Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования.

«Сибирский государственный университет телекоммуникации и информатики»

Кафедра вычислительных систем

Курсовой работа по дисциплине «Технологии разработки программного обеспечения» *“KeyboardNinja. Клавиатурный тренажер.”*

Выполнил:

Студент 2 курса гр. ИП-511

Рыжов И.Е.

Проверил:

Доц., к.ф- м.н Пудов С.Г.

Оглавление

[Введение и постановка задачи 2](#_Toc485811041)

[Техническое задание 3](#_Toc485811042)

[Тема: 3](#_Toc485811043)

[Состав команды: 4](#_Toc485811044)

[Функционал программы 4](#_Toc485811045)

[Формат входных данных: 4](#_Toc485811046)

[Интерфейс приложения: 4](#_Toc485811047)

[Описание выполненного проекта: 4](#_Toc485811048)

[Личный вклад в проект: 8](#_Toc485811049)

[Текст программы: 9](#_Toc485811050)

# Введение и постановка задачи

Клавиатурный тренажёр

Клавиату́рный тренажёр — вид компьютерных программ или онлайн-сервисов, предназначенных для обучения набору на компьютерной клавиатуре.

Обычно целями тренажёров являются:

* научить слепому методу печати, в частности, задействовать для набора все десять пальцев рук
* увеличить скорость набора
* уменьшить количество опечаток
* улучшить ритмичность набора (что позволяет уменьшить усталость при наборе)

Некоторые клавиатурные тренажёры предлагают ряд уроков для изучения раскладки, ведут статистику обучения.

Задача: Написать программу для игры в “виселицу”. Проект должен иметь многофайловую структуру и покрыт тестами.

# Техническое задание

### Тема:

KeyboardNinja. (Клавиатурный тренажер)

### Состав команды:

* Рыжов Иван (ИП-511)
* Чакин Сергей (ИП-511)

### Функционал программы

* Программа, предназначенная для изучения раскладки клавиатуры компьютера и обучения быстрой печати.
* Программный продукт должен решать следующие задачи:

1. Ввод определенного текста с клавиатуры в программу
2. Поиск ошибок
3. Вывод на экран результата теста

* Входными данными для работы программы является текст набранный с клавиатуры, выходными данными результат после прохождения теста.

### Формат входных данных:

* Строка вводимая с клавиатуры

### Интерфейс приложения:

* Приложение работает в консольном режиме.
* Меню следующие пункты:

1) Easy

2) Norm

3) Hard

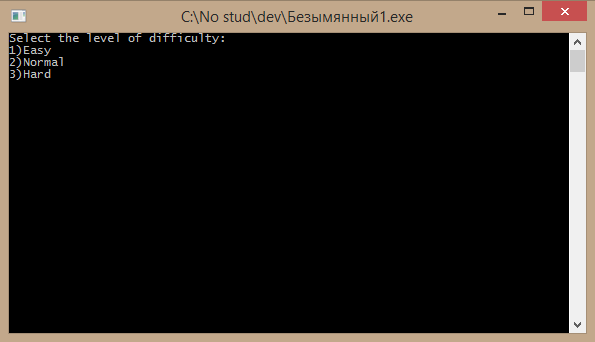
* Вывод результата теста и оценка

Ссылка на сайт:  
*https://github.com/SergeyChakin/KeyboardNinja/tree/vetka*

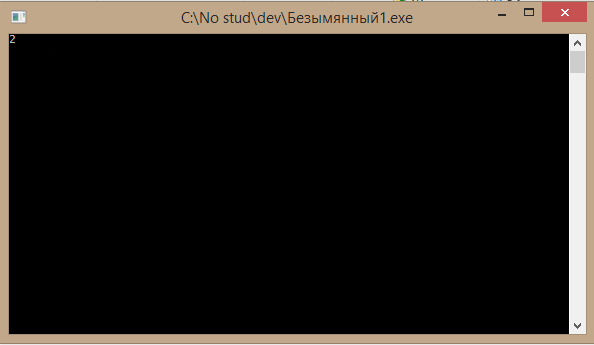
# Описание выполненного проекта:

