Оглавление

[1 Аннотация 2](#_Toc26715170)

[2 Введение 3](#_Toc26715171)

[3 Назначение и область применения программы 4](#_Toc26715172)

[4 Технические характеристики программы 5](#_Toc26715173)

[4.1 Постановка задачи на разработку программы 5](#_Toc26715174)

[4.2 Применяемые математические методы 5](#_Toc26715175)

[4.3 Описание и обоснование выбора метода организации входных, выходных и промежуточных данных 6](#_Toc26715176)

[4.4 Разработка модульной структуры программы 6](#_Toc26715177)

[4.5 Описание алгоритмов функционирования программы 10](#_Toc26715178)

[4.6 Обоснование состава технических и программных средств 17](#_Toc26715179)

[5 Вывод 18](#_Toc26715180)

# Аннотация

Представленная курсовая работа состоит из введения, трех глав, заключения, списка литературных источников и приложений.

В первой части работы рассматриваются общие сведения о программе, такие как вариант задания, основные модули программы, а также блок-схемы. Во второй части были описаны условия запуска программы, а также её проверка работоспособности программы.

# Введение

Целью данного курсового проекта является разработка программы расчета

среднего процента процессорного времени, которая обеспечивает получение оперативных данных об эффективности работы каждого процессора.

Предмет исследования данной темы – обработка данных, представленных в виде таблицы. Темой курсового проекта является разработка программы для обработки данных, представленных в форме таблицы. Любая таблица отражает какую-либо форму человеческой деятельности. С этой точки зрения программа, написанная в результате курсового проектирования, может считаться упрощенным подобием предметно-ориентированной базы данных.

Широко распространенным видом услуг, эффективно реализующихся с помощью компьютеров, является информационно-справочное обслуживание, которое подразумевает хранение сведений, прием новых сведений, обработку сведений, выдачу информации по запросам. Хранимые сведения в общем случае представляются записями. Для представления такого рода услуг создаются автоматизированные информационные системы (АИС) различного назначения.

Основная задача, решаемая в ходе создания АИС, состоит в том, чтобы организовать совместное хранение большого числа различных записей и выдавать по запросу любую из них независимо от того, какие записи и в каком порядке выдавались ранее.

# Назначение и область применения программы

Программное средство должно автоматизировать и упростить работу пользователя.

Областью применения данной программы является получение данных введённых пользователем о времени выполнения заданий на ЭВМ, а также суммы по видам времени по всем заданиям и средний процент времени центрального процессора по всем заданиям.

Программа предназначена для кампаний, которые хотят систематизировать данные о времени выполнения заданий на ЭВМ, а также суммы по видам времени по всем заданиям и средний процент времени центрального процессора по всем заданиям.

# Технические характеристики программы

## Постановка задачи на разработку программы

Даны сведения о времени выполнения заданий на ЭВМ (время измеряется в секундах). Структура записи: шифр задания (8 символов), код отдела (3 символа), Ф.И.О. программиста (15 символов), общее время прохождения задания, время центрального процессора − 5 знаков (t\_задания > t\_цп). Получить процент процессорного времени по каждому заданию, а также суммы по видам времени по всем заданиям и средний процент времени центрального процессора по всем заданиям.

## Применяемые математические методы

В ходе разработки данного программного обеспечения применялась формула расчёта процентного отношения. Для этого использовалось правило пропорции. Принцип работы показан на Рисунке 4.1.

