Оглавление

[Введение 3](#_Toc152186788)

[1 Постановка задачи 4](#_Toc152186789)

[2 Обзор литературы 5](#_Toc152186790)

[2.1 Обзор методов и алгоритмов решения поставленной задачи 5](#_Toc152186791)

[3 Функциональное проектирование 6](#_Toc152186792)

[3.1 Структура входных и выходных данных 6](#_Toc152186793)

[3.2 Разработка диаграммы классов 6](#_Toc152186794)

[3.3 Описание классов 6](#_Toc152186795)

[4 Разработка программных модулей 7](#_Toc152186796)

[4.1 Разработка схем алгоритмов (два наиболее важных метода) 7](#_Toc152186797)

[4.2 Разработка алгоритмов (описание алгоритмов по шагам, для двух методов) 7](#_Toc152186798)

[4.2.1 Алгоритм загрузки тестового задания 7](#_Toc152186799)

[4.2.2 Алгоритм 2 7](#_Toc152186800)

[5 Результаты работы 8](#_Toc152186801)

[Заключение 9](#_Toc152186802)

[Литература 10](#_Toc152186803)

[Приложение А Диаграмма классов 11](#_Toc152186804)

[Приложение Б Схема алгоритма А 12](#_Toc152186805)

[Приложение В Схема алгоритма Б 13](#_Toc152186806)

[Приложение Г Исходный текст программы (с комментариями) 14](#_Toc152186807)

# Введение

В данном разделе обще описывается о чём курсовой проект.

1. Постановка задачи

В данном разделе кратко описывается поставленная задача.

1. Обзор литературы

В данном фрагменте представлен обзор аналогов и теории по теме курсового проекта на 3-5+ страниц, например:

«При работе с персональным компьютером и для написания большого количества текста часто требуется владение навыком слепой печати. Данный метод был впервые применён в 1878 году Франком Эдвардом Макгуррином[1].

……….»

## Обзор методов и алгоритмов решения поставленной задачи

Для получения символа нажатой клавиши могут применятся несколько способов:

* Считывая символ клавиши с помощью std::cin;
* Получая скан-код нажатой клавиши.

Наиболее лучшим методом будет считывание скан-кода, поскольку нет привязки к языку системы.

…

1. Функциональное проектирование

## Структура входных и выходных данных

В качестве входных данных используется файл testText страутура которого представлена в таблице 3.1.1.

Таблица 3.1.1 – Файл testText.

|  |  |
| --- | --- |
| Num | Text |
| Номер примера | Текст примера |

## Разработка диаграммы классов

Диаграмма классов представленна в приложении А.

## Описание классов

Class TextPrimer

Данные:

Num: int – номер примера

Text: string – Текст примера.

Методы:

GetText() – ….

….

1. Разработка программных модулей

## Разработка схем алгоритмов (два наиболее важных метода)

В приложениях Б и В представлены схемы методов onKeyPressEvent() и errorProcessing() соответственно.

## Разработка алгоритмов (описание алгоритмов по шагам, для двух методов)

### Алгоритм загрузки тестового задания void loadTestText(const char\* str)

Загружает тестовое задание из файла.

Входные данные – название файла.

Выходные данные – нет.

* + - 1. Открыть файл с параметрами чтения, если файл не отрылся, то перейти к шагу 2, иначе – к шагу 4.
      2. Вывести сообщение об ошибке открытия файла.
      3. Перейти к шагу 5.
      4. Прочитать данные из файла в List<string>.
      5. Конец

### Алгоритм 2

1. Результаты работы

На рис. 5.1 представлено окно приложения с запущенным выполнением тестового примера.

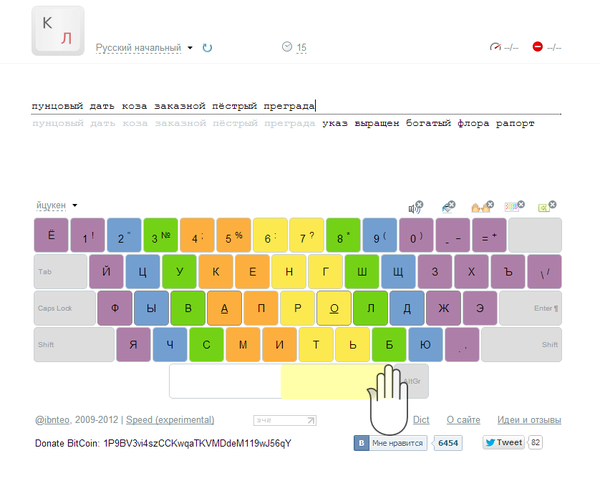


Рисунок 5.1 – Окно приложения

# Заключение

Текст

# Литература

1. Макгуррин, Франк Эдвард / Википедия [Электронный ресурс]. – Режим доступа : <https://ru.wikipedia.org/wiki/Макгуррин,_Франк_Эдвард> . – Дата доступа : 01.10.2023.

# Приложение А

(обязательное)

Диаграмма классов

# Приложение Б

(обязательное)

Схема алгоритма

# Приложение В (обязательное) Схема алгоритма

# Приложение Г (обязательное) Исходный текст программы (с комментариями)

1 #include <iostream>

2 #include <fstream>

3 #include <string>

4 using namespace std;

5

6 class String

7 {

8 char \*st;

9 int size;

10 public:

11

12 String(char \*ST, int SIZE) : size(SIZE)

13 {

14 st = new char[size];

15 strcpy\_s(st, size+1, ST);

16 }

17 ~String() { delete[] st; }

18 String(const String &s) // копирующий конструктор необходим, так как

19 { // при перегрузке << в функцию operator переда-

20 st = new char[s.size]; // ется объект, содержащий указатель на строку,

21 strcpy\_s(st,s.size+1, s.st); // а в конце вызовется деструктор для

22 } // объекта obj

23 friend ostream &operator<<(ostream &, const String);

24 friend istream &operator>>(istream &, String &);

25 };

26

27 ostream &operator<<(ostream &out, const String obj)

28 {

29 out << obj.st << endl;

30 return out;

31 }

32

33 istream &operator>>(istream &in, String &obj)

34 {

35 in >> obj.st;

36 return in;

37 }

38

39 int main()

40 {

41 String s("asgg", 10), ss("aaa", 10);

42 int state;

43 ofstream out("file");

44

45 if (!out)

46 {

47 cout << "ошибка открытия файла" << endl;

48 return 1; // или exit(1)

49 }

50 out << "123" << endl;

51 out << s << ss << endl; // запись в файл

52 ifstream in("file");

53 while (in >> ss) // чтение из файла

54 {

55 cout << ss << endl;

56 }

57 in.close();

58 out.close();

59 return 0;

60 }

61