**Курсовой проект по дисциплине**

**«Технологии разработки программного обеспечения»**

*“KeyboardNinja. Клавиатурный тренажер.”*

Выполнили:

Студенты группы ИП-511

Устинова Анастасия

Чакин Сергей

Рыжов Иван

Новосибирск

2016

Содержание:

1. [Техническое задание](#_Техническое_задание)
2. [Цель работы](#_Цель_работы)
3. [Код программы](#_КОД_ПРОГРАММЫ)
4. [Программа](#_Программа)

# **Техническое задание**

Программа, предназначенная для изучения раскладки клавиатуры компьютера и обучения быстрой печати.

Программный продукт должен решать следующие задачи:

* Ввод определенного текста с клавиатуры в программу
* Поиск ошибок
* Вывод на экран результата теста

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

Формат входных данных.

Входные данные подаются в текстовом формате.

1. Интерфейс приложения.

2. Меню следующие пункты:

1) Easy

2) Norm

3) Hard

3. Вывод результата теста и оценка

Приложение работает в консольном режиме.

# **Цель работы**

Создание клавиатурного тренажера – программы для обучения набору на компьютерной клавиатуре. Он помогает:

* Увеличить скорость набора
* Уменьшить количество опечаток
* Улучшить ритмичность набора

# **Код 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 разбитую 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 //дескриптор выходного потока данных (нужно для корректной работы //SetConsoleCursorPosition!!)

void ramka(){//функция рисующая рамку

setlocale(0,"");

point2.X = 0;//определяем координаты x и y 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 прибавляем введённый 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;

}

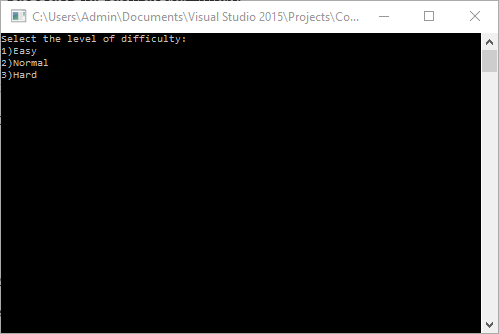
# **Программа**

Включив программу вам понадобится выбрать уровень сложности. Всего 3 уровня сложности:

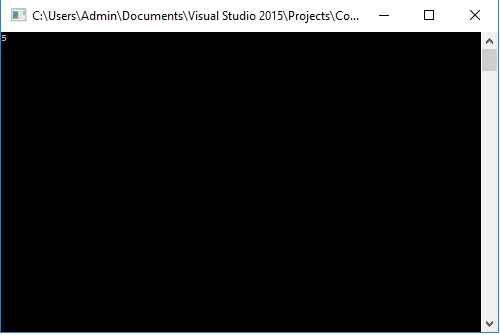
1) Easy

2) Norm

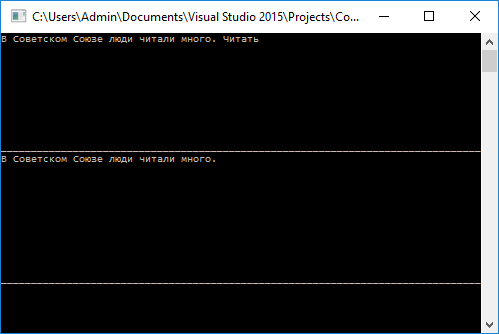
3) Hard

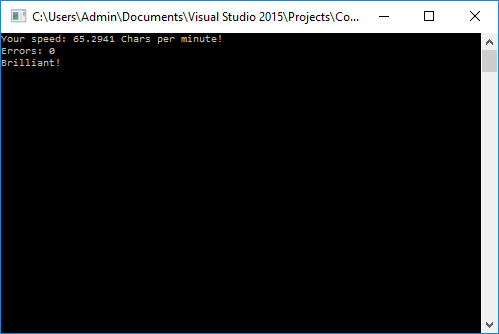


Выбрав уровень сложности включается обратный отсчет (5 секунд)



После того как пройдет 5 секунд начнется ввод



Когда введете полностью текст, программа покажет вашу скорость набора в минуту, сколько вы совершили ошибок и оценку

# **https://github.com/SergeyChakin/KeyboardNinja.git**