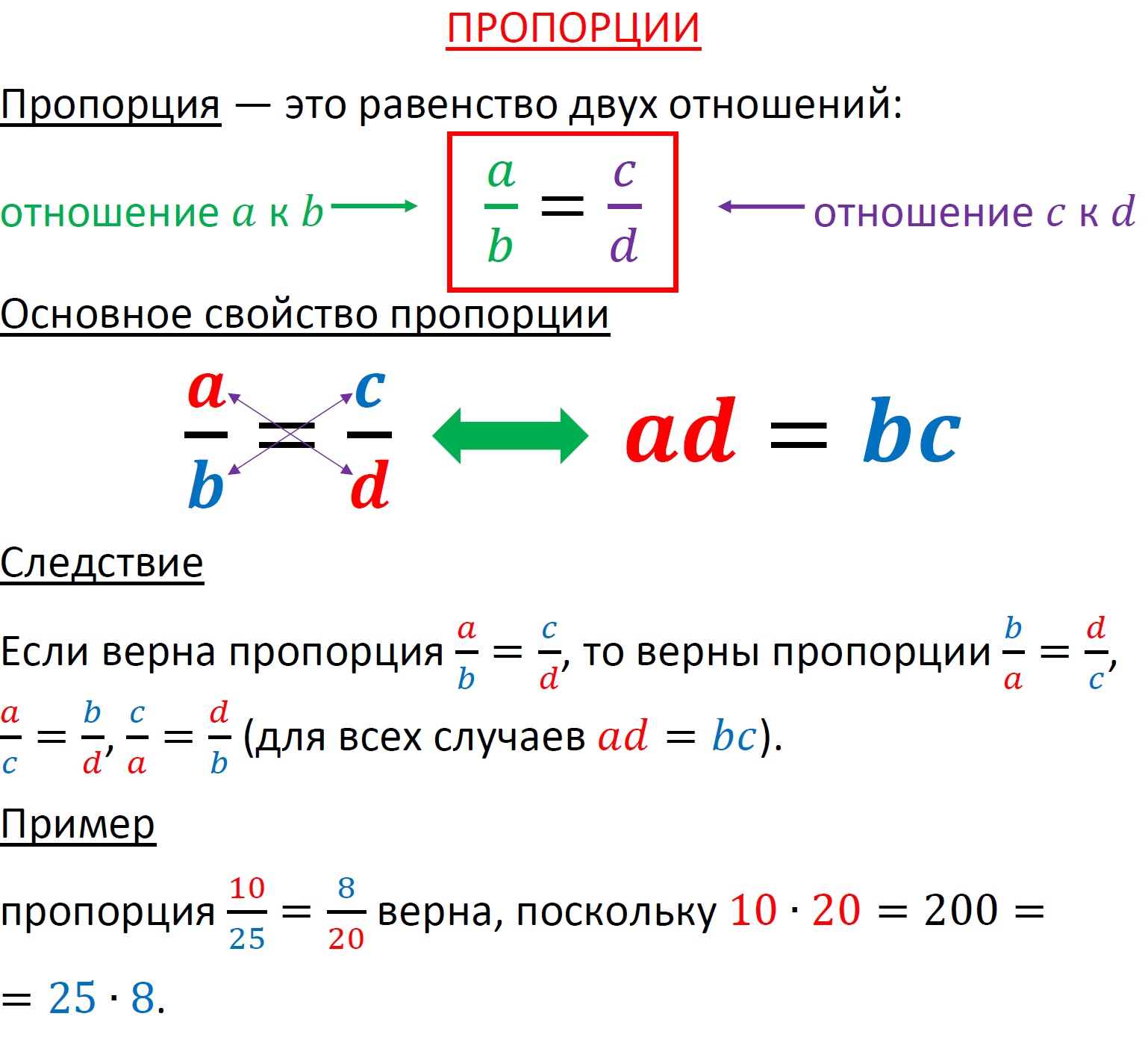


Рисунок 4.1 – правило пропорции

## Описание и обоснование выбора метода организации входных, выходных и промежуточных данных

Входными данными являются:

1. Шифр задания (8 символов)
2. Код отдела (3 символа)
3. Ф.И.О. программиста (15 символов)
4. общее время прохождения задания (5 знаков)
5. время центрального процессора (5 знаков)

При этом общее время прохождения задания должно быть больше времени центрального процессора.

В качестве организации данных выбран двунаправленный список, так как он более гибок по сравнению с однонаправленным списком.

В качестве выходных данных формируется таблица, в которой можно редактировать, удалять и сортировать данные.

## Разработка модульной структуры программы

Определим необходимые константы:

const int up = 72,

down = 80,

right\_btn = 77,

left\_btn = 75,

enter = 13,

esc = 27,

del = 83;

const int SIZE\_arr\_filename = 50;

// названия пунктов

const string items[7] = {

" Ввод данных ",

" Печать данных ",

" Сохранить данные ",

" Сохранить в другой файл ",

" Поиск ",

" Выбрать другой файл ",

" Выход из программы " };

// названия для сортировки

const string sort\_items[5] = {

"| Шифр задания ",

"Код отдела ",

"ФИО ",

"Общее время ",

"Время ЦП" };

Вывод данных на экран. Параметрами передаются указатель на первый и последний элемент, номер активного пункта меню, номер редактируемого элемента, счетчик для i и номер текущей страницы. Данная функция отвечает за постраничный вывод данных на экран:

time\_task\* print(time\_task\* end, time\_task\* real\_beg ,time\_task\* beg, int active, int edit\_el, int print\_count\_num\_pages, int print\_page);

Чтение из текстового файла. Параметрами передается название файла, а также указатели на начало и конец списка. Данная функция отвечает за чтение из текстового файла:

int read\_file(string filename, time\_task\*\* beg, time\_task\*\* end);

Определение типа файла. Тип файла выбирается пользователем при помощи пункта меню. Параметром мы передаем указатель на начало списка.

int write\_file(time\_task\* temp);