Программа *“KeyboardNinja. Клавиатурный тренажер.”, такая программа помогает пользователю совершенствовать скорость печати с клавиатуры.*И так, У нас должна быть программа с разбитым окном на две части. В верхнею часть окна выводиться текст (само задание, которое пользователь переписывает). И в нижнею часть окна, куда вводиться текст пользователя.   
Что вводиться в верхнею часть, пусть программа будет предлагать пользователю выбор сложности. В зависимости от выбранной трудности программа будет считывать текст (текст можно будет менять не выходя из программы).   
Необходима функция сравнения текстов. То есть первый и второй тексты будут сравниваться. Будет подсчёт кол-ва ошибок. Исходя из них, программа будет оценивать проделанную работу пользователя.  
А так же функцию Timer для подсчёта времени (пользователю будет интересно сколько он времени писал).   
А что выводит программа? Так, она сравнивает тексты и считает время, это программа и выведет. То есть, кол-во ошибок и скорость печати. А так же программа будет оценивать работу пользователя, в зависимости от ошибок.

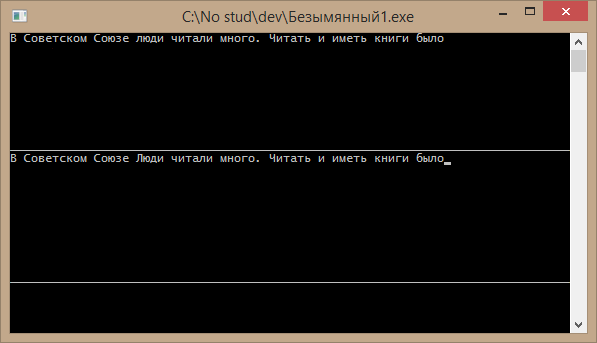
При запуске программы появляется окно с возможностью выбора сложности.

 *Рис 1 Запуск программы*.

Выбираем сложность (любую), вводим соответствующую цифру и нажимаем ENTER.  
После выбора у вас будет 5 секунд на подготовку.

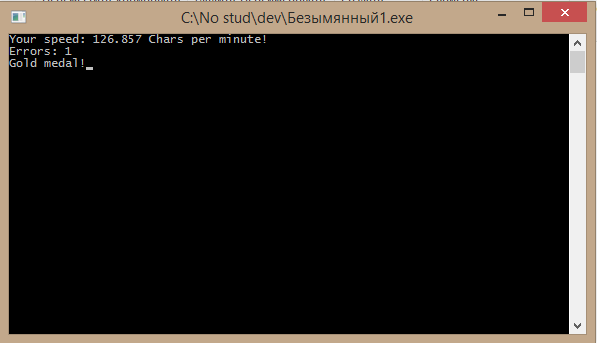
*Рис 2 обратный отсчёт.*

По истечению 5-ти секунд программа будет частями выводить текст.



*Рис 3 Ввод текста.*

После завершения ввода программа посчитает время, ошибки и оценит вас.



*Рис 4 Оценка пользователя.*

# Личный вклад в проект:

Я написал основной код программы, алгоритмы, идею.

# 

# Текст программы:

#include <stdio.h>

#include <stdlib.h>

#include <iostream>

#include "fstream"

#include <string>

#include <math.h>

#include <algorithm>

#include <windows.h>

#include <cstdio>

#include <conio.h>

#include <time.h>

#include <locale.h>

using namespace std;

string a, \*b, \*ans, input = "", output = "",name1;//Объявляем наши строки; b,ans это массивы строк, в которые мы записываем разбирую по словам

//входную и выходную последовательности; input, output это просто строки, в которой хранится текст, нужны для поиска ошибок;

//name1 это имя файла, который мы открываем для считывания

COORD point, point1, point2;//координаты, нужные для перехода к определенной строки и столбцу консоли

HANDLE hout = GetStdHandle(STD\_OUTPUT\_HANDLE);//Определяем дискриптор выходного потока данных(нужно для корректной работы SetConsoleCursorPosition!!)

void ramka(){//функция рисующая рамку

setlocale(0,"");

point2.X = 0;//определяем координаты x и н куда перенесем курсор в консоли

point2.Y = 9;

SetConsoleCursorPosition(hout, point2);//переносим курсор

for (int i = 1; i <= 80; ++i){//рисуем верхнюю линию

printf("%c", '\_');

}

point2.Y = 20;

