (minutes < 0);

do {

cout << "Введите секунды: ";

cin >> seconds;

} while (seconds < 0);

timer(hours, minutes, seconds); // Запускаем таймер

return 0;

}