Сохранение в файл. Данная функция несет ответственность за сохранение данных в файл. Параметры функции: указатель на начало списка, название файла, номер элемента массива всех названий файлов, а также тип файла, который выбрал пользователь.

void write\_filetype(time\_task\* temp, string filename, int el, int filetype);

Удаление элемента из списка. В данную функцию передается указатель на начала списка и номер элемента, который мы хотим удалить:

time\_task\* delete\_el(time\_task\* beg, int num\_del);

Чтение из бинарного файла. Параметрами передается название файла, а также указатели на начало и конец списка. Данная функция отвечает за чтение из бинарного файла:

int read\_bin\_file(string filename, time\_task\*\* beg, time\_task\*\* end);

Сортировка. Функция необходима для сортировки данных. Параметры: указатель на начало списка, поле, по которому пользователь хочет произвести сортировку и ее направление:

void sort(time\_task\* beg, int field\_for\_sort, int sort\_direction);

Редактирование. Функция предназначена для корректировки определенного поля, который пользователь может выбрать при просмотре меню. Функция принимает следующие параметры: указатель на начало и конец списка, номер активного пункта меню, элемент для редактирования, счетчик для i, номер текущей страницы и высота всей таблицы:

void edit(time\_task\* end, time\_task\* real\_beg, time\_task\* beg, int active, time\_task\* \_edit\_ob, int edit\_count\_num\_pages, int edit\_page, int edit\_i);

Поиск. Функция нужна для поиска определенного элемента, который захочет найти пользователь. В качестве параметра передается лишь указатель на начало списка:

void find(time\_task\* beg);

Инициализация меню. Данная функция осуществляет вывод пунктов меню на экран при помощи которых пользователь сможет взаимодействовать с программой. Параметрами этой функции будут являться номер активного меню, массив названий пунктов меню и их количество:

int menu(int& active, const string items[], int num\_el);

Расчет процента процессорного времени. Получает процент процессорного времени по каждому заданию, а также суммы по видам времени по всем заданиям и средний процент времени центрального процессора по всем заданиям. У функции два параметра – общее время выполнения задания и время процессора:

float percent\_time\_cpu(float a, float b);

Выделение памяти. Для этого функции передается указатель на начало

списка и конец, а также данные, которые ввел пользователь.

void input(time\_task\*& beg, time\_task\*& end, const time\_task& info);

## Описание алгоритмов функционирования программы



Рисунок 4.2 – структурная схема печати данных



Рисунок 4.3 – структурная схема печати данных



Рисунок 4.4 – структурная схема печати данных



Рисунок 4.5 – структурная схема печати данных (если нажали клавишу вверх)



Рисунок 4.6 – структурная схема печати данных (если нажали клавишу esc)



Рисунок 4.7 – структурная схема печати данных (если нажали клавишу вниз)



Рисунок 4.8 – структурная схема печати данных (если нажали клавишу вправо)



Рисунок 4.9 – структурная схема печати данных (если нажали клавишу влево)



Рисунок 4.10 – структурная схема печати данных (если нажали клавишу del)



Рисунок 4.11 – структурная схема печати данных (если нажали клавишу enter)

## Обоснование состава технических и программных средств

Для написания программы была выбрана операционная система Windows, так как она самая распространённая. В качестве среды разработки было выбрано Visual Studio 2019. В качестве языка программирования был выбран язык высокого уровня C++.

По умолчанию программа расположена на гибком диске 3,5 дюйма. В состав программы входит исполняемый файл Kursach\_IIP.exe. Именно с запуска файла Kursach\_IIP.exe начинается работа приложения. Следующий шаг после запуска исполняемого файла состоит в формировании начальной базы данных, т.к. в исходном варианте файл со сформированной справочной системой отсутствует. Формирование БД исключительная прерогатива конечного пользователя. Использование защиты от несанкционированного использования и копирования не предусмотрено.

# Вывод

В результате работы была написана программа, в основе алгоритма которой положена структура данных в виде двунаправленного списка, позволяющая выполнять просмотр данных в двух направлениях. База данных способна хранить большое количество записей и выдавать по запросу любую из них. Реализовано меню, дающее возможность пользователю взаимодействовать с программой и выполнять определенные действия над ней, такие как: удаление, просмотр данных, редактирование, поиск, запись в бинарный и текстовый файл, а также чтения с них.   
Таким образом цель курсового проекта была достигнута.   
Программное средство должно автоматизировать и упростить работу пользователя. Она может применяться не только индивидуальными пользователями, но и предпринимателями малых и средних корпорация, которые хотят систематизировать данные о времени выполнения заданий на ЭВМ, а также суммы по видам времени по всем заданиям и средний процент времени центрального процессора по всем заданиям, после соответствующей доработки.