SetConsoleCursorPosition(hout, point2);//переносим курсор

for (int i = 1; i <= 80; ++i){//рисуем нижнюю линию

printf("%c", '\_');

}

}

int modul(int x){//функция модуля

if (x < 0)

return -x;

return x;

}

void result(int anserr){//оцениваем результат

if (anserr>17) cout << "Train more!";

else if (anserr > 12) cout << "You can do batter!";

else if (anserr >= 5) cout << "Not bad!";

else if (anserr == 4) cout << "Wooden medal!";

else if (anserr == 3) cout << "Bronze medal!";

else if (anserr == 2) cout << "Silver medal!";

else if (anserr == 1) cout << "Gold medal!";

else if (anserr == 0) cout << "Brilliant!";

}

int main()

{

SetConsoleCP (1251);

SetConsoleOutputCP (1251);

time\_t start1, end1, forstart, forend;//время для таймеров

int counts = 0, countans = 0, i = 0, anserr = 0, lvl;

char c;

point.X = 0;

point.Y = 0;

point1.X = 0;

point1.Y = 10;

ifstream cin1("easy.in");

ifstream cin2("norm.in");

ifstream cin3("hard.in");

cout << "Select the level of difficulty:" << endl << "1)Easy" << endl << "2)Normal" << endl << "3)Hard" << endl;//выбираем уровень сложности

cin >> lvl;

b = new string[200];

ans = new string[200];

switch (lvl){

case 1:

while (cin1 >> a){//считываем с файла текст

a += ' ';

b[counts] = a;

input += a;

counts++;

}

break;

case 2:

while (cin2 >> a){//считываем с файла текст

a += ' ';

b[counts] = a;

input += a;

counts++;

}

break;

case 3:

while (cin3 >> a){//считываем с файла текст

a += ' ';

b[counts] = a;

input += a;

counts++;

}

break;

default:

break;

}

system("CLS");

time(&forstart);

int k = 1;

while (k < 7){//цикл отсчета времени(5,4,3,2,1)

time(&forend);

if (forend - forstart >= k){

system("CLS");

cout << 6 - k;

k++;

}

}

time(&start1);//засекаем время

for (i = 0; i < counts; ++i){//Идем по всем словам, которые мы считали

SetConsoleCursorPosition(hout, point);//Устанавливаем курсор в начало

for (int j = 0; j < i + 1; ++j){//Выводим слова, которые пользователю надо вводить

cout << b[j];

}

if (i != 0){//здесь мы вычисляем на сколько надо сдвигать курсор

point1.X += ans[i - 1].size();//а сдвигать его надо на размер предыдущего слова

if (point1.X >= 80)

point1.X -= 80, point1.Y++;

}

ramka();//рисуем рамку

SetConsoleCursorPosition(hout, point1);//устанавливаем курсор на место для ввода

while (1){//посимвольно читаем то, что вводит пользователь

c = getch();//собственно читаем через getch()

cout << c;//выводим то, что он написал

if (c == '\b'){//здесь идет обработка нажатия бэкспейса(удаления символа)

cout << " \b";//проблема в том, что бэкспейс мы не должны добавлять в наше слово, а при его нажатии, мы должны удалить последний введенный символ

}

if (c == '\b'){

if (ans[i].size()){//проверка, чтоб мы не удаляли из пустой строки

ans[i].erase(ans[i].size() - 1, 1);//вот тут мы удаляем символ

}

}

else{

ans[i] += c;//здесь мы прибовляем введеный символ к слову

}

if (c == ' ' || c == 13){//если нажат пробел или enter то ввод слова закончен

break;

}

}

output += ans[i];//Добавляем слово к выходной строке

}

time(&end1);

end1 -= start1;//смотрим сколько времени прошло

system("CLS");

for (int i = 0; i < min(input.size(), output.size()); ++i){//в этом цикле мы идем до размера минимальной из строк

if (input[i] != output[i]){//проверяем, ошибки

anserr++;//считаем их количество

}

}

anserr += modul(input.size() - output.size());//нам надо к ошибка добавить разницу размеров входной и выходной строки

cout << "Your speed: " << float(output.size()) / ((float(end1) / 60)) << " Chars per minute!" << endl;//выводим скорость ввода

cout << "Errors: " << anserr << endl;//количество ошибок

result(anserr);//и результат

getch();

return 0;

}